

SEMANTICS

ANDRÁS KORNAI

Draft version 1.03, March 1 2018 Please do not circulate, quote, or cite without express permission from the author, who can be reached at andras@kornai.com.

Proofreading complete

I do not know whether it must be said today that the critical task still entails faith in Enlightenment; I continue to think that this task requires work on our limits, that is, a patient labor giving form to our impatience for liberty.



To Ágnes

Preface

Semantics is the study of meaning. Except for the last Chapter, the primary focus of the book is the meaning of linguistic expressions, typically full sentences and longer texts, as opposed to the meaning of [computer programs](#), [mathematical formulas](#), or broader [semiotic](#) concerns. In everyday use ‘semantics’ refers more to the meaning of words, in fact the [Urban Dictionary](#) defines semantics as

The study of discussing the meaning/interpretation of words or groups of words within a certain context; usually in order to win some form of argument. *Now come on, let's not get bogged down in semantics.*

The field can be equally viewed as a chapter of linguistics, computer science, philosophy, or cognitive science, and to a certain extent the organization of this book will reflect this ambiguity by specifically marking some paragraphs on the margin as **Ling** and **Comp**, and occasionally as **Phil** or **CogSci**. Since no person can be expected to be an expert in all these fields, the prerequisites for each will be discussed separately.

Who should read this book

Our aim is to present the conceptual and formal tools required for building semantic systems capable of understanding text, both for specific tasks such as [information extraction](#) and [question answering](#) and for broad undertakings such as the [semantic web](#). Our goal here is to present the fundamental ideas that working systems rest on, and our textbook is aimed primarily at the computer science or engineering student interested in developing semantic systems. The ideal reader is a *hacker*, ‘a person who delights in having an intimate understanding of the internal workings of a system’. This means not just willingness to try and experiment with things, but also a positive attitude toward research, mathematical modeling in particular. The book is quite demanding in this respect: sentences are long, words of more than three syllables are often encountered, sidebar material and other typographical gimmicks aimed at supporting a rapid lets-find-the-keywords-and-get-it-over-with style of reading are avoided, and there is an



assumption that the reader will take the time to solve the exercises and read up on unfamiliar material as needed.

Comp



The emphasis is on the ideas, but a fair amount (at last count, some 2,500 lines) of code is also available at [GitHub](#), and the book is designed to support a very hands-on study plan that starts with the code and consults the book only as needed. Readers following this plan should be warned that only about a third of what is discussed in the book is actually accompanied by code, they have to contribute the rest themselves. Other, more traditional reading plans are suggested in Section 1.5.

Readers who study semantics because they need code that performs some semantic task will find references within the documentation of the code to the relevant landing sites in the book. Hyperlinks provide two-way crosslinking between the text and the growing body of Python code that implements the main ideas. This material is intended for the experienced software developer only –the book does not provide an introduction to Python– and should not be thought of as providing detailed documentation for the code. Readers are strongly encouraged to contribute to the main repository at <https://github.com/kornai/4lang> under a CC attribution or similar license that is weaker (more permissive) than GPL in that it must permit commercial reuse, and cannot have any viral effect on the rest of the code (GNU LPGL, BSD, and similar licenses are fine).



The computationally oriented reader, ideally a graduate student or advanced undergraduate in computer science/engineering, will find the book self-contained, except for the mathematical prerequisites summarized in Chapter 2. These will be used, and further developed, in the rest of the book with natural, rather than programming languages in mind. Because of this choice of subject, the material has surprisingly little in common with the mathematical logic prerequisites now taken for granted in programming language semantics, where the central attraction is [proofs as programs](#), the remarkable coincidence of two vocabularies, one built by logicians for the analysis of mathematical theorem proving (briefly touched in Section 2.6), and the other by computer scientists for the study of computation. For the trained functional programmer or logician the attraction of this nexus is almost irresistible, but natural language will pull us toward a considerably simpler, *zeroth order* theory, propositional calculus with some modal extensions, where the hard questions center around learnability of the concepts.



Ling

It requires a significant amount of linguistics to build systems that deal with natural language input, and paragraphs marked **Ling** are aimed at the reader who lacks these prerequisites. Within the confines of this volume we could not possibly present the technical machinery we take for granted in linguistics, and these paragraphs are intended as pointers to the linguistic literature, with the primary goal of facilitating self-study. The reader should be warned in advance that our selection of this material is strictly utilitarian, and following the pointers will not lead to a well-rounded picture of linguistic thought, not even of contemporary linguistic semantics. The timely appearance of an excellent volume devoted entirely to compositional semantics, Jacobson (2014), made it possible to devote more space to lexical semantics here and still keep this volume to manageable size.

Students of linguistics, especially those with a computational mindset, will likely have an easier time with the material they have to learn, though this will require considerable refactoring, and the occasional bit of unlearning, of the classical formal semantics curriculum. Those students whose computational background is weak should begin with Jurafsky and Martin (2009) or Bird, Klein, and Loper (2009). Very little is assumed from philosophy or cognitive science (see below), but there are notable mathematical prerequisites, discussed in Chapter 2.

The ideas discussed in paragraphs marked **Phil** will most often pertain to the branches of philosophy known as [philosophy of language](#) and [philosophy of science](#). References to philosophical thought will be especially frequent when we need to approach some question from the ground up, as in Chapter 3. Time and again we take the opportunity to point out connections with the ideas of philosophers, but such remarks, except perhaps for Section 9.1, do not amount to a goal-directed introduction to the relevant chapters of philosophy and are quite insufficient for self-study. Rather, they are intended for those readers who are already sensitive to philosophy, with the goal of orienting these readers how the positions taken here fit into the larger philosophical debate surrounding these subjects.

We offer new solutions to some well-known philosophical puzzles, in particular [The Heap](#) (sorites) and [Supererogation](#), but these are oriented toward practical goals (since the problems actually come up in system design) and not intended as a fully exhaustive philosophical treatment. In general, students of philosophy will learn a fair amount about some of the big questions: What is meaning? What is knowledge? What is truth? but no systematic philosophical treatment of any of these large subjects is given here. What is offered instead is a highly technical apparatus capable of *modeling* meaning, knowledge, and to a lesser extent, truth, both by computational and by mathematical (more algebraic than logic-based) means.

Neither the computational nor the mathematical prerequisites are normally covered in (philosophical) logic, but we will offer a reading plan geared toward philosophy in Section 1.5. In reading this book, one thing the philosopher has to unlearn, or at least strongly control, is the urge toward a highly technical language. When we analyze *right*, what we provide is an analysis of the everyday notion, not some refined theory of rights. We will of course distinguish *right*₁ ‘dextra’ from *right*₂ ‘bonus’ and *right*₃ ‘ius’, but our definition of the last one is simply ‘law’ and *law* in turn is defined as `rule, system, society/2285 HAS, official, ' ACCEPT, ABOUT can/1246[person[=TO]]` (see Section 6.5 for the formal theory of these definitions). In rough paraphrase, laws are systems of rules that societies have, they have official status, people accept them, and laws are about what people can do. But wait, aren’t there unjust laws, ones that people don’t accept? Aren’t there rights that transcend society? We hold that these questions, valuable as they are, cannot be very fruitfully approached through the study of everyday language. To quote from Kornai (2007):

Since almost all social activity ultimately rests on linguistic communication, there is a great deal of temptation to reduce problems from other fields of inquiry

Phil



to purely linguistic problems. Instead of understanding schizophrenia, perhaps we should first ponder what the phrase *multiple personality* means. Mathematics already provides a reasonable notion of ‘multiple’, but what is ‘personality’, and how can there be more than one per person? Can a proper understanding of the suffixes *-al* and *-ity* be the key?

CogSci



Originally, understanding systems were built by researchers like [Allen Newell](#) and [Herbert Simon](#) working on [Artificial Intelligence](#) who attempted to model human cognition, or at least borrow design ideas from what was known about the organization of mind at the time. Some of these systems, such as [SOAR](#) or [ACT-R](#), are still in use, while others have been abandoned in favor of newer cognitive architectures like [OpenCog](#). With the emergence of [functional MRI](#) techniques the field has grown enormously, but there is very little in this book that connects to this already vast, and still rapidly growing, literature. The reason, besides the obvious limitations of the author, is that trying to borrow ideas from nature turned out to be a dead end in natural language processing.

Major semantic systems like IBM’s [Watson](#) are not giant electronic brains, in fact they borrow very little from our understanding of biological systems. There may be [neural networks](#) used in various components, but more often than not there are other statistical learners like [support vector machines](#) which do away with the biological metaphor entirely. Within cognitive science there is a renewed effort towards biologically inspired models, led by the [BICA Society](#), but so far these have gained very little traction over the problems central to semantics.

As algorithms increasingly perform in a human-like fashion, the basic architecture dictated by the needs of natural language understanding may be of some interest to the philosopher and the cognitive scientist as well, and [Chapters 3](#) and [9](#) contain much pertinent material. Needless to say, these disciplines have many broader concerns that are out of scope here, and the philosophy student is strongly advised to consult at least [Chapter 16](#) of [Boden \(2006\)](#). The cognitive science student should of course read the entire two-volume set, and the more recent [Gordon and Hobbs \(2017\)](#), not as prerequisites to this book, but for gaining depth. By providing a rather detailed introduction to the technical machinery of contemporary semantics from the ground up, this book is largely complementary to [Boden’s](#), and covers a fair amount that she could not take into account for the simple reason that it has been published since her book was written.

The book is highly sympathetic to the central claims of [embodied cognition](#), but nevertheless approaches matters from a formal symbol-manipulation direction, because the focus is on building algorithms capable of performing semantic tasks such as schematic inferencing (see [Section 7.1](#)), even if this is done at the expense of cognitive realism. The book is for those interested in building flying machines, no matter how birds actually fly, and the only consolation for the cognitive science student is that some of the technical apparatus, the aerodynamics of understanding so to speak, will be of necessity shared between the two. The reading plan presented for the cognitive scientist in [Section 1.5](#) emphasizes this shared aspect.

Typesetting conventions

The book is primarily designed to be read on a computer. We make heavy use of inline references, typeset in blue, particularly to [Wikipedia](#) (WP), [PlanetMath](#), and the [Stanford Encyclopedia of Philosophy](#) (SEP), especially for concepts and ideas that we feel the reader will already know but may want to refresh. Because following these links greatly improves the reading experience, readers of the paper version are advised to have a cellphone on hand so that they can scan the hyperlinks which are also rendered as QR codes on the margin.

As a novel feature, the book comes with an external index starting at page 267 and also accessible at <http://hlt.bme.hu/semantics/external> that collects a frozen copy of the external references to protect the reader against dead links. A traditional index, with several hundred index terms, is still provided, but the reader is encouraged to search the file if a term is missing there. In some cases, a term may be used informally (with or without an inline reference) before we give a more formal definition. The notational conventions used in these diverse sources may not always coincide with the ones used in the book: for example we use $\langle a, b \rangle$ to denote the [ordered pair](#) that Wikipedia would denote by (a, b) .

The diversity of the technical material presented in the book will be somewhat mitigated by a unified methodological outlook. In philosophy and logic it is quite common to approach the matter normatively, simply condemning those forms of usage that the author sees as ‘illogical’, and devising an ideal language that supports only consistent logical use. To a great extent, the normative outlook also pervades computer science, where one is at liberty to define a [formal language](#) by a [formal grammar](#) and attach compositional semantics to it by means of standard software tools such as [yacc](#). Here we are interested in building a workable semantics for natural language expressions (by ‘workable’ we mean simply that it can be used as the basis of writing computer programs) and take actual usage as the primary empirical testing ground of the theory.

The book contains many exercises, mostly rather simple (under level 30 in the system of Knuth, 1971), but often with surprisingly deep implications, like [Schur’s Lemma](#). In many cases, the solutions can be found quite trivially on the web, or even by just reading a few more pages, but readers interested in developing an active knowledge of this field are strongly advised to attack the problem on their own. The goal of some exercises, marked with a raised \circ , is to check the understanding the reader has developed, and the best reading plan is to solve these problems *immediately* as the reader encounters them in the text, rather than waiting until the end of the section or chapter is reached. Other exercises, marked with a raised \rightarrow , point to material that could not be covered in the book, and often rely on additional knowledge, or presuppositions, that render the answer evasive. Still, the reader is best served by trying their hand at these both before and after consulting the hints collected after at the end of the volume. Harder exercises are marked with a raised $*$ and will generally be solved in the text not long after they are posed. In later chapters, we will increasingly mark exercises with a raised \dagger , meaning that there is no unique ‘good’ solution, but the reader should experiment with the problem, primarily



by building a formal or computational model that exhibits the desired properties. The numbering of definitions and exercises is absolute (includes chapter number), to facilitate cross-referencing and checking for hints at the end, but the numbering of tables, figures, and equations restarts in each chapter.

Linguistic examples are normally given in *italics*, and if a meaning (paraphrase) is provided, this appears in single quotes. Italics are also used for technical terms appearing the first time and for emphasis. The `4lang` computational system contains a concept dictionary, which initially had bindings in four languages, representative samples of the major language families spoken in Europe, Germanic (English), Slavic (Polish), Romance (Latin), and Finno-Ugric (Hungarian). Today, bindings exist in over 40 languages (Ács, Pajkossy, and Kornai, 2013). The English printnames of entries in this dictionary, as well as other computationally pertinent material, will be given in `typewriter` font.

Each chapter ends with a section on further reading. Generally, we recommend those papers and books that presented the idea for the first time. Since many of the issues discussed here have decades, and sometimes centuries, of research behind them, this policy may make the book look far more dated than the opposite policy of citing only the latest research would. We think our policy is justified not just by the need to give proper credit, but also because the early works often provide perspective and insight that later discussions take for granted. (A systematic exception is made for monographs and textbooks that have been republished in revised form: these are cited in their latest edition, since these are often better and always easier to acquire.) However, these works are often available only in paper, and fewer and fewer students or scholars are willing to make a trip to the library. Since this trend is clearly irreversible, we make an effort to provide online references, as clickable links, avoiding password-protected portions of repositories like [Project MUSE](#) and [JSTOR](#) as much as possible.

Acknowledgments

Some of the material presented in the early chapters appeared first in three papers, Kornai (2010, 2010a, 2012), and the help of those who commented on some versions of these papers, in particular Tibor Beke [UMass Lowell](#), Zoltán Szabó [Yale](#), Terry Langendoen [UArizona](#), Donca Steriade [MIT](#), Károly Varasdi [Henrich Heine University](#), is gratefully acknowledged. Chapter 8 is largely based on work first presented in Kornai 2014c and Kornai 2014d. We are grateful to Tibor Beke and Peter Vida [Mannheim](#) for trenchant remarks on these.

Work partially supported by OTKA grant #77476 and by the European Union and the European Social Fund through project FuturICT.hu (grant # TAMOP-4.2.2.C-11/1/KONV-2012-0013). Some of the writing was done at Harvard's [IQSS](#), BU's [CS department](#) and [Hariri Institute](#), at the [Algebra department](#) of the Budapest University of Technology and Economics (BUTE), and the [Research Institute for Linguistics](#) of the Hungarian Academy of Sciences (HAS RIL), but the bulk of the work was done at the [Computer Science Institute](#) (HAS CS).



Special thanks to readers of the early versions of this book, who caught many typos and stylistic infelicities, suggested excellent references, exercises, and hints for exercises, and offered penetrating advice on many points evident only to someone with their expertise: Judit Ács [BUTE AUT](#), Eric Bach [Wisconsin-Madison](#), Michael Covington [University of Georgia](#), Gérard Huet [INRIA](#), Paul Kay [Berkeley](#), Marcus Kracht [Bielefeld](#), Márton Makrai (HAS RIL), András Máté [ELTE Logic](#), Imre Orthmayr [ELTE Philosophy](#), Katalin Pajkossy (BUTE), Gábor Recski (BUTE AUT), András Simonyi [PPKE](#), Ferenc Takó [ELTE Philosophy](#), Madeleine Thompson [Empirical](#), and Attila Zséder [Lensa](#). Needless to say, they do not agree with everything in the book, the views expressed here are not those of the funding agencies, and all errors and omissions remain my own.

I would like to single out the Springer proofreader, XYYY, who managed to transform the sometimes over-excited tone of the manuscript to one of scholarly decorum by a more precise placement of commas and by spreading prepositions to each conjunct, and liberated the printed page from the typographical poverty of `ascii`. My editor, Ronan Nugent, is also to be thanked for making the otherwise often painful publication process an enjoyable experience.

The main developers are Gábor Borbély (BUTE), Márton Makrai (HAS RIL), Dávid Nemeskey, (HAS CS), Gábor Recski (BUTE AUT), and Attila Zséder. The material was taught at BUTE twice, Spring 2010 and Fall 2014, and I am grateful to many of the students, including Kara Greenfield (now with [MIT Lincoln Labs](#)), Sarah Judd (now with [Girls who code](#)), and Dániel Vásárhelyi (HAS RIL).

Contents

Preface	v
1 Introduction	1
1.1 Compositionality and contextuality	1
1.2 Selecting the subject matter	3
1.3 Information content	6
1.4 Plan of the book	7
1.5 Suggested reading plans	10
1.6 Further reading	14
2 Linear Spaces, Boolean Algebras, and First Order Logic	17
2.1 Algebras, Boolean algebras	17
2.2 Universal algebra	20
2.3 Filters, ultrafilters, ultraproducts	22
2.4 The propositional calculus	24
2.5 First order formulas	30
2.6 Proof theory	33
2.7 Multivariate statistics	35
2.8 Further reading	45
3 Prolepsis	47
3.1 Understanding	48
3.2 The minimal theory	52
3.3 Space and time	54
3.4 Psychology	59
3.5 Rules	62
3.6 Regularities	65
3.7 The standard theory	70
3.8 Desiderata	75
3.9 Continuous vector space models	79

3.10 Further reading	84
4 Graphs and Machines	87
4.1 Abstract finite computation	87
4.2 Formal syntax	94
4.3 The smallest machines	102
4.4 Graph and machine operations	104
4.5 Lexemes	107
4.6 Inner syntax	112
4.7 Further reading	119
4.8 Appendix: defining words	120
5 Phenogrammar	123
5.1 Hierarchical structure	124
5.2 Morphology	126
5.3 Syntax	133
5.4 Dependencies	143
5.5 Representing knowledge and meaning	150
5.6 Thoughts in the head	153
5.7 Pragmatics	158
5.8 Valuation	165
5.9 Further reading	169
6 Lexemes	171
6.1 Lexical entries	172
6.2 Concepts	175
6.3 Lexical categories	178
6.4 Word meaning	182
6.5 The formal model	191
6.6 The semantics of lexemes	194
6.7 Further reading	197
7 Models	199
7.1 Schematic inferences	200
7.2 External models	203
7.3 Modalities	208
7.4 Quantification	215
7.5 Further reading	218
8 Embodiment	221
8.1 Perception	222
8.2 Action	229
8.3 Adverbials	236

8.4 Further reading 238

9 The meaning of life 241

9.1 Moral philosophy 242

9.2 The empirical basis of moral law 245

9.3 Metatheoretical considerations 250

9.4 A formal model 253

9.5 Summary and conclusions 256

9.6 Further reading 258

Hints for the exercises 259

Solutions for the exercises 260

References 263

Index 263

Index 263

External index 267

Introduction

Contents

1.1	Compositionality and contextuality	1
1.2	Selecting the subject matter	3
1.3	Information content	6
1.4	Plan of the book	7
1.5	Suggested reading plans	10
1.6	Further reading	14

In [1.1](#) we set the stage by introducing the *interpretation relation* that connects linguistic expressions (words, sentences, larger texts) to their meanings, and draw a distinction between the two main parts of semantics, *lexical* and *compositional*. The enormous range of ideas, feelings, thoughts, and facts that natural language can convey makes the task of analyzing the meaning of natural language expressions very broad, and we need to prioritize. This is done in [1.2](#) and [1.3](#) based on frequency of occurrence and information content respectively. After these basics are in place, we discuss the overall plan of the book in [1.4](#), and the reader can then make a more informed choice about the reading plans offered in [1.5](#).

1.1 Compositionality and contextuality

The central idea of semantics, which goes back to Plato's *Theaetetus*, is that to know something is the ability to give an account of its constituent parts. The modern formulation of this idea is standardly attributed to Frege (although, as [Janssen \(2001\)](#) demonstrates, this is something of a mischaracterization of Frege's thoughts on the matter) and is known as the *Principle of Compositionality*:

The meaning of a complex expression is determined by its structure and the meanings of its constituents

What the principle demands is an algorithm that can *parse* an expression, i.e. establish its structure and constituents, and parse the constituents recursively until we arrive at



atomic units. To obtain the meaning of an expression we will need another resource, some kind of wordlist or dictionary, where the meanings of the atomic units are stored: we will call this the *lexicon*. As a first approximation – things will get more devilish as we get to the details – the theory describing the meaning of the atomic units is called *lexical semantics* and the theory describing how to build the larger constructions from the smaller ones (and, conversely, how to parse the larger ones into smaller ones) is called *compositional semantics*.

By *semantic interpretation* we mean some mechanism that computes the meaning of expressions, by *generation* we mean the converse process that starts with some meaning and produces a well-formed expression that has this meaning. The two together are subsumed under a single *interpretation relation* composed of $\langle \text{expression, meaning} \rangle$ pairs. While the theory of semantics is greatly concerned with the technical details of interpretation and generation algorithms, it is worth keeping in mind that the chief practical interest is actually more in the output than in the process, and thus systems that produce wrong output some of the time may still be preferable to ones that only produce correct output at the expense of remaining silent too often. This phenomenon, known as *error-reject tradeoff*, is quite general, not just in semantic systems but also in speech and character recognition, machine translation, and all algorithms dealing with natural language input or output.

Cases of different meanings m_1, m_2, \dots, m_k standing in relation to the same expression e are referred to as *ambiguity*; we say that the expression e is k -fold ambiguous; cases of different e_1, e_2, \dots, e_l standing in relation to the same meaning m are referred to as *synonymy*. Since expressions that totally lack ambiguity are relatively rare, subscripts are commonly applied to distinguish different meanings, as in *chrome*₁ ‘hard and shiny metal’ and *chrome*₂ ‘eye-catching but ultimately useless ornamentation, especially for cars and software’.

When there are infinitely many expressions to consider, as is the case both for natural language and for mathematical expressions, some form of compositionality is inevitable. For example, in an arithmetic example such as $3 \cdot (8 + 1)$, we progress in the order indicated by the parentheses and perform the addition before the multiplication. Most of the time, the parentheses can be omitted, since we have the convention that \cdot binds more strongly than $+$. Since natural language does not offer us the benefit of parentheses (though some forms of sentence intonation come close), the simple act of stringing words together may itself be a source of ambiguity, and we must consider the parse tree a separate information source, one that will decide whether the correct ‘reading’ (as these are called in semantics) should be *(the man on the hill) with the telescope* or *the man on (the hill with the telescope)*. (If we consider a full sentence, *I saw the men on the hill with the telescope*, we obtain yet another reading, since *seeing with a telescope* would also make sense.)

When we try to visualize these cases we notice that *man with the telescope* tends to imply a small hand-held variety of telescope, while *hill with the telescope* conjures up the image of a building that houses a large telescope. The question of whether the evident

size difference justifies talking about *telescope*₁ and *telescope*₂ is one that we defer to Section 6.4, where we argue for *monosemy*, the methodological principle that we should appeal to distinct senses only as a last resort. In general, the phenomenon of the context imposing a specific meaning, what Frege called *contextuality*, is easy to demonstrate in cases where there is no doubt about the ambiguity, as in *pen*₁ ‘writing instrument’ and *pen*₂ ‘enclosed area for children or cattle’. When we say *The box is in the pen* we clearly have *pen*₂ in mind, and when we say *The pen is in the box* it is *pen*₁. Even though an artist like [Christo](#) could in principle box up an entire pen, this possibility does not even enter our mind when we hear the sentence, unless we are specifically [primed](#). Thus we have the *Principle of Contextuality*:

Never ask for the meaning of a word in isolation, but only in the context of a sentence.

For structure-dependent cases, numbering makes little sense as a disambiguation method. Even parenthesization soon loses its grip, because there are important cases when the compositional structure involves discontinuous elements, as in *call her up*, where the parts to be composed are the phrasal verb *call . . . up* ‘telephone’ and the object pronoun *her*. Remarkably, in the [Begriffsschrift](#) Frege (1879) had already noticed this problem, perhaps because it is more evident under German word order, and used special attachment symbols to deal with it. Starting with Bach (1981), modern semantics has developed several methods for dealing with such cases (see in particular Jacobson (2014) Section 5.5 et passim), but a truly perspicuous notation is still missing.

If there are only finitely many expressions to consider, as is the case for example with European traffic signs or Chinese characters, a lexicon alone is sufficient, but even in these cases it may be useful, in particular as a mnemonic aid, to analyze the signs into constituent parts and learn some rules such as that warning signs are triangular while prohibitory signs are circular. But such *redundancy rules* are not compositional, as there are no signs that look like the one in Fig. 1.2 except with a circular border instead of



Fig. 1.1. A non-compositional sign

triangular. One could argue that in principle such a sign could exist, and if it did exist it would mean ‘children prohibited here’, but this is a rather strange kind of evidence to base theories on.

1.2 Selecting the subject matter

Most readers will be familiar with [Zipf’s law](#), which states, in a quantitative form, that



the frequencies of words, pairs of words (called **bigrams**), triples of words (trigrams), and in general word **n-grams** follow a **power law** distribution. What is perhaps less known is the qualitative observation made by Zipf (1935) that the most frequent words are, by and large, also the most basic, elementary units of language, with the longest attested history. Zipf could not quantify this observation because he lacked a formal theory of meaning, but once such a theory is at hand we can use the negative correlation between complexity and frequency as a means of selecting our targets: we must deal with the simplest and most frequent cases first, moving to increasingly more complex and increasingly rare cases only after the simpler and more frequent ones have been taken care of. We must learn to crawl before we can learn to walk, and in this book we will often pointedly ignore the challenges of running and ice-skating until we get the more basic forms of locomotion under control.

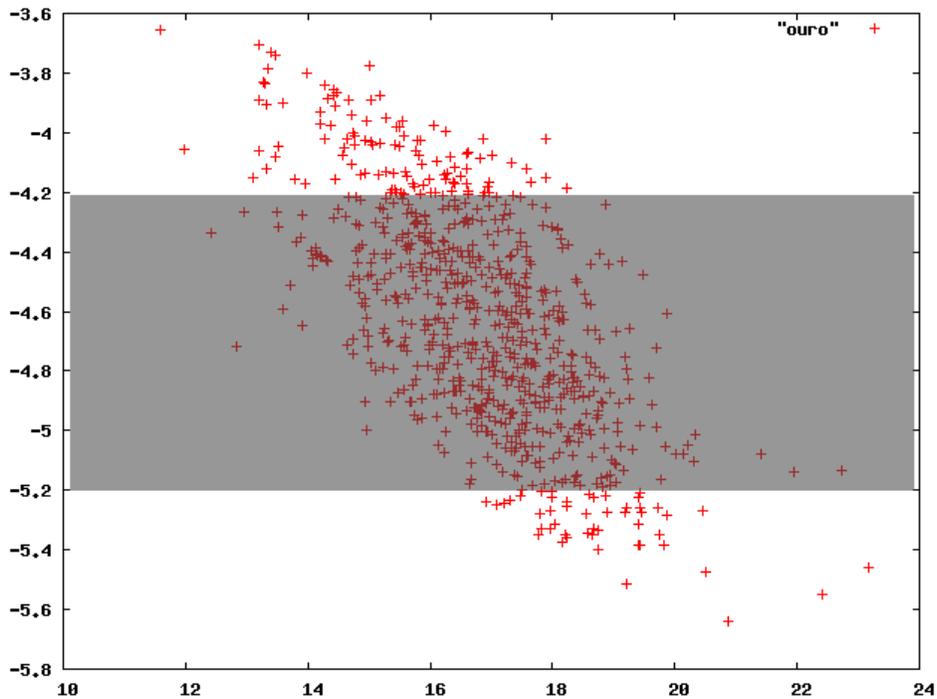


Fig. 1.2. Vocabulary complexity plotted against frequency

Fig. 1.2, only illustrative at this point, plots word complexity, measured roughly by the length of dictionary definitions (see Definition 6.3 on page 178 for a more precise definition) against log word frequency. The words are from the **Longman Defining Vocabulary**, and the frequencies from the **Google 1T corpus**. While the negative correlation between complexity and log frequency is quite visible, to make the quantitative case for the qualitative Zipf law discussed above we will need quite a bit more, both in terms of extending



the graph outside the basic vocabulary domain and in terms of extending the complexity measure from words to constructions – we will deal with both of these problems in the book. The region below the grey band, containing only words with considerable frequency (over 10m) is analogous to what Google calls ‘low pass’ semantics (see [Pereira 2012](#)).

In this band (‘low’ is meant in the complexity, not in the frequency domain), we are anchoring the system in real-world entities and their properties. Currently, the largest publicly available structured collection of such entities, [Freebase](#), has well over 45 million entity nodes (called *topics* there), and about 1.9 billion *facts* stored as labeled edges between the nodes, for example that *Profession(Leonardo, mathematician)*. The questions of how to extract such facts from natural language text, known as [relationship extraction](#), and how to establish whether two linguistic expressions like *Leonardo* and *da Vinci* refer to the same real-world entity, known as [coreference resolution](#), are of central interest to low pass semantics, and are the subject of much current research, facilitated by the availability of standardized data sets and evaluation methods or *shared tasks*. In general, we can get many cases right, perhaps as much as 80%, by rather superficial methods, but it takes quite a bit of world knowledge to figure out that the mathematician Leonardo is Leonardo Fibonacci, not the default Leonardo da Vinci.

In the ‘high’ region above the grey band we also find issues that connect more to *constructions* than to individual words. Since constructions like *Larry a doctor?* (expressing incredulity that Larry could be a doctor) are independent of the elements that make them up, it is hard to measure their complexity, but whatever measure we settle on, the result must be higher than that obtained for isolated words. Issues like temporal reasoning, modalities, quantifiers, and almost everything that takes center stage in academic research, are intimately connected to constructions and constructional meaning, and we will, for the most part, try to avoid these for the following reasons. Examples of high pass phenomena are easy to construct but hard to find in actual natural language data (when was the last time you heard that *at most three professors flunked at least five students in more than four subjects?*), patterns differ widely across languages, and intuitions are often quite uncertain. Some phenomena are robust, but many of the difficulties, for example in quantifier scoping, can only be appreciated by those who have special training.

We will, on the whole, steer clear of the high band, and concentrate on the mid-zone, by presenting a coherent set of semantic techniques that remain usable not just for real-world material but also for fiction, where all the known issues of non-existent entities rear their ugly head. Given his multitude of talents, it is quite possible for some mathematical manuscript of da Vinci to come to light, at which point we would need to add ‘mathematician’ to the already impressive list of professions (painter, sculptor, civil engineer, architect, engineer, anatomist, military engineer, musician, botanist, writer) that Freebase lists for him. He may not *actually* be a mathematician, or at least we do not (yet) know this about him, but he is *potentially* a mathematician, a fact that logical semantics encodes with reference to some [modality](#). It is of course not just the relationships



Phil





between entities, such as the possible ‘profession’ relationship between ‘da Vinci’ and ‘mathematician’, that are subject to modal considerations, but also the [very existence of the entities themselves](#), and in Section 7.3 we will present a system of inferencing that recovers some of the modal ground currently forfeited by Google. We will go beyond the low pass approach in two other respects as well, the interpretation of fragmentary input, and the use of relations that cannot be depicted as graph edges because they involve more than two arguments.

1.3 Information content



Written language has only the words (lexical content) and their order (compositional content) to guide our search for the meaning of an expression, but spoken language offers a rich set of additional cues like tempo, volume, intonation, hand and facial gestures, etc. These are reflected only sketchily, if at all, in the written form, even if we try to employ capitalization and extra punctuation as in *John went WHERE??* to convey incredulity, anger, and the like. Here we will make little distinction between what are traditionally called the ‘emotive’ and ‘connotative’ components of meaning, especially as different linguistic and [paralinguistic](#) means can be employed toward the same purpose, for example a warm tone, a lexical choice like *kitty* instead of *cat*, or simplified syntax can all work to signal the owner’s affection when describing a pet to someone. A neighbor less fond of the same animal will use rather different tone, words, and syntax.



The first question anyone familiar with the basics of [information theory](#) is bound to ask concerns the relative contribution of these factors to the information content of a sentence. The word entropy of natural language is about 12–16 [bits/word](#) (see Kornai (2008) Section 7.1, for how this depends on the language in question), and we know from studies such as Brown et al. (1992) and subsequent work that capitalization and punctuation, our best [proxies](#) for intonation and related factors, contribute less than 7% (0.12 bits of 1.75 bits per character, see Table 3 of Brown et al. 1992). Syntax, as we discussed in Section 1.1 above, is an information source of its own. There are C_n binary [parse trees](#) over n nodes, where C_n is the n -th [Catalan number](#). Because asymptotically $C_n \sim 4^n / \sqrt{\pi n}^{1.5}$, encoding the parse requires less than 2 bits per word. Remarkably, the medieval [Masoretes](#) used only 2 bits (four levels of symbols) to provide a binary parse tree for nearly every Biblical verse (Aronoff (1985) describes in some detail how). What we have learned of coding since would now enable us to create an equally sparse system that is sufficiently detailed to cover *every* possible branching structure with slightly *less than* two bits on average.

Altogether, we conclude that logical structure accounts for no more than 12–16% of the information conveyed by a sentence, a number that actually goes down with increased sentence length, and emotive content for even less, perhaps 5–7%. This back-of-the-envelope calculation is confirmed by everyday experience. As anyone trying to communicate in a language they have mastered only at a tourist level will know, a lack of crisp grammar is rarely a huge barrier to understanding. If you can produce the words, native

speakers will generally be forgiving if the conjugation is shaky or the proper auxiliary is missing. But if you don't have the words for beef stew or watch repairman, knowing that the analytic present perfect combines stage-level and individual-level predication and thus gives rise to an inchoative meaning will get you nowhere.

After computing the meaning (also called the *sense*) of some expression, we often use the result in further computations of a logical nature. When someone says *Sorry I'll be abroad that week* it is not the fact that his body will be physically located outside the country that we care about but the *implication* that he cannot attend the meeting. To get to this implication, we need to rely on some theory of physical objects that contains the restriction that bodies cannot be in two different places at the same time. We will see that one need not bring the full force of modern physics to bear; a far more skeletal theory of *naive physics* (Hayes, 1979) is sufficient. Equally important, when someone says *I'm low on gas* and gets the response *There is a gas station on Main Street*, it is not said that the gas station is actually open. We need to rely on some theory of speaker behavior that says that the information about the location of the gas station is considered relevant to the goal of getting gas right now (Grice, 1981).

We emphasize that conveying information is rarely the primary goal of everyday communication in ordinary language: as members of society we are bound by a large variety of rules and conventions, and simply stating the truth will often be viewed as impolite or downright insulting. Conversely, stating the obvious as in *It's really raining heavily* may serve an important communicative function even though the information thus conveyed is not about the weather but rather about the willingness of the speaker to engage in further conversation. Similarly, the information content of a judge saying *Guilty* is not that the accused is guilty according to some universal standard, but rather that the judge, and by the powers vested in judges, society as a whole, has found the accused guilty, a difference most keenly felt in crimes of conscience. Often, these and similar phenomena are discussed under the heading of *pragmatics*, but in this book we construe semantics broadly so as to include these. This requires the theory of semantics to go beyond the specification of a lexicon and an interpretation relation and include some store of *background knowledge* about physical objects, speakers, and the like, and a specification for drawing inferences.

1.4 Plan of the book

The book is designed for a lecture plus lab course. In the Preface we have already discussed the 'hands on' reading plan that starts with the code and consults the book only as needed, but this is more a statement about the density of the cross-links between the code and the text than an actual study plan. Here we give a bird's eye-overview of each chapter to help the reader select a good study plan – the code has its own top-level [README](#).

Chapter 2 collects the mathematical preliminaries together in one convenient location. Most readers familiar with linear spaces (LSs), Boolean algebras (BAs), and first order logic (FOL) will find only a few new things here, mostly a few odds and ends that



will come handy in Chapter 6. BAs and FOL are central to any understanding of mathematical logic, and offer a unique opportunity to compare and contrast the methods used in explicating mathematical meaning and those required for a treatment of linguistic semantics. We assume the reader to be familiar with the basic notions of sets and relations (for a classic introduction, see Halmos, 1974), and we use standard notation such as \emptyset for the empty set, $\{ \dots \}$ for set, $\langle \dots \rangle$ for ordered tuple, \in for the ‘element of’, and \subset for the ‘subset of’ relation without discussion. We also assume the reader to be familiar with the basic notions of algebra, linear algebra in particular, but to fix the notation and terminology we begin with a highly condensed refresher section. The reader who lacks at least a passing familiarity with abstract algebraic structures like groups, rings, or fields will likely have a hard time with this material, and may want to consult for example Judson 2009 or Dummit and Foote (2003), not so much for the rich material covered in these volumes (only a small fraction of which will we actually rely on) as for general background and motivation. As an introduction to linear algebra we recommend the more leisurely Strang (2009), especially as [lectures are available online](#), or the more densely written Halmos (2013).



Phil

In Chapter 3 we address the problem of what is learned and what is innate. Clearly, when we learn a language we learn not just the generative skills to form words and put them together in ways that express what we want to say but also the interpretative skills of somehow undoing this process and making sense of what others say. In this process of gradually acquiring the meaning of words and the meaning of constructions learners may be aided by innate propensities of various sorts such as [categorical perception](#) or even specific pieces of innate knowledge, for example that languages will invariably be [head-initial or head-final](#) (Chomsky and Lasnik, 1993). The actual amount of innate knowledge in the system is heavily debated, and our goal in this chapter is to present what we believe to be the absolute minimum of what must be presupposed to be innate, and to outline how a well-understood formal framework, [finite state automata](#) (FSA) is capable of carrying this minimum.



In Chapter 4 we consider a class of algebraic structures, the *machines* of Eilenberg (1974), which generalize the better known [finite state transducers](#) and FSA in a manner suitable for semantic purposes. We will pay particular attention to the loose coupling between inner syntax (tectogrammar; see Section 4.6) and outer syntax (phenogrammar; see Chapter 5) that characterizes machines, and begin to describe in rather abstract terms how the tectogrammatical function–argument structure can be encoded in a special class of machines we will call *lexemes*.

Ling

We begin to situate our theory in the space of linguistic theories of semantics in Chapter 5, where we discuss phenogrammar. A full introduction to phenogrammar, syntax in particular, would stretch the plan of any semantics book beyond recognition. Yet no discussion of pheno- and tectogrammar can proceed without an understanding of the basic structure of words and phrases. As a compromise, we will present a somewhat simplified picture of both morphotactics (Section 5.2) and phrasal syntax (Section 5.3), aimed at the computationally oriented reader willing to use the output of morphological and syntac-

tic software packages without actually opening the black box. Broadly speaking, there are two classes of theories to consider, heterogeneous ones such as Montague Grammar (MG) that keep syntax and semantics as substantively different algebraic systems tied together by some homomorphism, and homogeneous ones such as [generative semantics](#) which assume that the meaning of sentences is best expressed by structures which are highly similar to syntactic structures. The theory developed in this book, while retaining some of the characteristics of the heterogeneous approach, in particular the insistence on interpretation in model structures, is fundamentally homogeneous.



In Chapter 6 we return to lexemes and tectogrammar. In developing a formal theory of lexical entries our starting point will be the informal practice of lexicography, rather than the more immediately related formal theories of artificial intelligence (AI) and knowledge representation (KR). Lexicography is a relatively mature field, with centuries of work experience and thousands of eminently usable work products in the form of both mono- and multilingual dictionaries. In contrast to this, KR is a rather immature field, with only a few decades of work experience and few, if any, usable products. We discuss both the way to formalize the informally stated, but nevertheless highly informative, lexical entries used in standard dictionaries and the relationship of machine-based lexemes to modern formal standards such as [OWL](#) and [RDF Schema](#).



Models are discussed in Chapter 7 from the perspective of inferencing and learnability. Standard logic is chiefly concerned with valid inferences that will always lead to true consequences from true premisses. The subjects of natural language discourse rarely lend themselves to the kind of categorical generalizations we express with mathematical axioms, and in everyday conversation our interest is more in persuading the listener than in watertight argumentation. Aristotle already distinguishes [enthymeme](#) from syllogism, and our eventual goal is to capture this notion as part of the formal theory of semantics. While FOL stays within the bounds of consistency and completeness, and in fact is the maximally expressive theory that does so, natural language is clearly more expressive in that it is capable of referring to inconsistent objects, procedures, and situations, and we need a model theory that is capable of carrying this burden. We introduce a *non-involutionary* logic system, 4L, where the negation of negation does not lead back to the original.

Chapter 8 is devoted to embodiment. There are many design decisions that do not follow from the understanding of semantic theory developed in this book and thus appear arbitrary. But from the standpoint of agent-based systems that can plan and act, and draw conclusions about their own plans and acts and those of others, such decisions are severely constrained, especially if we note that these systems can be easily equipped with facilities to replicate, mutate, and to make decisions that cannot be predicted by their creator.

The concluding Chapter 9, modestly entitled ‘The meaning of life’, investigates the meaning of artificial life. Having devoted an entire book to the design of systems that can understand language and produce it meaningfully, we must stop and ask *what for?* The matter is investigated both from the perspective of the creator, whose main concern is



with the [friendliness](#) of the system, and from the perspective of the creature, a viewpoint hitherto largely neglected.

1.5 Suggested reading plans

The default plan. It should be clear from the preceding that there are several reading plans possible based on the goals of the student. The default plan is, of course, to read the book from cover to cover, in the order presented here, doing the exercises as you go along. Based on experience so far, this takes a bit less effort than a two-semester sequence, so that students taking the second (lab) semester actually have a bit of time to do a more exciting individual project, something that seriously contributes to thesis work and/or is good enough to get published at major conferences.

The philosophy plan. It is in the nature of philosophy to ask the hardest questions: What is meaning? How are meanings modeled? Our answer to the first question is that meanings are, roughly speaking, thoughts (concepts, ideas, patterns of neural activation levels) in the head. This has been the received answer from Aristotle (Modrak, 2009) to Locke, and is also taken for granted in much of AI. Chapter 3 presents the view (Nelson, 1982) that patterns of activation in FSA are sufficient for covering all aspects of meaning. Chapters 4 and 6 spell out in great detail how this is to be done for words, which were the primary focus of attention in all philosophy of language before Kant; and Chapter 5 describes this for syntax, where much of the philosophical attention has shifted since Frege and Russell.

How meanings should be modeled is a much more contentious question. There are three broad approaches. First, there is the Frege–Russell–Tarski–Montague–Kamp tradition, what we call the *standard* theory in this book. This is described only briefly in Section 3.7, as there are many excellent introductions, of which we single out Dowty, Wall, and Peters (1981) and Jacobson (2014), aimed more at linguists, and Eijck and Unger (2010), aimed more at computer scientists. Here meanings are modeled by logic formulas. Second, there is the Firth – Harris – Osgood tradition of *distributional* theory, which we begin describing in Sections 2.7 and 3.9. Here meanings are modeled as vectors in a continuous vector space \mathbb{R}^n . Third, there is the [semantic network](#) theory, of which both `4lang` and [Abstract Meaning Representation, AMR](#) are modern instances, which we begin describing in Chapter 4. Here meanings are modeled as hypergraphs.

Until very recently, symbolic models, subsuming both the logic-based standard approach and the more algebra-based network approach, reigned supreme, with the distributional theory relegated to the [information retrieval](#) fringe. In the past 4–5 years this has changed completely. Long-standing hard problems, chief among them the issue of word meaning, but also extending to more syntactic problems such as prepositional phrase (PP) attachment (see Section 4.1) and parsing in general (Chen and Manning, 2014), and even extending to tasks beyond the purely linguistic, such as generating photo captions based on images alone (Karpathy, Joulin, and Li, 2014), are now solved using



continuous vectors obtained from text corpora. This change is now leading to a broad reappraisal of the value of logic in semantics, in that the [correspondence theory of truth](#), and its accompanying [reist](#) model theory, are no longer seen as central to the field. A version of four-valued logic called 4L, and the accompanying model theory, is introduced in Chapter 7.

Finally, students of philosophy may be interested in the solution offered to two classic problems of philosophy, [the Heap](#), and [supererogation](#), both formalized using a new generalization of FSA called Euclidean automata (see Section 8.1) that straddle the discrete–continuous boundary.

The cognitive science plan. The book, for the most part, doesn't address directly any of the major issues of cognitive science like emotions, perception, memory, or comprehension, but asks the question: how can we program a computer to understand when people talk about these and similar things? People's understanding of *my tooth aches* depends to a considerable degree on shared hardware: I don't know how *your* tooth aches but I know only too well how *mine* can ache, and I simply assume that we share the basic condition. Once we have a machine that lacks any part that could hurt, this straightforward sympathetic path to understanding is blocked, and we must rely on a far more abstract approach. First, we need to come to some understanding of what *ache* or *pain* means. Are these really the same thing? When are two things the same? The general issue is discussed under the heading of *synonymy* in Chapter 5, but for illustrative purposes we assume here that they are. Consulting the `4lang` dictionary provides 'bad, sensation, injury CAUSE'.

At first blush, it appears we are in even bigger trouble: how do we know pain is bad? Surely pain is the most important warning system in animals, and the survival prospects of those with [congenital analgesia](#) are very bad, so what is bad locally appears to be quite good globally. Second, even if we enhance the definition somehow to include this important caveat, what is *bad*? Perhaps the Earth is overpopulated, so a lot of people dying painlessly would be good. And who is to say what is good and what is bad, in a given situation, for a given individual or group? Is there a central authority? Should the question be voted on? Fortunately, neither semantics nor cognitive science need concern itself with questions this large: we follow the path of computer science in [separating mechanism from policy](#). Our concern here is with the mechanism alone: as we will see in Sections 3.5 and 5.8, automata states can be *valued* as good or bad, right or wrong, etc. without deciding in advance on a policy of what should be so marked and under what conditions. For example, if we equip our computer with sensors detecting low energy, or physical harm to its parts, we may designate those states where such sensor readings obtain as 'bad'.

The chief contribution this volume makes to the toolkit of the cognitive scientist is precisely this: an abstract theory that connects algorithms to emotions, perception, memory, and, in particular, understanding. Just as the theory of partial differential equations is completely neutral about whether the physicist employs it towards a better understanding of mechanics, heat, or electromagnetism, the theory of finite automata (broadly



construed to include transducers, Eilenberg machines, and Euclidean automata) is just a modeling mechanism. There is no claim that the algorithms actually *feel* these emotions, *have* these perceptions, *remember* things, or *really* understand anything – at any rate, it is unclear how one could go about settling such claims one way or the other. The claim is that we need this much, and no more, to talk about these things in natural language.

Our models of memory (Section 6.1), perception (Section 8.1), and action (Section 8.2) do not aim at a high degree of realism. Take memory: we are concentrating on explicit, declarative, long-term semantic memory at the expense of implicit memories (valuations are typically implicit), and of procedural and short-term aspects of memory, simply because we don't see these aspects as critical for linguistic performance. What we do see as critical is some kind of **learning mechanism**; how this is precisely tied to actual **engrams** is out of scope here, and the same goes for the naive theories of perception and action the mechanism relies on. No doubt these could be improved, and it is clear that the state of the art in cognitive science is already beyond the naive theories described in Chapter 3. Importing the insights of cognitive science is left not as an exercise, but as a lifelong task for the cognitively oriented reader.

The linguistics plan. The book does teach the bare essentials of linguistics that are needed for doing semantics, and readers who don't want to know more can get by with these, provided they are willing to take many things on trust. Students who just want to familiarize themselves with semantics, and instructors who have only a single semester allotted to semantics, will probably want to be very selective about the material in Chapter 2, which will be mostly known to math and computer science majors, and Chapter 5, which will be mostly known to linguistics majors, leaving more time for the less commonly taught material. At the other extreme, those who have already studied linguistics may find the following orientation useful.

We describe a theory of meaning representation that differs significantly from the **logical form** that is standard in linguistic semantics in that it does *not* abstract away from the meaning of the content words. If possession is nine tenths of the law, word meaning is seven eighths of semantics, and both the classic (graph- or hypergraph-based) and the modern (vector-based) theories of word meaning are discussed here in sufficient detail to bring the student to the stage where she can start reading current research. We deemphasize many of the classic logico-philosophical puzzles that take center stage in the MG tradition, and devote the central Chapters 4 and 6 to word meaning, with Chapter 5 discussing the syntax that glues the words together.

Our grammatical theory is, broadly speaking, Pāṇinian: we assume people have ideas in their head and they want to express these so that others will understand them: grammar is a formal transduction mechanism from the ideas (meaning representations) to utterances (strings of words). We take several technical devices from Pāṇini, chief among them the theory of *kāraḱas* or deep cases, which we discuss in Section 4.6, and the mechanism of rule prioritization, whereby the more specific takes precedence over the general. The same directionality, from meaning to form, characterizes **generative semantics**, another theory whose technical devices, in particular meaning decompositions,

we will rely on, together with [case grammar](#), [dependency grammar](#) (see Section 5.4), functional syntax (Foley and Valin, 1984), and a host of other frameworks. While we cannot even go near the entire range of linguistic phenomena covered for example in Dixon (2009), our goal is to present an abstract enough theory that remains compatible with what is known in linguistics at the expense of the more ambitious goal of trying to explain or predict linguistic facts.

The book is greatly influenced by the concerns and methods of [cognitive linguistics](#) – readers of Langacker (1987), Talmy (1988), or Jackendoff (1990) will find many of our themes very familiar. One difference from this school is the insistence on formal rigor: our interest is in things we can explain to a computer, not just those observations we can only explain to our fellow humans. Both the linguistics and the formalism presented here have been part of the mainstream for half a century, often longer. Foundational work on finite automata goes back to McCulloch and Pitts (1943) and Kleene (1956); on Kolmogorov B-complexes to Kolmogorov (1953); and on machines to Eilenberg (1974). Euclidean automata, introduced in Section 8.1, have clear predecessors in classical [circuit theory](#) and in McCulloch (1945). The central novelty of this book, such as it is, consists in bringing these strands of research together. On occasion, this puts some well-known part of grammar, for example the treatment of adverbials (see Section 8.3) in a new light, but on the whole we steer close to the common core of all linguistic theories as befits a textbook.

Some sections are specifically aimed at readers who are already familiar with some part of linguistic theory: Section 5.4 assumes a knowledge of the modern theory of [Universal Dependencies](#), Section 6.5 will be best appreciated by those already familiar with formal grammar, and Section 5.7 is likely to make most sense to readers of Sperber and Wilson (1996).

The engineering plan. We have already mentioned the ‘hands on’ reading plan, perhaps more suited for a lab semester than a lecture semester, where the student just jumps in and reads only as much or as little of the text as is sufficient to make sense of the code. One particularly attractive plan, already promoted in Kurzweil (2012), is to “create a mind”. While it is rather unlikely that [artificial general intelligence \(AGI\)](#) will come about as a result of a term project, building bits and pieces is always a good idea. A necessary, but clearly insufficient condition for building a mind is intelligence, usually equated with the ability to pass the [Turing test](#). To the extent that semantics is viewed as criterial for intelligence by many like Davidson (1990), Chapter 3 is a good place to start. To build a mind will take a bit more: we will also want some kind of life-like behavior, striving to stay alive and accomplish self-imposed goals, what Aristotle called [entelechy](#). This is easier than it looks; see the Exercises at the end of Chapter 4. We also want *free will*, see Chapters 3.4 and 7.3; and some kind of morality, see Chapter 9.

We may also want other things, creativity, curiosity, playfulness, or even the ability to feel lonely (Churchman, 1971). The list of things X proposed in “We cannot have true artificial (general) intelligence until the system is capable of X ” is very long, and there is no way a book on semantics could provide all these ingredients. Some of these, in





particular the ability to act in the real world, are rich subjects in their own right, and the reader interested in these should study [robotics](#) more than semantics. That said, a semantic system still needs the ability to reason about actions, form plans, and reason about the actions and plans of others in the real world, even in the face of imperfect knowledge about the other (automatic or free-willed) agents, and about the world in general.

1.6 Further reading

The definition of *semantics* as the study of meaning is fairly standard, but the definition of *meaning* can vary greatly from author to author, school to school. For an excellent (and at 40 pages still very concise) guide through this terminological thicket, see Chapter 1 of Lyons (1995). We will defer to established usage and speak of *lexical* and *compositional* semantics throughout the book, even though these are simply two parts of one and the same recursive definition. For historical reasons that no longer appear relevant, lexical semantics is still often referred to as ‘cognitive’ and compositional as ‘formal’ semantics, and the term ‘natural language semantics’ is sometimes used exclusively to cover the latter. Here we will carefully avoid these confusing terms and usage, both because lexical semantics, as we shall see, is no less formal than compositional semantics, and because compositional semantics is no less cognitive than lexical semantics – on this point see in particular Partee (1980). For constructions, see in particular Kay (2002). For the early history of [construction grammar](#), see the [Berkeley Construction Grammar](#) website.



The main reason behind the error–reject tradeoff is clear: the more we restrict ourselves to skimming off the top, the better our results will be, and conversely, the more we insist on dealing with the actual variety of cases, the worse our results will be. For example, classical compositional semantics, starting with [Montague Grammar](#), begins with a very small ‘fragment’ of a grammar, and has grown over the years; see in particular the Appendices to Parts I–III and Chapter 19.2 of Jacobson (2014). Today, with allowances for missing lexical items, the fragment accounts for many subtleties, but at the price of leaving uncovered well over 90% of ordinary text such as that of the Wall Street Journal. Low pass semantics now covers perhaps 80% of text at the WSJ level of sophistication, but the analysis is very shallow. Remarkably, for tasks where the correctness of the analysis is a simple yes/no matter (the depth of the analysis is uniform), for example in speech- or optical character recognition, the error-reject curves show not just qualitative but also quantitative agreement; see Hansen, Liisberg, and Salamon (1997).



There is a great deal of material on [ambiguity](#) and synonymy (for the latter see Lyons (1995), Section 2.3), but we will get along with the basic idea of using subscripts for surprisingly long before we have to make finer distinctions. The *pen in the box* example of contextuality goes back to [Bar-Hillel \(1960\)](#), for whom it goes to show the “nonfeasibility” of high-quality machine translation. PP PP attachment is one of the earliest problems in syntactic ambiguity, both in the engineering sense of appearing almost immediately as soon as we try to build a parser that assigns structure to sentences, and in the historical

sense of having been discussed from early on in the literature. Indeed, Hindle and Rooth (1993) already describe the *man on the hill with the telescope* example as “timeworn”, and we can trace it back at least to Simon (1969). Suggestions of using the semantics to disambiguate the syntax, as in Ray Moonie’s example *eat spaghetti with meatballs* v. *eat spaghetti with chopsticks*, go back at least to Marcus’ 1977 MIT thesis, published as Marcus (1980) and quite possibly earlier. Until the mid-1990s PP attachment was seen as something of a hard test case for any proposal about semantics, overshadowed only by the fact that we discovered a large number of even worse problems. Little progress was made until very detailed lexical resources were brought to bear (Bailey, Lierler, and Susman, 2015) and continuous vector space models (see Section 3.9) were applied (Belinkov et al., 2014).

There is a large literature surrounding Zipf’s Law – for an introduction, consider Kornai (2002), Mitzenmacher (2004), or Section 4.4 of Kornai (2008). For information content, as measured in bits, Section 7.1 of Kornai (2008) offers a brief introduction. Comparing the relative information content carried by lexical, syntactic, and extralinguistic means originates with Kornai (2010a). In one very characteristic view, put forth in Gazdar (1979), pragmatics is a wholly separate field of study from semantics. Others, most notably Sperber and Wilson (1996), take the view that pragmatics is part and parcel of semantics, and this is the view we take here. Grice’s approach has been enormously influential, and is set forth with great clarity in his numerous works – for a more pedagogical introduction, see Levinson (1983).

Whether the Turing test really measures intelligence is much debated; we recommend the discussion in Shieber (2007). A new test set, more demanding from the natural language understanding standpoint, is presented in Levesque, Davis, and Morgenstein (2012); this will be discussed in Chapter 7. For more on entelechy, see Kornai (2015). The idea that some form of morality is necessary for AGIs is standard; see for example Wallach and Allen, 2009.



Linear Spaces, Boolean Algebras, and First Order Logic

Contents

2.1	Algebras, Boolean algebras	17
2.2	Universal algebra	20
2.3	Filters, ultrafilters, ultraproducts	22
2.4	The propositional calculus	24
2.5	First order formulas	30
2.6	Proof theory	33
2.7	Multivariate statistics	35
2.8	Further reading	45

After a brief introduction to algebras, in [2.1](#) we begin with linear spaces (LSs) and Boolean algebras (BAs). In [2.2](#) we cover the basic notions of universal algebra, building up to, but not including, [Birkhoff's Theorem](#). Ultrafilters are introduced in [2.3](#). In [2.4](#) we turn to the propositional calculus, and the (lower) predicate calculus is discussed in [2.5](#). The elements of proof theory are sketched in [2.6](#), and some preliminaries from multivariate statistics (which are, for the most part, just linear algebra in new terminological garb) are discussed in [2.7](#). The material is selected for its utility in later chapters, rather than for internal cohesion, and is largely orthogonal to the standard logic prerequisites to compositional semantics covered in for example the two volumes of Gamut ([1991](#)). A first course in linear algebra (but not in multivariate statistics) is assumed.



2.1 Algebras, Boolean algebras

In general, an *algebraic structure*, or *algebra* for short, is built from objects belonging to various *sorts*. For example, power series over a ring will involve both elements of the ring (known as *scalars* in this setup) and *indeterminates* X, Y, \dots and their powers and formal products X^2, X^3, XYX, \dots . In all cases, we will collect the sorts into a denumerable (typically, finite) list $S = \{s_1, s_2, \dots\}$. In many important cases, such as (semi)groups, (semi)rings, fields, and Boolean algebras, the notation is overdesigned since we will have just one sort. However, there are enough important other cases, for

example transformation (semi)groups, where it is really helpful to have separate sorts. The central example of a multi-sorted system that we will deal with in this book is a *linear space*, or *vector space* which has two sorts: *vectors* coming from an Abelian group V , and *scalars* coming from a field F , typically the field of real numbers \mathbb{R} or the two-element finite field $\mathbf{GF}(2)$. In some cases, like string rewriting systems, we actually have a bit of freedom as to whether we wish to declare all objects as belonging to the same sort or we wish to treat for example terminal and nonterminal symbols as separate sorts. For each sortal type s_i , the objects belonging to that sort are collected in a *base set* (often called the *universe*) U_i .

In addition to the elements of various sorts, an algebra will have two more things: *operations* and *relations*. The typical operation is primarily defined through some *arity* $\langle s_1, \dots, s_n \rangle \rightarrow s$, meaning that if o_1, \dots, o_n are objects of sorts s_1, \dots, s_n respectively, the result of the operation is an object of sort s . For example, in a linear space the scalars operate on the vectors by an operation of *multiplication* that assigns to each scalar λ and each vector \mathbf{v} their product vector $\lambda\mathbf{v}$ (equal, by definition, to $\mathbf{v}\lambda$ so that the issue of whether this product operation commutes does not even arise). The arity of this ‘multiplication by scalar’ operation is $\langle F, V, V \rangle$, quite different from the unluckily named ‘scalar multiplication’, better called the *inner product* or *dot product*, which has arity $\langle V, V, F \rangle$. *Vector addition* is simply the group addition of V , with arity $\langle V, V, V \rangle$. *Vector multiplication* or the *cross product* has the same $\langle V, V, V \rangle$ signature as vector addition, but it is peculiar to 3D space, while vector addition exists in any dimension.

If there is only one underlying sortal type, the arity reduces to the number n . Later we will consider operations that are undefined for some combinations of inputs (think of division by zero in a field), but we begin with the plain case when no such exceptions exist. For ease of presentation, we will often treat operations as special kinds of relations with signature $\langle s_1, \dots, s_n, s \rangle$, where the last member in any relational tuple is uniquely determined by the others: if R is an $n + 1$ -ary relation signifying the result of an n -ary operation O for which $R(o_1, \dots, o_n, o)$ and $R(o_1, \dots, o_n, o')$ both hold, $o = o'$. In other words, an n -ary operation is an $n + 1$ -ary relation that behaves as a *function* of n arguments that assigns a unique value.

The *signature* of an algebra is the enumeration (again a denumerable, typically finite, and in fact very short list) of the pairs $\langle o_i, a_i \rangle$, where o_i is the operation and a_i is its arity. We add in *constants* in the form of nullary operations: no input is required and the output is always the same element. For example, **groups** have signature $\{\langle \cdot, 2 \rangle, \langle ', 1 \rangle, \langle e, 0 \rangle\}$, where the binary operation \cdot is the group multiplication, the unary $'$ is reciprocation, and the nullary e is the unit element of the group; **rings** have signature $\{\langle \cdot, 2 \rangle, \langle +, 2 \rangle, \langle ', 1 \rangle, \langle -, 1 \rangle, \langle 1, 0 \rangle, \langle 0, 0 \rangle\}$, where the two binary operations are multiplication and addition, the unaries are multiplicative and additive inversion, and the nullaries are the multiplicative and the additive units. In linear spaces, we must combine the signature of the field F , the signature of the group V , and the signature of the multiplication-by-scalar operation to obtain the full signature.



In all algebras, regardless of signature, we also need at least one distinguished operation I , the unary *identity* operation, which outputs its input. Viewed as a relation, I becomes a binary relation, denoted $=$ and called *equality*. The primary reason for keeping equality outside the sortal framework is that we need it to state the axioms, such as associativity or distributivity, that define groups, rings, and algebras in general. There are other relations, besides equality, which often recur in algebraic structures, most importantly (partial) orders, denoted as usual by $<$ or \leq , $>$ or \geq , depending on whether the ordering is strict.

Exercise^o 2.1 Define the *transitive closure* of a binary relation T as the smallest relation T^* under set-theoretical containment that contains T and is transitive (aT^*c follows from aT^*b and bT^*c). Are partial orders, as binary relations, always the transitive closure of some unary operator?

Exercise^o 2.2 (Leibniz) Can equality be uniquely defined if we are given an ordering (partial or full, strict or not)? Can ordering be uniquely defined if we are given a relation of equality?

Definition 2.1 The signature of *Boolean algebras* contains two binary operations \wedge, \vee ; one unary operation \neg ; and two nullary operations \top, \perp . To get what we need we also need some *axioms* connecting these: the laws of associativity, $(a \wedge b) \wedge c = a \wedge (b \wedge c)$, $(a \vee b) \vee c = a \vee (b \vee c)$; the laws of commutativity, $a \wedge b = b \wedge a$, $a \vee b = b \vee a$; the laws of absorption, $a \vee (a \wedge b) = a$, $a \wedge (a \vee b) = a$; and the laws of negation $a \vee \neg a = \top$, $a \wedge \neg a = \perp$. We also need axioms connecting the nullaries to each other, $\neg \top = \perp$, $\neg \perp = \top$; plus the axiom that \vee distributes over \wedge , $(a \vee b) \wedge c = (a \wedge c) \vee (b \wedge c)$.

Notation 2.1 The smallest Boolean algebra has only the two distinguished elements \top, \perp – it is trivial to check that these satisfy the axioms given above. We will denote this BA by \mathbb{B} .

One salient point, illustrated for BAs here, but quite true in general, is that there is no *unique* axiom system describing these structures. For example, we could equally use the above laws plus the axiom that \wedge distributes over \vee , $(a \wedge b) \vee c = (a \vee c) \wedge (b \vee c)$, to define the exact same BAs. We could have replaced the laws of negation given above by the laws of identities $a \vee \perp = a$, $a \wedge \top = a$.

Exercise^o 2.3 Prove the above statement by showing that distributivity of \wedge over \vee follows from distributivity of \vee over \wedge given the associative, commutative, absorption, and negation laws. Do you need to prove the other direction?

Exercise^o 2.4 Do real numbers between 0 and 1 fulfill the BA axioms if $x \vee y$ is defined as $\min(x, y)$, $x \wedge y$ is defined as $\max(x, y)$, $\neg x$ is defined as $1 - x$, \top is 1, and \perp is 0?

For those more familiar with rings, it can be helpful to know that Boolean algebras can be converted to *Boolean rings* by defining the usual ring multiplication $s \cdot t$ as $s \wedge t$ and the ring addition $s + t$ as $(s \wedge \neg t) \vee (\neg s \wedge t)$, i.e. as the symmetric difference of s and t .

Exercise^o 2.5 Prove that in those rings which are obtained from Boolean algebras by the above definition every element is *idempotent*, i.e. it satisfies $s^2 = s$. Given some

ring where every element is idempotent, and taking $s \wedge t$ to be st , is there a way of constructing some Boolean operation \vee from ring multiplication and addition such that this newly defined \vee will distribute over \wedge ? Can you define \neg based on \cdot and $+$ so that the resulting structure will satisfy all the axioms of Boolean algebras? If you convert a Boolean algebra B to a Boolean ring R , and convert the result back to another Boolean algebra B' , will $B = B'$?

A very significant generalization of Boolean algebras is obtained by considering Boolean-like structures which lack complementation and may not be fully distributive. Given some set U , a *partial order* on U is defined as a relation \leq satisfying the reflexive, transitive, and antisymmetric properties $s \leq s$; $s \leq t, t \leq u \Rightarrow s \leq u$; and $s \leq t, t \leq s \Rightarrow s = t$. If s and t are elements of a Boolean algebra B , we can define $s \leq t$ to hold iff $s \vee t = t$ or, equivalently, iff $s \wedge t = s$. Thus, every Boolean algebra gives rise to a partial order, but the converse is not true; there are many partial orders that cannot be obtained from, or converted back to, Boolean algebras. We define a **lattice** as a structure with two binary operations \vee and \wedge that satisfy the commutative, associative, and absorption laws.



Exercise \rightarrow **2.6** Provide an example of a lattice L for which adding a unary operation \neg that would turn it into a Boolean algebra is impossible. Provide an example of a partially ordered set that cannot be turned into a lattice.

2.2 Universal algebra

Phil One of the central goals of algebra is to *understand* the structures it studies. ‘Understanding’ is a pre-theoretical notion, and it is not entirely evident what we mean by it in this context, but the everyday concept of understanding involves both *decomposition*, finding simpler parts that we feel we understand better, and *manipulation*, knowledge of some properties that enables dealing with the object in an efficient manner. In mathematics, the ability to rapidly compute things is considered a hallmark of understanding, and knowledge of some properties will be judged important to the extent it is useful in speeding up computations. Here we discuss some methods of decomposing and manipulating algebras which are applicable to many algebraic structures, not just Boolean algebras, but also groups, rings, lattices, and so on.

Given some universe U endowed with some operations o_i of the appropriate signature, a subset $V \subset U$ can be *closed* under these operations in the sense that if all inputs to the operation are taken from V , the output will also be in V . For example, real-valued real functions form a ring under the usual definition of function addition and multiplication, and the set of *even* functions satisfying $f(x) = f(-x)$ forms a subring. In general, a *substructure* is a subset of the original structure closed under all operations, including the nullary ones, if present. Not only are we interested in finding the substructures of a given structure, we are also interested in recognizing a given structure as a substructure of a larger, but perhaps better understood, structure.

Given a number of algebraic structures S_1, S_2, \dots of the same sort, we can always form their *direct product* $S = \prod_{i \in I} S_i$ by taking the base set to be the Cartesian product of the base sets of the S_i and performing operations coordinatewise. This method extends to the case when the collection of structures indexed by I is infinite. A *subdirect product* is a subalgebra S' of the direct product S that spans all coordinates, i.e. a subset $S' \subset S$ that is closed under the operations and satisfies $\pi_i(S') = S_i$ for all $i \in I$, where π_i is the i -th *projection* (a mapping that discards all coordinates other than the i -th). Knowing that some structure is a (sub)direct product enables a high degree of parallelization: to compute the result of some operation o it is sufficient to compute the result of the operation at each coordinate, and this can be done in a parallel fashion.

Exercise^o 2.7 Let V be a three-dimensional linear space over the reals, and W be a 27-dimensional linear space over the complex numbers. Is V a substructure of W ? Can you form the direct product $V \times W$?

Mappings between structures that commute with all operations are known as *homomorphisms*. A particularly important case is provided by invertible mappings that are homomorphisms in both directions – these are known as *isomorphisms*. Algebraic structures are only investigated up to isomorphism – we make no distinction between isomorphic objects. However, *learning* about the existence of an isomorphism is often considered a significant source of knowledge, for example, if we learn that T is isomorphic to a direct product of some well-understood S_i , we have thereby acquired a method for organizing computation in T in parallel, something that we didn't have before.

Exercise^o 2.8 Consider the plane as a two-dimensional linear space over the reals. Now consider it as a one-dimensional linear space over the complex numbers. Are these spaces isomorphic?

Another important method of reducing the complexity of an algebraic structure is by congruence relations. An *equivalence relation* is any reflexive, symmetrical, and transitive relation. If o is some n -ary operation of the structure S and for all n -tuples $\langle s_1, \dots, s_n \rangle$ and $\langle t_1, \dots, t_n \rangle$ such that $s_i \equiv t_i$, we can conclude that $s = o(s_1, \dots, s_n)$ and $t = o(t_1, \dots, t_n)$ satisfy $s \equiv t$, we say that the equivalence relation \equiv *respects* (sometimes called 'is compatible with') the operation o . We call an equivalence a *congruence* of an algebraic structure if it respects all the operations. If \equiv is a congruence of S , we can define the *quotient structure* S/\equiv by taking the elements of this structure to be the equivalence classes and taking the operations among these to be defined by the equivalence class of the result of performing the operation on arbitrarily selected members of the classes – since \equiv is a congruence, it makes no difference which members are selected. Taking the quotient of S by a suitably selected congruence \equiv will always result in a structure that is no more complex, and typically far less complex, than S was.

The notions of substructure, homomorphic image, and quotient are strongly related. If f is a homomorphism from S to T , the *image* of S under f will be a substructure T' of T , the relation \sim defined by $a \sim b$ iff $f(a) = f(b)$ will be a congruence, and we have $S/\sim = T'$, where '=' is used, as is common in algebra, not set-theoretical identity, but isomorphism.

Exercise[◦] 2.9 Prove the above statement, known as the First Isomorphism Theorem.

In many cases of great practical importance, such as groups, rings, fields, and linear spaces, it is possible to replace $f(a) = f(b)$ by $f(a) - f(b) = 0$ or $f(a)/f(b) = 1$. When f is a homomorphism, this means that $f(a - b) = 0$ or $f(a/b) = 1$ can be used to capture the congruence, or in other words, the structure that maps on the distinguished (additive or multiplicative) unit, called the *kernel* of the relation, is sufficient to capture the entire relation.

Some care must be taken, since the set of elements that is mapped onto the unit by a homomorphism will always be a substructure, but mapping a random substructure on the unit may not be extensible to a full homomorphism that also has non-unit members of T in its range. This is because the unit is commutative with everything even if the structure itself is not commutative, so that for example in groups if $f(a) = e$ and $f(b) = x$ holds, we must also have $f(b'ab) = f(b')ef(b) = x'ex = e$. In other words, to be a kernel of a homomorphism implies not just closure under the multiplication, reciprocal, and unit operations, but also closure under the *conjugation* operation $b'ab$, where a is in the kernel but b may lie outside it.

Exercise[◦] 2.10 Consider the group of rigid transformations that map an equilateral triangle onto itself. Using function composition as multiplication, these transformations form a group. How many elements does this group have? Take a reflection r about a fixed median plus the identity transform e ; do these form a subgroup? Does this subgroup characterize a homomorphism?

2.3 Filters, ultrafilters, ultraproducts

Given some nonempty base set U (often called the *universe*), subsets of 2^U are referred to as *systems of sets*. We say that a nonempty system \mathcal{I} of sets has the *finite intersection property* iff for every finite subset \mathcal{J} of \mathcal{I} we have $\cap \mathcal{J} \in \mathcal{I}$. We call a set of sets \mathcal{U} *upward closed* (or *upper* for short) if for every $X \in \mathcal{U}$ and $U \supset Y \supset X$ we have $Y \in \mathcal{U}$.

Exercise[◦] 2.11 Given an infinite base set U , define \mathcal{S} as the set of those subsets of U which have only finitely many elements. Does \mathcal{S} enjoy the finite intersection property? Is it upward closed? Define \mathcal{B} as the set of those subsets of U that have at least two elements. Is \mathcal{B} closed under finite intersections? Is it upward closed?

Let us define a *filter* \mathcal{F} as an upward closed system of sets that does not contain \emptyset but is nevertheless closed under finite intersection. (Were we to permit the empty set in \mathcal{F} , closure under intersection would be too easy to satisfy.) A good example is provided by those subsets of some larger base set U that contain some nonempty $V \subset U$ – these are known as *principal filters*. A more exciting example is the so-called *Fréchet filter*, which is formed from an infinite base set U by taking all those subsets $V \subset U$ whose complement is finite. Clearly, if X and Y are two such sets, the complement of their intersection is the union of their complements, and the union of finite sets is again finite.

We define an *ultrafilter* as a maximal filter with respect to set-theoretical containment, i.e. a system of sets to which no further set could be added without destroying the properties that make it a filter. Principal filters over singleton sets are important examples of ultrafilters both over finite and over infinite base sets, but over infinite base sets they are not the only examples (for finite U see Exercise 2.14 on p. 24). To see that the ultrafilter definition can be satisfied we use the algebraists' homebrew version of the Axiom of Choice, known as **Zorn's Lemma**: if in some partially ordered set P every chain (totally ordered subset) has an upper bound, P has a maximal element. This is highly applicable in algebra, because we generally take the partially ordered set to be the set of substructures of some structure, ordered by inclusion, and in this setup the union of chains will always itself be a subalgebra.



Exercise° 2.12 Why?

Now we apply Zorn's Lemma to the case at hand, constructing a maximal filter. Given an ascending chain of filters \mathcal{F}_i , their set-theoretic union, that is, the set \mathcal{F} that contains all sets that appear in at least one \mathcal{F}_i , will again be a filter, since (i) as the original \mathcal{F}_i did not contain the empty set, neither will \mathcal{F} , and (ii) if we take the intersection of two sets X and Y in \mathcal{F} , one belongs in some \mathcal{F}_i and the other in some \mathcal{F}_j by the definition of union, and we have either $\mathcal{F}_i \subset \mathcal{F}_j$ or $\mathcal{F}_i \supset \mathcal{F}_j$ since both are part of the same chain, so one of these two will contain both X and Y , and since that one is a filter, it will contain their intersection as well. Now, if $X \cap Y$ is in some \mathcal{F}_i , it is by definition also in \mathcal{F} , so \mathcal{F} is indeed closed under intersection.

Discussion There are several important schools of mathematical philosophy that reject the Axiom of Choice (AC). The issue is that the results obtained by use of the AC are not entirely constructive. For example, we know that the Fréchet filter \mathcal{F} over the set of integers can be extended to an ultrafilter \mathcal{U} simply by considering the set of all filters that contain \mathcal{F} – here all chains have an upper bound, so there exists at least one maximal element, perhaps several. Yet we have a hard time pinning down exactly which sets are members of such a \mathcal{U} : for example, will the set of even numbers be a member? What we emphasize in this regard is not that the AC is somehow 'more true' than these philosophers regard it, or even that it is more useful, but rather the simple fact that understanding something is not the same as endorsing it. What we prove here is that *if* the AC holds so will the theorem that guarantees that every filter can be extended to an ultrafilter. This is not different from any other mathematical theorem, since every theorem is contingent on some axioms. We do not have to make any commitment as to whether Euclidean geometry is more or less 'real' or 'true' than non-Euclidean geometry; in fact, this does not matter at all for the truth or reality of the various theorems – what matters is to understand which axioms a particular theorem requires.

Phil

Exercise° 2.13 Let \mathcal{U} be an ultrafilter over some set B . Prove that for every $X \subset B$ either X or $B \setminus X$ is a member of \mathcal{U} . Prove the converse, namely that a system of sets over B that does not contain \emptyset , is closed under finite intersection, and contains every set X or its complement $B \setminus X$ is an ultrafilter.

Starting with the Fréchet filter, (ultra)filters formalize the intuition of *generally*. Clearly, if some statement is true in general, over an infinite set, a few (finitely many) exceptions do not matter. When we say that n th-degree polynomial equations cannot be solved in general, the fact that for $n = 1, 2, 3, 4$ they are solvable is negligible compared with the general truth that for $n = 5, 6, \dots$ they are not.

Since we are more interested in general rules than in a few exceptions, the use of ultrafilters is particularly widespread in logic, where they serve as the basis of the *ultraproduct* construction, to which we turn now. Given an infinite number of structures S_1, S_2, \dots of the same sort, indexed by some set I , suppose \mathcal{U} is an ultrafilter over I . Elements of the ultraproduct S/\mathcal{U} are defined as equivalence classes of the elements of the direct product S : two such elements (s_1, s_2, \dots) and (t_1, t_2, \dots) are equivalent if $s_i = t_i$ generally, i.e. if the set of indexes where this equality holds is in \mathcal{U} . We will say that a property holds *almost everywhere* in I if its characteristic set is in \mathcal{U} – the terminology is taken from that used for sets of measure 1 in a measure space. Operations on elements of the ultraproduct are performed coordinatewise, and it doesn't matter if at a few coordinates the results are undefined as long as they are defined almost everywhere.

What makes this construction particularly attractive is that all elementary relations can be meaningfully lifted to the product as a whole. Suppose, for example, that each component structure S_i is endowed with some relation R_i , for example an ordering relation. As is well known (by [Arrow's Impossibility Theorem](#) and the strongly related [Condorcet voting paradox](#)), there is no overall relation that will aggregate these orderings into a single coherent relation as long as there are only finitely many structures to begin with and we insist that the overall result is not simply the ordering on one of these.

Exercise^o 2.14 Prove that over a finite set, every ultrafilter is a principal ultrafilter. Show that the Impossibility Theorem follows from this.

In the infinite case, ultrafilters will average out n -ary relations R_i in the following sense. Let $\langle s_1^i, \dots, s_n^i \rangle$ be an n -tuple of elements of S_i . Either $R_i(s_1^i, \dots, s_n^i)$ holds in S_i or it does not – we collect the indexes $j \in I$ for which it does in the set $J \subset I$. For any given ultrafilter \mathcal{U} over I , either $J \in \mathcal{U}$ or its complement $I \setminus J$ is in \mathcal{U} (Exercise 2.12). In the former case we define the relation R to hold over the n -tuple of elements s_1, s_2, \dots, s_n in the direct product; in the latter case we define it not to hold. In the direct product, an element s_k is of course an infinite sequence $s_k^1, \dots, s_k^i, \dots$ ($i \in I$), and the existence of such elements, called *choice functions* (because we choose one element of S_1 , one element of S_2 , and so forth), is guaranteed only by the Axiom of Choice. In short, the relation R will hold in the ultraproduct $S = \prod_{i \in I} S_i/\mathcal{U}$ iff it holds in almost every component.

As we shall see shortly, the same method of taking a majority vote (requiring almost all indexes to exhibit the required behavior) will work not just for any single relation, but for any finite Boolean combination of any relations, including the base identities, such as commutativity and associativity, that we use to define our structures in the first place. In fact, we can go beyond Boolean combinations and consider all formulas that can be created by quantification, both universal and existential, over elements of the universe



– this will be the *Łoś Lemma*. Since this notion of logical formulas will play a crucial role in what follows, we will take some time to develop in detail what appears at first sight a rather clumsy recursive definition, beginning with the notion of the *elementary statement* or *proposition*.

2.4 The propositional calculus

At a pre-theoretical level, we feel rather strongly that certain statements such as *Ice is cold* or *Lung cancer has no cure* are true. The first one is true by virtue of what it means: by definition, *ice* is ‘frozen water’ and ‘below freezing point’ is part of the temperature domain we call *cold*. Following Kant, we will say that such statements are *analytic*. The second is true in a weaker sense: it does not follow from the definition of *lung cancer* or from the definition of *cure* that the former lacks the latter, it is just an observational fact about the world as we know it, more reflective of the state of the healing arts in the 2010s than of some inherent properties of ‘uncontrolled cell division’ (cancer) or ‘restoring normal functioning’ (cure). We will never encounter a situation when the first sentence becomes false, unless we somehow manage to change the meaning of *ice* or *cold* (or perhaps the meaning of *is*), while we may very well see some cure for lung cancer without changing what we mean by *lung*, *cancer*, or *cure*. Kant used the term *synthetic truth* to describe statements of this weaker kind. There are strongly related notions, such as *a priori* and *a posteriori* (given v. observed), and *necessary* and *contingent truth*, which we will return to in Chapter 3 and again in Chapter 7, but for the moment we will abstract away from these finer distinctions and concentrate on the major division between those expressions that *can be* true or false as they are, and expressions which *cannot be* considered true without providing further information.

The first kind, called *truth-bearers*, are defined to include not just true but also false statements such as *The Sun is cold*. Examples of the second kind include $x < y$ or *Tom hates Bill*, which require some specification of which x and y , which Tom and Bill, are meant. In logic, statements that require no further specification are called *propositions* or *closed sentences*; those that do are called *open sentences*. One simple diagnostic to tell the two apart may be the presence of variables such as x or y (later, as we introduce variable binding, we will modify this diagnostic to say *unbound* variables). That the actual situation must be a bit more complex can be seen both from examples like $x^2 \geq 0$, which contains a variable but is nevertheless true among the reals, and *Tom hates Bill*, where we intuitively feel that neither *Tom* nor *Bill* are truly variable – in any context, we are likely to have some definite Tom and Bill in mind, and the identity of these participants cannot be changed by shifting the context. A somewhat similar problem is seen in simple sentences like *I am hungry*, whose truth depends both on the person who utters them and on the specific time when they make the utterance. There are several technical methods that address these problems – in Section 3.10 we consider *fluents*, variables whose truth is a function of time, and in Section 7.3 we consider *indexicals*,



expressions such as pronouns *I, you, ...* or *here, now, ...* which point to different people or times depending on context.

In mathematics, variables are used for two strikingly different purposes: on the one hand, they are used to denote quantities that vary, typically with a change of time or space coordinates, and on the other, they are used as placeholders, to be manipulated just as ordinary elements, at least until their exact value is determined. The former usage is typical of analysis, the latter of algebra, and it is not at all obvious that the two notions, varying quantities and unknown quantities, can be subsumed under the same heading. What really makes this possible is that variables are not actual objects that truly share the sortal type t of the universe U they are taken from, but rather are cleverly disguised functions from some base B to U . When we speak of variables in the analytic sense, the base set is generally \mathbb{R} , the set of reals, or \mathbb{R}^n , Euclidean space. When we speak of variables in the algebraic sense, the base set is some singleton set, which we will denote by 1. The power of the idea comes from the fact that such functions, by defining operations pointwise, can be made to fit into the same structure that U has. If f and g are, say, positions of particles (vectors) that vary with time, their pointwise sum $(f + g)(t)$, defined as $f(t) + g(t)$, will again be a vector that varies with time. If x and y are unknown real numbers, so is $\sqrt{x^2 + y^2}$ or any other expression formed by any sequence of operations that can be performed on reals. (Some care must be taken to ensure that the operation can really be performed; for example, $x/(y - y)$ will be undefined, but this will not affect the definition unduly.)

Discussion While the notion of variables is highly intuitive, it would be a mistake to assume that variables are in any way necessary to develop mathematics – even the most analytic parts of mathematics, with functions, derivatives, integrals, etc. can be recast in a variable-free notation (Givant, 2006).

While our immediate concern is with propositions (closed formulas), it will be expedient to build up the languages of closed and open formulas together. If we are given some structure S over some base set U , we define an *atomic expression* (also called an *atomic term*) either as a member of S or as a variable over S . Atomic terms have complexity 0, and if we collect enough of them to perform some operation with these as arguments, the result is considered a term of complexity 1, and so forth. In general, if s_1, \dots, s_k are terms of complexity c_1, \dots, c_k , the result of a k -ary operation is a term of complexity $\max(c_1, \dots, c_k) + 1$. The collection of terms obtained in this way is called the *term algebra* or *free algebra* generated by the variables. One case of particular interest is when the algebra is a **monoid** (one associative binary operation with a nullary unit) – in this case the variables are called *letters* and are collected into an *alphabet*, and the elements of the free monoid are simply the *strings* (finite sequences of letters, also called *words*) that can be formed from these.

Members of S are often called *constants* in logic, but this name is justified only in comparing them to variables, because in a larger sense only the distinguished elements of S , and those elements that can be built from these, are truly constant. For example, in a ring R there is a unit element 1, and even when our knowledge of R is imperfect (we



may not know *which* of several rings is exactly the one under consideration) we know that this element will behave as a multiplicative unit. Once we have 1, we also have $1+1$, $1+1+1$, $1+1+1+1$, and so on, but it is a stretch to denote these by $2, 3, 4, \dots$, in that the ring axioms do not guarantee all these to be distinct.

An *atomic formula* is some relation whose terms are expressions, not necessarily atomic. For example, if our structure is the ring of complex numbers \mathbb{C} , $3+2i$ and $z+2w$ are atomic expressions, and $3+2i = z+2w$ is an atomic formula. Complex formulas, generally called *well-formed formulas*, or *wffs*, are formed from atomic formulas by Boolean operations, with parentheses denoting the order of operation. These will again have a complexity based on the depth of the parenthetization required, and there will be a somewhat tedious assertion, known as the *Truth Theorem*, that these behave as expected.

Truth Theorem If a set T of closed formulas given in first order language has a model, T is consistent (one cannot derive both a formula and its negation).

Exercise^o 2.15 Define the depth of wffs inductively, use this to prove the above theorem.

The propositional calculus deals with those wffs that contain no variables at all. That said, we will still find it expedient to use variables, but these will range over the atomic formulas (rather than elements of the universe). We begin with a collection, not necessarily finite, of atomic formulas $A_i, i \in I$. While these may represent statements that themselves are complex, such as *Triangles have three sides*, here we take them as unanalyzed at least as far as their meanings are concerned. The *propositional calculus* over the A_i is defined as the [free Boolean algebra](#) generated by the A_i .

Actually, what we have defined so far are just the well-formed expressions and formulas of the propositional calculus, a technical language to talk about the real objects of interest. More precisely, we have defined a whole *family* of languages, the specific choice of one being determined by the set of constants, variables, operations, and relations we care about. For example, if our interest is in groups, we will want, at the very least, the operations of multiplication, inverse, and unit, and the relation of equality to be part of the language. With these, we can write terms such as $M(M(a, M(b, c)), R(b))$ where M stands for multiplication and R stands for the inverse (reciprocal), and formulas like $M(a, R(a)) = E$.

Discussion Since using the general notation would be very clumsy, we are keen on introducing a more fluent notation that takes advantage of things we know about groups, such as the fact that multiplication is associative, to write $abc b'$ and $aa' = e$ in the above examples. In fact, further abbreviation such as writing a^2 for aa , a^4 for $aaaa$, or a^{-2} for $a'a'$ is often done without much thought, but here we call attention to the fact that this amounts to incorporating the language of integers \mathbb{Z} or at least the language of natural numbers \mathbb{N} into the formulas. As we shall see, \mathbb{N} is a highly nontrivial object, whose proper discussion requires quite a bit more than what is available in the propositional calculus. Therefore, we will not rely on numbers in our development of algebra and logic (but we will continue to use them as examples, of course). Since the simpler notation already presumes associativity, we will actually check our urge to simplify and use the more clumsy, but also more correct shorthand $(a(bc))b'$ to denote what we denoted



by $M(M(a, M(b, c)), R(b))$ above, and take some time to analyze seemingly innocent statements such as “parentheses will be omitted when convenient”.

Exercise[◦] 2.16 If we permit no variables, the only terms we can write are $e, (ee), e'$, and so on. What is the structure of these terms?

Exercise[◦] 2.17 Since the above terms are all equal to the multiplicative unit, the only group we could study is the trivial group with one member, which would fall rather short of our goal. It seems we need to assume an *element* other than e , say f , for which $f = e$ is false. Should such an element be considered a constant or a variable?



When we define a language of wffs capable of expressing group operations, our goal is to study actual groups such as Z_8 , the **cyclic group** of eight elements, or D_4 , the **dihedral group** of eight elements. To fix notation, we will denote elements of the former by z_0, \dots, z_7 , and elements of the latter by d_0, \dots, d_7 . As is well known, there is no mapping f between the z_i and the d_j that would respect the group operations; these two groups are *non-isomorphic*. One way to check this would be to consider all the 40,320 ways the elements of Z_8 could be mapped onto the elements of D_4 , and see where each of them will fail – this is straightforward but rather tedious. A much simpler argument runs as follows. Suppose indirectly that f is an isomorphism between the two, and take any two elements z_i and z_j . Since Z_8 is cyclic, it is Abelian, and therefore $f(z_i)f(z_j) = f(z_iz_j) = f(z_jz_i) = f(z_j)f(z_i)$, i.e. every two elements that appear in the range of f commute in D_4 . Since f is an isomorphism, every element of D_4 appears in the range and must therefore commute, which means that D_4 itself is Abelian, a fact we know to be false.

In general, proof of non-isomorphism can be effected by finding some property, such as commutativity, that separates the two objects in question. The converse is false: if we cannot find any first order property that would distinguish two objects from one another, this is insufficient to guarantee that they are isomorphic.

Exercise[→] 2.18 Find an example of two structures A and B that share every property expressible in the propositional calculus yet fail to be isomorphic. Find an example of two such structures that share every property expressible in the *predicate calculus* (see the next section).

A key point, already observable in the propositional calculus, is that one system’s arbitrary property is another’s axiom. Groups may happen to be commutative, but Abelian groups are of course commutative by definition. *Every* property can be regarded as an axiom, and every odd selection of axioms can be regarded as an axiom system. We will shortly define what it means for a structure to *satisfy* (also called to *enjoy* or simply to *have*) a property, but rather than doing this separately for the propositional calculus and the predicate calculus we start this work here and note the parts specific to the predicate calculus as we go along.

Recall that a wff of the propositional calculus is some Boolean combination of relations whose terms are constants and variables. Two things are not included: first, we cannot recursively substitute one formula in another (this is a limitation that we will address in Chapter 5), and second, there is no true quantification, a limitation that we will

transcend in the next section. For a wff to be *about* something we will need, at the very least, some objects that can correspond to the constants and other syntactic elements that make up the formula. It was realized early in the 20th century that the inventory of such objects itself needs to be regulated in some way, especially when we wish to use formulas that denote infinite concepts. Here we take the fruits of this development for granted, and identify these objects with [sets](#).

That sets are a good choice is not at all trivial, even for mathematical objects, since these come in a large sortal variety: we have numbers, triangles, derivatives, permutations, truth values, equations, topological spaces, graphs, measures, and so on and so forth, and it is not at all obvious that all these can be modeled by sets. It is evident only in retrospect, from classic summations of mathematics such as [Bourbaki](#), that sets are in fact capable of carrying all this weight. Outside of mathematics, the appropriate choice of model objects is even less evident; we will return to this issue in [Chapter 3](#).

We will define a *model structure* as a set (or, in the case of more complex signatures, several sets), equipped with the requisite operations and relations as dictated by the signature of the structure we wish to model. We will now concentrate on the one-sorted case, as it already requires the full machinery, and leave generalizing to the multi-sorted case for those occasions where this actually simplifies the presentation. We will interpret constants of the formal language as elements of this set, variables as freely ranging over this set, and operations and relations as the operations and relations of the model structure. Since the wffs themselves can be construed as elements of a set, *interpretation* is a function from this set of formulas to the model structure. This function assigns to each constant *and to each variable* of the language a particular element of the model set. In consequence, since variables get bound to elements of the model, terms will all get bound to particular elements, so the truth of formulas can be directly checked in the model structure for any given interpretation.

It is at this point that we make the cardinal distinction between *axioms* and other wffs: for axioms we demand that they hold in every model under any interpretation, while for wffs in general we make no such demand. It is worth emphasizing that it is *not* the case that axioms are true in general. Axioms acquire a distinguished status only when we discard every model structure where they fail under some interpretation. A *theory* is defined as a collection, not necessarily finite, of axioms. In the finite case, it is always sufficient to have a single axiom, because a Boolean conjunction of wffs is guaranteed to be a wff. Whether this conjunction can be satisfied is another matter entirely – it is quite possible to demand so many things at the same time that they cannot be realized in any model together. We can demand $x = 3$ and we can demand $x = 4$ but we cannot demand the two together; the wff $x = 3 \wedge x = 4$ is false, and there is no model.

For the propositional calculus, the issue of whether a set of wffs has a model is very hard to decide. In the *pure* calculus there are no operations or relations (other than equality), so each formula is simply a Boolean combination of variables (called *literals* in this context) which range over the two-element Boolean algebra \mathbb{B} , i.e. can only take the value \top or \perp .





Exercise^o 2.19 Prove that for every wff with literals x_1, \dots, x_n there exists a **conjunctive normal form** $c_1 \wedge c_2 \wedge \dots \wedge c_r$ where the conjunct clauses c_i are disjunctions of the literals and their negations. Do you need to prove the existence of an analogous disjunctive normal form?



The celebrated **Cook–Levin theorem** asserts that checking whether there is an assignment of truth values to variables that fulfills a given formula is **NP-complete**. Needless to say, when we add non-Boolean (for example group) operations, or relations other than equality, the problem of deciding whether a wff can be true does not become any easier. In fact, another celebrated theorem asserts that, with some further constraints, the general case is so hard as to be *unsolvable*.



Since the finite case is already very hard, it is particularly noteworthy that there are important positive results about the infinite case. But before turning to these in the next section, let us first see how the infinite case can even arise. Why would anybody want to have an infinite number of axioms? To see this, consider the relationship between wffs and model structures in general. We say that $M \models \phi$ (pronounced *M models ϕ*) if M is a model and ϕ is a wff of the same signature and ϕ evaluates as true under some interpretation. In this case we also say that ϕ can be *satisfied* in M . If all interpretations make ϕ come out as true, we say that ϕ is *valid* in M . Typically, a formula will have many models, not just infinitely many but in fact so many that set theory may not contain a set that collects all these models together (the collection is a proper class). Similarly, a model, even one based on a finite set, can model an infinite number of formulas. What we see here is that \models creates a **Galois connection** between sets of wffs and sets of models; obviously, if S and T are theories such that $S \subset T$ (every axiom of S is also an axiom in T), the collection $\mathcal{M}(S)$ of those models that model every formula of S is a superset of $\mathcal{M}(T)$. This connection will not preserve finiteness at all, and infinite sets of satisfiable or valid formulas will arise quite often even in the study of finite objects.



2.5 First order formulas

The way we have been able to steer clear of quantification thus far was by keeping it implicit. When we say the formula $x \geq 0$ is satisfiable over the reals \mathbb{R} , what we mean is that there exists *some* interpretation which maps x onto a nonnegative number, and when we say the the formula $x^2 \geq 0$ is valid we mean that *all* interpretations, mapping x onto any real number, will map x^2 on a nonnegative number. To make this and similar distinctions explicit we enlarge the language of propositions by two new symbols, \forall and \exists , and introduce variable scoping by means of parentheses. The wffs of the resulting system, known variously by the names (*lower*) *predicate calculus* and *first order logic* (FOL) will be our first order formulas. The full recursive definition is fairly complicated.

Atomic expressions are built in the same way as in the quantifier-free (propositional) case, but at the same time as we are building up the formulas we also build two book-keeping objects with each one: $B(\phi)$, the set of *bound variables* in ϕ , and $F(\phi)$, the set

of *free variables* in ϕ . If ϕ is an atomic term, $F(\phi)$ contains the variable x iff ϕ contained x , and $B(\phi)$ will be empty. If ϕ is a constant, both $F(\phi)$ and $B(\phi)$ will be empty – in general, if ϕ has variables x_1, \dots, x_k , each of these will appear in exactly one of $B(\phi)$ and $F(\phi)$. In atomic terms, all variables are free, and if we continue as we did in the definition of propositional formulas we will never get a bound variable. The only way to get one is to *bind* it by \forall or \exists , which is done by the following rule: if ϕ is an expression (not necessarily atomic) and $x \in F(\phi)$, the formulas ψ written as $\forall x(\phi)$ and ψ' written as $\exists x(\phi)$ are both expressions whose associated free sets are diminished by x and whose bound sets are increased by x . (Minor variants of the system can be obtained by permitting vacuous quantification such as $\forall x(3 = 3)$, or by permitting reuse of the same variable in which case we need to keep track of free and bound occurrences of the same variable, but these need not concern us here.)

Exercise^o 2.20 Write the above bookkeeping rules using set-theoretical notation.

We also need to keep track of free and bound variables whenever we perform Boolean operations on formulas or substitute terms in relations. The bookkeeping this requires is not just tedious, but also very expensive in an algorithmic sense. Historically, this has not been realized by the developers of the theory, since all the algorithms required are [computable](#), and the issue raises its head only when models of computation weaker than [Turing machines](#) are considered. We will return to this issue in Chapter 3.

As with the propositional calculus, we have defined a whole family of languages, the specific choice among which is determined by the choice of constants, variables, operations, and relations we care about, which in turn are primarily determined by the signature of the objects we want our language to be about. Note that each predicate language has a propositional subpart, and that by convention variables found in formulas of these propositional fragments are interpreted as being (implicitly) universally bound. Most importantly, the collection of model structures in which we interpret the wffs remains entirely unchanged by the extension from propositions to predicates. A model structure is defined just as above, and we define validity and satisfaction in the expected fashion. In particular, $\exists x(\phi)$ is *satisfied* in a model if there exists an interpretation that maps x onto an item s that makes the formula $\phi[s|x]$, obtained from ϕ by replacing the letter x by the letter s throughout, come out true, and it is *valid* if all interpretations can be made to exhibit such a substitution. Now we are in a position to state and prove the following lemma.

Łoś Lemma. Let S_i be structures of the same signature for $i \in I$. If S/\mathcal{U} is the ultra-product of S_i , and ϕ is any closed formula, ϕ holds in the product iff the set of indexes J where ϕ holds on the components satisfies $J \in \mathcal{U}$.

We prove this by induction on the complexity of ϕ as follows. If ϕ is atomic, the statement is true by definition. If ϕ is a Boolean conjunct, it follows from the ultrafilter properties of being closed under intersection and superset, and if $\phi = \neg\psi$ it follows from the fact that \mathcal{U} contains every set or its negation. Formulas whose main connective is a disjunction now follow from DeMorgan's laws. Existential quantification also follows trivially (given the AC), since if we find instances s_i that fulfill the existential



requirement at each component, the choice function $s_i, i \in I$ will fulfill the existential requirement for the ultraproduct.

Exercise^o 2.21 Do we need a separate proof step for formulas that have universal quantification? Why?

Recall that a *theory* T is simply a set of axioms (not necessarily finite) and a *model* M of a theory is a structure for which there is a joint interpretation of T that satisfies all formulas in T . We say that two models are *elementarily equivalent* (written with the \equiv sign) iff they satisfy exactly the same first order wffs. What the Łoś Lemma asserts is that if structures S_i are elementarily equivalent, their ultraproduct will also be so. In particular, since any structure S is elementarily equivalent to itself, for any set X of any cardinality, the ultrapower S^X/\mathcal{U} will also be equivalent to S . This will give us a powerful method for constructing models of any desired cardinality, once we have in hand the following theorem.

Compactness Theorem. Given an infinite theory T where any finite subset T_i is known to have a model M_i , the theory itself will have a model.

We begin by replacing T by a theory composed of all finite conjunctions of axioms in T – clearly, the original T will be equivalent to this new T' and, further, the condition of every finite subset of wffs having a model will hold or fail at the same time in T and T' , meaning we can assume T to be closed under finite conjunction of axioms without loss of generality. Now let F be the collection of finite subsets of wffs in T , and for any formula $\phi \in T$ take $F(\phi)$ to be the collection of those finite subsets of T whose models model ϕ . The set of such models is nonempty (since every finite set of formulas in T , including the singleton set $\{\phi\}$, has a model), and is closed under finite intersection, because we have added all finite conjunctions of formulas. Therefore, F can be extended to an ultrafilter \mathcal{F} , and the ultraproduct $\prod_{S \in T} M_S/\mathcal{F}$ will, by the Łoś Lemma, model all wffs in T .

Discussion A clever choice of terminology can make this theorem look bigger than it is. In everyday usage, a set of statements is called *consistent* if no contradiction can be derived from it. As we will shortly see, under certain assumptions ‘having a model’ and ‘containing no contradiction’ are closely related notions, so the former is called *semantic consistency* (also called *satisfiability* or just *consistency*), while the latter is called *syntactic consistency*. Using (semantic) consistency, the theorem says that if an infinite set of formulas T is such that each finite subset is consistent then the set, as a whole, is consistent.

We call an axiom system *categorical* if it succeeds in defining its subject matter up to isomorphism. This would mean that for a model M we could find a theory T such that $M \models T$ and $M' \models T$ implies $M = M'$. Those who took the time to solve Exercise 2.18 will know that this is possible only for finite structures, where every part can be individually tied down. In the infinite case, even for those familiar and well-understood structures like the points, lines, angles, and vertices of geometry, the natural numbers \mathbb{N} , and the real numbers \mathbb{R} , elementary equivalence is less than isomorphism, since bi-

jectively mapping two models of different cardinality on one another is impossible. We have the following theorems.

Löwenheim–Skolem Theorem (downward) If T has an infinite model, it has a denumerable model.

Löwenheim–Skolem Theorem (upward) If T has a denumerable model, it has a model at every infinite cardinality.

The two theorems, taken together, imply that a semantically consistent first order theory with any infinite model will have a model at every infinite cardinality, so the model structures are too numerous to fit into a single set. Among finite models, it is always possible to restrict attention to models with at least n elements by adjoining variables x_1, \dots, x_n and axioms $x_i \neq x_j$.

Exercise^o 2.22 Is it possible to restrict attention to models with at most n elements?

2.6 Proof theory

Besides a formal language with a well-defined syntax that lets one express many (but not all) logical expressions that contain variables, and a means of interpreting the language in models, first order logic also contains the means for *proving* statements from axioms. By a proof we mean a finite sequence of formulas, structured in *lines*, each line being a finite list of some formulas. The first line can only contain axioms, the last line must contain the wff to be proven, and all lines except the first must be *deducible* from the previous line in the following sense: either a conjunct already appears in the previous line or there is a specific *rule of deduction* that sanctions introducing it there.

There are several rules of deduction in common use, and the theory of logic is greatly concerned with establishing their range of applicability. Any such rule will have one or more inputs, called its *premisses*, and one output, called its *conclusion*. We say that a rule is *deductively valid* or just *valid* (not to be confused with the validity of a formula) iff for every model where the premisses hold the conclusion also holds. Weaker forms of validity (sometimes collected together under the name *inductive validity*) include rules of deduction based on probabilities, degrees of belief, etc. We will discuss some examples in Section 7.3. It should be kept in mind that deductive validity is not an absolute property of rules: some rules are deductively valid in some systems of logic but not in others. A typical case is the deductive rule of **double negation**, which leads to the conclusion ϕ from the premiss $\neg\neg\phi$. This rule is deductively valid in the classic two-valued system discussed in this chapter, but it fails for example in many-valued systems of logic. The most important of these is **intuitionistic logic**, which disallows both double negation and the closely related **tertium non datur** (law of the excluded middle).

Exercise^o 2.23 Take any classic theorem that is routinely proven by contradiction, such as $|\mathbb{R}| > \aleph_0$, and show how the proof fits into the scheme described above.

We will use the \vdash (read ‘proves’) symbol and write $\phi \vdash \psi$ to denote the situation when the list of formulas ϕ (traditionally separated by commas rather than \wedge but



interpreted conjunctively) is used as the first line and some specific *finite* sequence of deductive rules permits the formation of a line containing ψ . This is not to be confused with the extended Boolean connective $\phi \rightarrow \psi$, used simply as an abbreviation of $\neg\phi \vee \psi$, nor the semantic notion of implication \Rightarrow , which connects a theory (known in this context as the *hypothesis*) Φ and a formula ψ by demanding, for every model $M \models \Phi$, that $M \models \psi$. (Sometimes the word ‘theory’ is reserved for deductively closed sets of statements, but we will not follow this usage here.)

This latter situation, with a slight abuse of notation, is often denoted $\Phi \models \psi$, but we will refrain from doing so, restricting \models to the sortally correct use where the object to the left of \models is a model structure and the object to its right is a formula. We do make one exception for *tautologies*, which are formulas that are true in every model: if ϕ is a tautology such as $x \vee \neg x$, we will sometimes write $\models \phi$ to denote the fact that ϕ obtains in every model (rather than the empty model, as the notation would suggest). Tautologies, these seemingly trivial statements, are of great importance in the study of logic, since proofs often proceed by moving from some statement ϕ to a tautologically equivalent statement ψ . A typical case would be proving $\exists x f(x)$ from $\exists y f(y)$.

In FOL, once we have a method for dealing with the tautologies that arise in connection with renaming variables, we need only one rule of deduction to get good results, **modus ponens**, already known to Aristotle: $P \rightarrow Q, P \vdash Q$. What have at this point are two seemingly different notions, \vdash and \Rightarrow , that can connect a theory (collection of axioms) T and a formula ϕ : with $T \vdash \phi$ we say that we can prove ϕ by a finite series of mechanistic formula-manipulation steps, and with $T \Rightarrow \phi$ we say that there is no model that satisfies T but not ϕ . What connects the two is the following theorem.

Completeness Theorem Given a first order theory T and a wff ϕ , $T \vdash \phi$ iff $T \Rightarrow \phi$.

To prove the ‘if’ part of the theorem requires only some demonstration that our proof theory is *sound* in the technical sense that it contains no rule of inference that leads to a false conclusion from true hypotheses. Clearly, tautological steps like the renaming of bound variables will be sound.

Exercise^o 2.24 From the formula $\forall x f(x) \vee y$ can we obtain $\forall y f(y) \vee y$? Why or why not?

To prove the other direction assume $T \Rightarrow \phi$. There are two important extreme cases: when T is empty (so that ϕ is a **tautology**, true in every model), and when the implication is true vacuously (because T has no model at all). For the first case, we will state without proof the following lemma.

Validity Lemma If $\models \tau, \vdash \tau$.

The lemma says that if τ is a semantic tautology, it has a proof, i.e. our proof theory is strong enough to reach every unconditionally true statement. Needless to say, our primary interest is not with this special case (tautologies are boring), but rather with $T \Rightarrow \phi$ for some substantial, possibly infinite, T . If T has no model at all, we can rely on the following lemma.



Inconsistency Lemma Given a finite set of jointly unsatisfiable hypotheses F , and any desired conclusion ϕ , both $F \vdash \phi$ and $F \vdash \neg\phi$.

This is easy if F already contains two syntactically opposed formulas ψ and $\neg\psi$, because by monotonicity we conclude from ψ that $\psi \vee \phi$ and from this, now relying on $\neg\psi$ we can conclude ϕ by modus ponens. The same proof steps of first introducing $\neg\phi$ and then eliminating ψ will also prove $\neg\phi$.

The difficulty is in constructing, from two or more (finitely many) axioms that are not syntactically opposed, but nevertheless jointly unsatisfiable (take, for example, $x = 2$ and $x = 3$) a proof of a syntactically opposed pair, i.e. a direct contradiction. Since we have finitely many axioms, we can take their conjunction Φ , which is trivially provable from F , but has no model. Therefore $\neg\Phi$ is true in every model and, being a tautology, has a proof by the Validity Lemma. We have thus constructed two statements, Φ and $\neg\Phi$, which are now syntactically opposed.

Finally, there remains the central case of the completeness theorem, where $T \Rightarrow \phi$ holds neither by virtue of the simultaneous unsatisfiability of the premisses in T nor because T is empty. Here we form the theory T' from T by adjoining $\neg\phi$. Since every model of T is a model of ϕ , the theory T' has no model. But if it has no model, there must exist, by the Compactness Theorem, some finite subset F of T' that already has no model. Now it is either the case that $\neg\phi \notin F$, meaning that our hypothesis T was already inconsistent, and the Inconsistency Lemma is applicable, or it is the case that every finite $F \subset T'$ that lacks a model will have $\neg\phi \in F$. Denoting by ψ the conjunction of all members of F other than $\neg\phi$, we know that $\psi \wedge \neg\phi$ has no model, so its negation, being true in every model, has a proof by the Validity Lemma. This negation, $\neg(\psi \wedge \neg\phi)$, combined with the premiss ψ , is the last step in the sought after proof of ϕ by modus ponens.

As the preceding makes clear, the only hard parts of the Completeness Theorem are to show that our proof system is strong enough to prove everything from a contradiction (the Inconsistency Lemma, which we have already proved above), and to show that it is strong enough to prove all tautologies (the Validity Lemma, which we do not prove here).

Altogether, Boolean algebra (the propositional calculus) and FOL (the predicate calculus) represent the culmination of centuries of work in logic. There is a lot more to be said, especially in the realm of higher order and modal logics, but none of these more powerful systems will ever be as good as FOL. This is stated in the following theorem.

Lindström's Theorem FOL is the *strongest* logic satisfying the downward Löwenheim–Skolem theorem and compactness that is closed under conjunction, isomorphism, negation, and type-abstraction.

(Here [strength](#) is a technical term referring to expressive strength, the ability to capture [elementary classes](#).)



2.7 Multivariate statistics

Compositional semantics puts the emphasis on deduction in the sense of the previous section, establishing the truth of some statement based on our knowledge that certain premisses are true. For lexical semantics, the key step appears to be *concept formation*, the ability to acquire words based only on a few examples. We begin with some terminology common to linguistics, cognitive science, and philosophy, where the issue of concepts and concept formation is generally approached through some theory of [natural kinds](#), and gradually recast the main observations in the language of machine learning.



Phil

First, it is evident that children are quite capable of ‘learning’ natural kinds based on very few exposures, and that much of that capability remains with them as adults. It is enough to take a guided tour in a forest with the guide pointing out kinds of trees, bushes, or animals you have never seen before, and pretty soon you are able to identify these for yourself. What you learn is of course not the species, but a *name* for it, and ideally some salient properties, whether the fruit of the tree is edible, whether the animal will attack humans, etc. – we return to the matter of capturing such regularities in Section 3.6. The selectional advantage this knowledge confers is evident both at the individual and at the group level.

This is one of the areas where the old AI adage ‘if a human can do it, a computer can do it’ is truly tested. Not only can humans do it, they do it really well, and whatever enables them to do it extends smoothly from natural kinds to cultural kinds such as distinguishing the letters α and β . (Here we consider a species, such as α , to be a natural kind as it stands, but it is possible to construe an entire collection of species, such as ‘the Greek alphabet’, as a higher-order natural kind.) Even with the latest crop of deep learning algorithms, computers still do this rather badly, and require hundreds or even thousands of learning examples. An important case in point is optical character recognition (OCR), where the standard [MNIST](#) dataset contains about 6k training images per character (handwritten digit), well over two orders of magnitude more than what a human requires to learn the classes.



Exercise \rightarrow **2.25.** Train yourself on an alphabet you haven’t learned at school, recognizing for example handwritten Arabic or Hindi digits. How many examples did you need in order to learn how to read these? Does this number somehow depend on the prolepsis (‘foreshadowing’, advance knowledge, see Chapter 3), for example on the fact that you have already learned, presumably with a great deal more effort, some other alphabet first?

Second, it is equally evident that the pattern matching skill deployed during the acquisition of those words denoting natural kinds cannot account for the entirety of concept formation. People know exactly what it means to betray someone or something, yet it is unlikely in the extreme that parents tell their children “here is an excellent case of betrayal, here is another one”. Studies of children’s acquisition of lexical entries such as McKeown and Curtis (1987) have made it clear that natural kinds, however generously defined so as to include cultural kinds and artifacts, make up only a small fraction of the

vocabulary learned, even at an early age, and that children’s acquisition of abstract items “but not concrete word learning, appears to occur in parallel with the major advances in social cognition” (Bergelson and Swingley, 2013).

On occasion, we see group names such as ‘mammal’ that are definable by means of relatively short lists or visible external criteria. For these, a simple disjunctive theory of word meaning could be grounded in concrete ‘natural’ terms, but some words of caution are in order. A theory so constructed will not coincide with the biological taxonomy we learn in school – we need to learn that dolphins and whales are not fish, but rather air-breathing mammals. Similarly, our naive definition of rats or mice is independent of their being mammals, and the mammaries of their females can hardly serve as a salient distinguishing characteristic akin to wings on a bird. And for the rest (actually, the vast majority) of words, neither salient nor obscure distinguishing characteristics are readily available: it is a duck because it looks like a duck and quacks like a duck.

To formalize this, we replace objects with features computed from them. These features are typically binary if the input is discrete, and typically real-valued if the input is continuous. Either way, they are computed from the input by shallow, mechanistic methods, which makes them very different from features like *has-webbed-feet* that a biologist might consider helpful for distinguishing duck from non-duck. Most images of duck-like objects will have the feet obscured by water or grass, and the machine learner is in no position to chase after the bird and inspect its feet. Since the feature is applicable only in a few corner cases, machine learning algorithms may not even realize its value.

In what follows, we assume binary features are mapped onto the values -1 and $+1$, and k -valued features are mapped equidistantly between these two extreme values. For the sake of simplicity we also assume that continuous-valued features are *squished* into this interval by the application of some [sigmoid function](#) such as the hyperbolic tangent $\tanh()$. With these assumptions, the entire input range is mapped onto a hypercube $[-1, 1]^n$ of the n -dimensional Euclidean space \mathbb{R}^n , making every object a point of this space called the *feature space* or, when a probabilistic interpretation is desired, the *sample space*. This makes eminent sense for static tasks such as character or object recognition. For dynamic tasks, we consider the input as a sequence of such points or, if continuous time is essential, a trajectory in the feature space, but we will mostly discuss the static case here, following ideas first put forth in Valiant (1984).

We define a *concept* by its characteristic function in the feature space or by its density function in the sample space. To speak of *learning* or *forming* a concept we also need to specify the *hypothesis space*, generally as a family H of concepts (sets) that are all subsets of the same universe, the *sample space* S , endowed with a fixed (but not necessarily known) probability measure P that dictates both how data will be presented to the learning algorithm and how the goodness of fit is to be measured between the target concept C and a hypothesis C' proposed by the algorithm. We say that C' *approximates* C within ε if $P(C \Delta C') < \varepsilon$ (here Δ is used to denote the symmetric set difference). As for our criterion of success, we say that an algorithm δ, ε *probably approximately correctly learns* or *PAC-learns* C if, after being presented with a sufficient number n

Comp



of randomly (according to P) chosen labeled examples, it produces, with probability $> 1 - \delta$, a concept C' that approximates C within ε . Our chief interest is in algorithms that are polynomial in $n, 1/\delta$ and $1/\varepsilon$, and ideally we would want algorithms that are robust under a change of the distribution P or even *distribution-free* (i.e. independent of P). In Chapter 4 we will introduce a conjunctive theory of concepts, and will rely heavily on Valiant's classic result that such concepts are PAC-learnable as long as the number of disjuncts in a conjunct is limited to some constant k .

Valiant defines concepts as *subsets* in the sample space, but this formulation of the problem is in many ways interchangeable with a simpler one, where we assign only a single *point* (feature vector) to a concept. First, if the set is very narrowly concentrated on its mean value, using the mean as the single statistic to characterize it can be very reasonable: in everyday life, this is how we identify mountains with their peaks for most practical purposes. Second, even if the set is less narrow, a well-placed n -dimensional Gaussian, given by the n means and the $n(n - 1)/2$ covariances, may still give a very reasonable picture of it, at the cost of moving from n -dimensional to $n(n + 1)/2$ -dimensional space. Even if the data is more clustered, a weighted sum of k such Gaussians can still give a very good description of the original set by a single vector, now having $kn(n + 1)/2 + k - 1$ dimensions. In fact, we can get arbitrarily close (within any prescribed ε) to any distribution at the cost of increasing the number of components k in such a Gaussian mixture model (GMM). Third, in situations where the probabilities are very low and hard to estimate, we may be satisfied with a simple polyhedron surrounding the concept set, now given in terms of linear inequalities defining its faces, again at the cost of Kn parameters for some fixed K . A particular case, when we define the regions of interest by simple affine cones (half-spaces), will be discussed in Chapter 9.

For the remainder of this chapter we switch from the set-based viewpoint to the vector-based viewpoint, without particularly committing ourselves to any of the three view-switching techniques described so far. If concepts (possibly associated to single words, but quite possibly to more complex linguistic expressions) are to be modeled as vectors, the key question is how to compute the features that make up the coordinates of these vectors. The direct approach, pioneered by Osgood, is simply to ask the people, and the method he used for making sense of the answers is central not only to his theory of [semantic differential](#) (SD) but also to many of the indirect methods that we will discuss later.



Example 2.1 *Ranking on a scale.* In surveys we often find questions asking us to rank actions, objects, or statements on a polar scale: snails au gratin are very appetizing (+2); somewhat appetizing (+1); neither appetizing nor disgusting (0); somewhat disgusting (-1); or very disgusting (-2). Sometimes the scale runs from -3 to +3, sometimes from 1 to 5 – in all cases we can begin by normalizing to $[-1, 1]$. Assume you have obtained a total of N responses from r respondents to n questions about m items: this can be summarized in a 3D array S whose (i, j, k) element is the response of respondent

i to question j about object k . (Somewhat confusingly, such questions are often called *measurements* in statistics and *features* in machine learning.)

There are several strategies to make sense of such data. First, we may want to know if the answers are consonant: are the slices $S_{i,,}$ of the array similar? If not, are there clusters among the respondents? Were the questions probing the same thing, are the slices $S_{.,j}$ similar? (In survey research, this is a standard technique, asking the same question in different ways, so as to obtain some estimate of the consistency/reliability of the respondents.) Are the objects drawing the same response patterns, i.e. are the slices $S_{.,k}$ similar across different values of k ? To some extent, the responses to these three kinds of inquiries are mutually reinforcing: if we already have a reliable classification of objects and questions, we can use these to pool the data and obtain, perhaps, a much more reliable classification of respondents.

Osgood and his coworkers displayed great methodological care in establishing the geometry of the semantic space, including specific tests to make sure that the adjectives at the two ends of the scale are indeed polar opposites, that zero is indeed at the middle of the scale, etc.

Exercise[†] 2.26 Formulate a test to decide whether *unappetizing* or *disgusting* is a better opposite of *appetizing*. Devise a method for finding all, or at least a large number of, polarly opposed adjectives. How would you test whether neutral judgments on a particular scale are indeed halfway between the opposites? Can you provide some cases where you suspect that the null hypothesis is actually wrong?

We now describe a classical method of multivariate statistics aimed at answering some of these questions. The method, originally due to Pearson (1901), is called **principal component analysis**, or PCA for short. We begin by normalizing the 3D array S used so far in two ways: first, we ignore the j, k structure and assume all responses by a given subject i are collected in a row vector that now has nm coordinates, and collect these vectors in a data matrix D with r rows and $c = nm$ columns. Second, we normalize the data by subtracting the mean of each column from each entry in that column. An optional third step is to normalize the variables for variance as well, by dividing each column by its variance. The procedure we describe here is meaningful both with and without such normalization for variance, but normalization for the means (also called ‘means centering’) must always be performed.

In statistics, we call the columns *variables* and think of the rows as (independent) observations of the variables. The i, j element of the **covariance matrix** C is given by the scalar product of columns i and j of D . $C = D^T D$ is symmetrical and positive semidefinite. C is positive definite as long as the columns of D are linearly independent, a condition not always satisfied as we tend to have more columns (in current applications, 10^7 or more) than rows (in current applications, about 10^5). Be that as it may, the variance in an arbitrary direction \mathbf{x} is given by $\mathbf{x}^T C \mathbf{x}$, and the first **principal component** of the data is defined as the direction that maximizes the variance. To find it, we need to solve



$$\frac{d}{dx} \mathbf{x}^T C \mathbf{x} - \lambda \mathbf{x}^T \mathbf{x}, \quad (2.1)$$

where the second term is a [Lagrange multiplier](#) that comes from the constraint of keeping the length of \mathbf{x} fixed. Thus the critical points are obtained from solving $C\mathbf{x} = \lambda\mathbf{x}$, so the solutions λ_i are, by definition, the eigenvalues and the x_i are the corresponding eigenvectors.

Exercise \rightarrow **2.27** Prove, without reference to singular value decomposition or the spectral theorem, that eigenvectors belonging to different eigenvalues of a real symmetric matrix C are orthogonal. Do you have to rely on C having the special form $D^T D$?

Arranging the eigenvalues in decreasing order (all are real, since C is symmetric), we call the first eigenvector \mathbf{x}_1 the *first principal component*, \mathbf{x}_2 the second, and so on. In practice, C is large, and we are rarely interested in more than the first few (or few hundred) principal components. If the [singular value decomposition](#) of D is UGV^T , the columns of V are exactly the eigenvectors of C , and the positive singular values found in the diagonal matrix G (conventionally arranged to run from larger to smaller) are the square roots of the eigenvalues λ_i of C , which we use to measure the “goodness” of principal components. Writing $\Lambda = \sum_{i=1}^c \lambda_i$, we say, slightly misleadingly, that each λ_i *accounts for* a fraction λ_i/Λ of the total variance.

By the Eckart–Young theorem, if the first a columns of U are collected together in U_a , the first a columns of V in V_a , and the first a singular values (by decreasing size) in G_a , the matrix $C_a = U_a G_a V_a^T$ is the best rank- a approximation (in [Frobenius norm](#)) of C . This approximation is unique as long as the first a eigenvalues are distinct, a condition generally met in the cases of interest. Thus we can think of PCA as a data compression technique, replacing the original data matrix by a much simpler one. Until the advent of computers, engineering practice was largely restricted to vectors of rather small dimension, and even in the 1970s, the basic study of semantic space (Osgood, May, and Miron, 1975) was restricted by the inability of computing machinery to invert matrices larger than 100 by 100. Today, the use of vectors of length 10^5 – 10^6 is commonplace, thanks in great part to well-defined, highly optimized, and historically well-debugged foundational libraries, [OpenBLAS](#) in particular.

Data compression is a worthy goal in and of itself, but in terms of interpreting the results we may want to go further. We have already noted that part of our goal may be to *cluster* the data: for example, we may find out that people who find snails au gratin appetizing also find frog legs appetizing, and we may be able to correlate this with them having had significant contact with French cuisine and French culture in general. This leads to a characterization of the data in an essentially non-scalar, discrete manner, using some number h of clusters C_i such that points within a cluster are relatively close to one another, making the within-cluster variance small. PCA is at the opposite, continuous end of the scale, in that the similarity $s(x, y)$ among observations x and y is expected to be accounted for in terms of a (Euclidean) distance $d(x, y)$, using some monotonically decreasing function of a single variable f to express $s(x, y)$ as $f(d(x, y))$ (the lower the

distance, the higher the similarity). In between these two extremes we find hierarchical clustering, which assigns a tree structure (dendrogram) to the clusters.

Since PCA provides the best rank- a approximation to the correlations C in the original data, the resulting matrix C_a is typically dense even if the original data matrix D is sparse. One way to improve the perspicuity of the result is by change of basis: we keep the transformation that C_a expresses between the eigenvectors and the variances, but instead of describing them in the natural basis, we select a new *varimax* orthonormal basis that *maximizes* the *variation* of the squared components of the new basis vectors. The components will vary maximally if they are as far from one another as they can be, so a few ones and many zeros are preferred by the varimax criterion. Since the change of basis is from one orthogonal system to another, it is spoken of as a *rotation*. The table presented in 2.1, reproduced from Table 3.3 of Osgood, May, and Miron (1975), illustrates the results of the procedure for three principal components.

The original data was about how people rate one adjective like *dark* on a seven-point scale of some other polar adjective pair such as *full–empty*. After PCA, three main components tend to emerge, dubbed EVALUATION, POTENCY, and ACTIVITY, so that for example the Finnish equivalent of *honorable–despicable* has 0.94 of the variance associated with it explained by EVALUATION, while in English the same pair (great care was taken to select translation equivalents) did not make it to the list of the six most highly correlated adjective pairs.

No matter what the data says, it is always possible to take the three main principal components. The justification for these particular three components E–P–A, which are the central finding of the SD theory, comes from three sources. First, that three components account for quite a large chunk, about two thirds, of the total variance. Second, that the loadings are remarkably high, on what appear to be rather clear oppositions. Finally, that the components so selected are reasonably (though of course not entirely) stable across languages/cultures.

That said, a more critical view of SD will discover some problems. First, to get the data aligned across languages one has to make a somewhat arbitrary decision about whether to rotate or not. Second, maintaining the cross-linguistic identity of the factors can get very tricky: for example, the ACTIVITY factor has no common elements in the top six across Dutch and Japanese. Finally, the naming of these factors leaves something to be desired: at the very least we expect the three principal components to be sufficiently different to be able to explain how *sturdy–fragile* has to do with potency and *soft–rough* with evaluation and not the other way round. Consider the factor loading data shown in Table 2.2, collected from Landis and Saral (1978) in Sewell and Heise (2010).

As is evident from the different loadings computed under different assumptions, very little of the analysis stays entirely stable. The relative order of the best-fitting polar opposite scales is strongly perturbed, again calling the reality of the principal components into question. The issue of naming is much bigger than it appears at first blush, as it leads to the issue of *reification*: just because we can compute something, it does not necessarily correspond to something in reality.

*Scale-on-Scale Analysis: Salient Scales after Orthogonal Rotation
of Principal-Component Factors**

	EVALUATION	POTENCY	ACTIVITY	
ENGLISH	FACTOR I (44%)	FACTOR II (15%)	FACTOR III (9%)	
	nice-awful	.92 big-little	.86 burning-freezing	.81
	fine-coarse	.92 powerful-powerless	.81 hot-cold	.76
	heavenly-hellish	.91 strong-weak	.77 fast-slow	.65
	smooth-rough	.91 long-short	.75 sharp-dull	.53
	mild-harsh	.88 full-empty	.67 light-dark	.50
	clean-dirty	.87 many-few	.65 young-old	.49
DUTCH	FACTOR I (42%)	FACTOR II (15%)	FACTOR III (10%)	
	beautiful-ugly	.93 impressive-insignificant	.84 thin-thick	.73
	pleasant-unpleasant	.93 loud-soft	.75 yellow-blue	.70
	good-bad	.92 big-little	.73 loose-firm	.61
	pretty-not pretty	.92 strong-weak	.72 fast-slow	.55
	happy-unhappy	.91 wild-tame	.67 unexpected-expected	.49
	tasty-dirty	.91 much-few	.67 new-old	.49
FINNISH	FACTOR I (47%)	FACTOR II (11%)	FACTOR III (7%)	
	right-wrong	.95 large-small	.77 young-old	.74
	honorable-despicable	.94 deep-shallow	.76 growing-diminishing	.69
	good-bad	.94 heavy-light	.73 strong-weak	.53
	valuable-worthless	.93 difficult-easy	.64 courageous-timid	.50
	useful-useless	.93 black-white	.63 fast-slow	.45
	clever-stupid	.92 dark-light	.63 glad-sad	.44
FLEMISH	FACTOR I (42%)	FACTOR II (11%)	FACTOR III (10%)	
	agreeable-disagreeable	.94 deep-shallow	.78 violent-calm	.81
	good-bad	.94 serious-frivolous	.73 impetuous-quiet	.77
	magnificent-horrible	.91 big-small	.71 quick-slow	.57
	beautiful-ugly	.91 difficult-easy	.66 strong-weak	.57
	pleasant-boring	.90 long-short	.63 young-old	.57
	clean-dirty	.90 heavy-light	.62 frequent-seldom	.57
JAPANESE	FACTOR I (45%)	FACTOR II (16%)	FACTOR III (7%)	
	pleasant-unpleasant	.94 deep-shallow	.86 cheerful-lonely	.81
	good-bad	.93 thick-thin	.81 noisy-quiet	.76
	happy-sad	.93 complex-simple	.68 near-far	.65
	skillful-unskillful	.90 strong-weak	.68 hot-cold	.53
	thankful-troublesome	.90 sturdy-fragile	.67 intense-calm	.50
	agreeable-unagreeable	.90 heavy-light	.67 early-late	.49
KANNADA	FACTOR I (49%)	FACTOR II (7%)	FACTOR III (8%)	
	best-mean	.94 big-small	.73 fast-slow	.83
	clear-unclear	.92 wide-narrow	.65 wonderful-ordinary	.66
	soft-rough	.91 huge-small	.58 many-few	.57
	pure-impure	.90 great-little	.55 red-black	.54
	beautiful-ugly	.89 plenty-few	.54 public-secret	.49
	delicate-rough	.88 many-few	.53 fatty-slim	.45

*Except for Dutch and Japanese, where loadings are for unrotated principal-axes solutions

Table 2.1. Table of results from Osgood, May, and Miron (1975)

Evaluation	Potency	Activity
Adjectives for Black English		
Good–foul (0.88 0.73 0.77)	Large–small (0.84 0.68 0.63)	Fast–slow (0.51 0.56 0.36)
All right–mad (0.86 0.70 0.80)	Big–small (0.83 0.66 0.58)	Jive–straight (0.12 0.37 0.19)
Clean–nasty (0.83 0.66 0.76)	Big–little (0.81 0.57 0.49)	Frail–wide (0.09 0.26 0.20)
Together–wrong (0.77 0.73 0.75)	Get–lay (0.20 0.14 0.16)	Beat–straight (0.08 0.34 0.21)
Adjectives for White English		
Nice–awful (0.96 0.89 0.94)	Big–little (0.81 0.84 0.79)	Noisy–quiet (0.56 0.53 0.54)
Sweet–sour (0.94 0.87 0.89)	Powerful–powerless (0.75 0.61 0.47)	Young–old (0.56 0.30 0.18)
Good–bad (0.93 0.87 0.94)	Deep–shallow (0.69 0.65 0.61)	Fast–slow (0.64 0.65 0.73)
Helpful–unhelpful (0.90 0.84 0.91)	Strong–weak (0.67 0.63 0.47)	Alive–dead (0.55 0.46 0.55)

Table 2.2. Adjectives for semantic differential scales in American Black English and American White English (extracted from Landis and Saral (1978)) *Note:* The first number in parentheses is the factor loading of the scale in an indigenous principal components analysis without rotation. The second number is the factor loading in a bi-cultural principal components analysis, with varimax rotation. The third number is the factor loading of the scale in a pan-cultural factor analysis.

Example 2.2 *Latent semantic analysis (LSA).* Surveying people’s opinions is a slow, error-prone, and expensive process: to guarantee reasonably replicable results, great care needs to be taken in selecting the subjects, the stimuli, the experimental protocol, and so forth. Many, if not all, of these problems can be avoided by looking at the spontaneous products of linguistic behavior. Textual documents are produced in abundance, and instead of investigating how well *nice* and *large* correlate in people’s opinion, we may simply consider how often they cooccur in texts. This time we build a *term–document matrix*, where rows correspond to words (or word stems, to reduce the number of rows) and columns to documents, with the i, j -th entry counting the number of times word i occurs in document j . By the time this method made its appearance (Deerwester, Dumais, and Harshman, 1990), PCA of a matrix with several thousand rows and columns was feasible, and by keeping the first hundred eigenvectors LSA provided a measurable improvement over contemporary methods of document retrieval.

Example 2.3 *Embeddings.* Instead of investigating the cooccurrence of words within the same document, we may be interested in a finer classification based on much smaller contexts, roughly the size of a single sentence. Let us define a *context window* around a word w_i in some text by the words $w_{i-l}, w_{i-l+1}, \dots, w_{i-1}, w_{i+1}, w_{i+2}, \dots, w_{i+r}$ (l words to the left and r words to the right), together with some weights $\alpha_{i-l}, \alpha_{i-l+1}, \dots, \alpha_{i-1}, \alpha_{i+1}, \alpha_{i+2}, \dots, \alpha_{i+r}$ that may be used to shape the context window. The *term–context* matrix has as many rows as there are distinct words in some large corpus, and as many columns as there are contexts. Computational limitations still require pruning of words (those that occur fewer times than some threshold T are omitted) but the number of contexts (columns) can run into the billions. The measure of association between a word and a context is not the raw count (how often w occurs in c) but, rather, *pointwise mutual information* (PMI) (Church and Hanks, 1990) or the maximum of the PMI and 0 (positive PMI, PPMI). Keeping the first d (typically a few hundred) eigenvectors after





PCA of this term–context association matrix is a good method for obtaining an [embedding](#) of words into d -dimensional space. There are many other ways of generating such embeddings, often strongly connected to ideas about [deep learning](#).

For now, all we need to keep in mind is that vectors offer a significant method for representing meaning, one that has characteristics and failure modes entirely different from the use of logic formulas that we described in Section 2.5 above. Yet another method, based on (hyper)graphs, will be introduced in Chapter 4.

How many instances of a word (in context) do we need to train such vectors? The question is particularly acute in light of the fact that most words are rare. Valiant’s results can be used to put upper bounds on the number of training instances (and, in some cases, oracle questions) that an algorithm may need, but a lot of assumptions are required to make this work and, even so, the results are not very promising, in that in practice we may be satisfied with considerably fewer examples than the theory suggests. Let us now turn to the issue of learning based on few examples, often just one or two – we leave the matter of “zero-shot learning” (Socher et al., 2013) for later. Assume there are some distinguished points t_1, t_2, \dots in n -dimensional Euclidean space, or distributions strongly centered on small neighborhoods of these points, that we wish to learn. What we have as input data are sets of samples (instances) $a_{1,1}, \dots, a_{1,k_1}$ for t_1 ; $a_{2,1}, \dots, a_{2,k_2}$ for t_2 ; and so on. We will write A_i for the instance sets and $t_i = \lim A_i$ even though there is no ordering among the instances. This is not like ordinary limits in \mathbb{R}^n : the k_i sample set sizes are small. Often, in cases of “one-shot learning”, we have only one instance, and there is no assumption that the limit must equal the instance.

Example 2.4 Let the space be one-dimensional and the t_i be the integers. For an instance a , which can be any real number, we define $\lim a$ as the integer nearest to it. There are some ambiguous training samples sitting exactly halfway between two adjacent integers, but these have measure zero and can be ignored.



The trivial generalization of Example 2.4 is a set of points t_1, t_2, \dots and the [Voronoi tessellation](#) they give rise to. The problem is that learning some t_i in such a setup would require knowledge of the t_j near it. This is what linguists generally refer to as knowing something ‘in opposition’ to the other elements of the system. You learn that houses have fixed walls, as opposed to tents; are occupied by single families, as opposed to condos; have only a few stories, as opposed to skyscrapers; and so on. Trivial, but seldom noted, is the concomitant fact that we do *not* learn that houses don’t have two legs and two arms, as opposed to persons; or that they are not used to preserve raw meat, as opposed to salt. Things close by in semantic space are relevant; those farther away are not. Also, negative facts (treated on a par with positive facts in foundational studies ever since Russell) are seldom relevant: *blind* means without sight, but a piece of rock, while clearly lacking vision, is not considered blind. Negative facts seem only to enter the picture through explicit denial of positive defaults, a matter we shall return to in Section 7.3.

If real-life examples were like Example 2.4, building learning algorithms would be a trivial matter. One problem is that we don’t have the t_i at hand; we only have the A_i . Another problem we must address is that we can already learn *house*, or a good

approximation to it, without having learned *tent*, *condo*, *skyscraper*, and related terms. If anything, the process is the opposite – we learn about *house* first, and consider tents, industrial buildings, etc. to be vaguely house-like objects. This is evident for words like *mirror* or *wiggle* that are *sui generis*, since these offer no significant contrasts.

To make the problem more concrete, in Section 3.9 we will discuss the case where the objects to be learned are words, and their feature vectors are considered to be a model of their meaning. Suppose that the t_i are semantic vectors corresponding to English words w_i , and that the A_i are our training examples, typically objects and actions observed in the world, such as a mirror or someone wiggling his toes. Clearly, there are many similar objects and actions – the central fact to be explained is that I have learned *mirror* from exposure to different samples than you, yet our notions of mirrorhood coincide to a remarkable degree. An experimenter can place two people in a huge warehouse that contains, among many other objects, a few mirrors, and ask them to annotate all these objects for mirrorhood. The inter-annotator agreement will be near perfect, but why?

The standard answer, going back to Aristotle, is that mirrors have a *genus*, ‘smooth, near-flat surface’, and a *differentia specifica* ‘reflective’, and what we learn are these two. But this just seems to be pushing the problem back, since now we may ask how we learn ‘smooth’, ‘flat’, and ‘surface’. Also, the genus seems redundant, since only smooth surfaces reflect, and only near-flat ones reflect without distortion. And what do we do with *the structure of this book mirrors the historical development of the subject*? To claim that such usage is metaphorical contributes nothing to the knowledge acquisition puzzle, since people, upon encountering such situations, will again show high inter-annotator agreement.

The goal, then, is to discover some function P over \mathbb{R}^l that makes the t_i its local minima near the points in A_i . We say R^l , since we must assume some embedding not only of words into n -dimensional Euclidean space but also of the stimuli, be they actions such as wiggling one’s fingers; objects, such as houses; or sounds, colors, etc. We may begin by assuming some projection $T : \mathbb{R}^l \rightarrow \mathbb{R}^n$ so that we compare $T(a_{i,j})$, rather than the $a_{i,j}$ directly, with the t_i to be learned. This is slightly more complex than the setup used by Baxter (1995b) and Baxter (1995a), who simply considers regions of the Euclidean space to be labeled by the descriptor terms (in our case, words), but has the advantage of leaving it unspecified how the stimuli associated with initial examples are encoded. We will make considerable progress toward this goal in later chapters, but the take-home lesson should already be clear: the semantic space, in addition to Euclidean structure, must also carry some potential function P whose local minima are the targets of learning. How much of this function is universal (independent of the choice of language), perhaps even inborn, and how much must be inferred from data, is the question we turn to in the next chapter.



2.8 Further reading

The view of algebra as the study of structures is a relatively recent development, perhaps best dated from the appearance of *Moderne Algebra* (Van Der Waerden, 1930).

The downward Löwenheim–Skolem theorem is of considerable technical interest in logic, as it leads to the rather counterintuitive conclusion that the set of real numbers \mathbb{R} and even the axioms of set theory have a countable model (this latter fact is known as the [Skolem paradox](#)). We will not present the proof in detail here (see [Stanley Burris’ class notes](#) and the Wikipedia article on [Skolemization](#)) because our primary interest is in finite sets.

The *strength* of a system of logic is the class of structures it can define. For a more detailed statement and proof of Lindström’s Theorem, see [Nate Ackerman’s talk](#).

In philosophical logic, a [proposition](#) is often defined as the result of assigning meaning to a (declarative) sentence, and from this standpoint the whole propositional calculus can be viewed as a realization of the program of replacing the pretheoretical, informal idea of meaning by a more theoretical, fully formalized construct, that of the (closed) wff. In the model-theoretic approach to natural language semantics, this usage (equating ‘meaning’ or ‘sense’ with ‘proposition’) is often retained even when the logical calculus used is more complex than the propositional calculus set forth here. A similar terminological difficulty exists in regard to the meaning of noun phrases or *individual concepts* in that the class of mathematical objects offered as reconstructing the philosophical notion is by no means uniform – for a discussion, see McCarthy (1979).

There exist systems of logic weaker than FOL where the Inconsistency Lemma (also known as the [Principle of Explosion](#) or by its traditional Latin name as *ex falso quodlibet*) does not hold. Systems of deduction that will to some degree tolerate inconsistencies are of great technical interest: readers may wish to look at *paraconsistent logic* (Priest, 1979; Priest, Routley, and Norman, 1989). The [original proof](#) (Gödel, 1986) of the Validity Lemma is quite complex; for an illuminating discussion, see Chapter VI of Kleene (2002). For a proof that demonstrates the soundness and completeness of Hilbert-style deduction systems see Greg Restall’s [lecture](#).

There are many online courses exploring FOL, and the reader may wish to consult [Tarski’s World](#). We mostly use just propositional calculus (for which the completeness proof is [considerably simpler](#)), and we can rely on [minisat](#).

Approximation by finite Gaussian mixtures was pioneered by Pearson (1894). For a still valuable survey, see Titterton, Smith, and Makov (1985). For a more detailed introduction to PAC learning, see Chapter 7 of Mitchell (1997). For an introduction to PCA from an engineering perspective, see Apley (2003). Applications to social science are more controversial; see for example the discussion surrounding the [psychometric g factor](#), where Gould (1981) brought the charge of reification. For a comparison with the strongly related [factor analysis](#), see Arnold and Collins (1993). The PPMI-based method described in Example 2.3 is by no means the only way to obtain an embedding of words into Euclidean space: for a contemporary survey, see Levy, Goldberg, and Dagan (2015), for a detailed comparison with LSA, see Turney and Pantel (2010). The idea that word



Phil



Comp



meaning is highly correlated to word distribution goes back to [Firth](#), who famously said “You shall know a word by the company it keeps”. Modern implementations start with Schütze ([1993](#)).



Prolepsis

Contents

3.1	Understanding	48
3.2	The minimal theory	52
3.3	Space and time	54
3.4	Psychology	59
3.5	Rules	62
3.6	Regularities	65
3.7	The standard theory	70
3.8	Desiderata	75
3.9	Continuous vector space models	79
3.10	Further reading	84

Learning something is not a trivial act – the ancient Greeks were quite aware of the difficulties attendant to the creation of something from nothing. If knowledge in the learner’s head can be created from nothing, the floodgates are open, and all kinds of things can be created from nothing, contrary to everyday experience. Prolepsis, often translated as ‘foreshadowing’ or ‘preconception’, is a technical term originating with the Stoics, for whom it meant a naturally endowed and innate system of thought involving universal concepts. The roots of the idea go back to the Platonic method of ‘recollection’ (anamnesis), as exemplified in *Meno*, where Socrates teaches the slave boy that a square D built on the diagonal of a smaller base square B will have twice the area of B . Some form of [innatism](#) or [nativism](#) is argued for by many philosophers, from Leibniz and Descartes to Chomsky and Fodor.

The opposing view, that something can be created from nothing, is very hard to defend on philosophical grounds. Even the staunchest critics of innatism like Piaget (see Piattelli-Palmarini, Piaget, and Chomsky (1980)) are forced to admit that *some* part of our knowledge acquisition process is innate, especially in cases of higher functions such as self-awareness (see [Suddendorf and Collier-Baker, 2009](#)), where the biological underpinnings are clearly manifest in animals. More important than the philosophical weaknesses of the non-innatist (also known as *constructivist*) position is the total lack of machine learning systems capable of exhibiting constructive behavior – to the extent

Phil



Comp

that our focus is on algorithmic (mechanizable) theories of semantics, we may as well admit that all systems known to us have a non-negligible proleptic component.

As we have discussed in Section 2.7, adult speakers of a language have some concepts in their heads which are, to a surprising extent, independent of the actual examples that may have served as the basis for learning the words denoting these concepts. When we inquire about how children form these concepts, our goal is twofold: on the one hand, we want to understand what kind of prolepsis needs to be assumed to support the abstraction process whereby new concepts are acquired, and on the other we would like to see how the abstraction is actually performed. Clearly, the more we need to attribute to prolepsis the less interesting the resulting theory of learning will be: in the limiting case we could just assume that there is no learning whatsoever, just anamnesis.

In 3.1 we begin with a simple story, as analyzed by John McCarthy, and discuss what we mean by understanding this story. In 3.2 we turn to the larger issue of the prolepsis needed for supporting the kind of inferences required for understanding everyday language: at the very least, we will need some kind of *objects*. Smith and Casati quote Scanlon (1988):

Intrinsic to the natural concept of the world is the unshaken belief that all the component parts of my environment exist and develop, change or remain constant, in interaction with one another, in some form of stable regularity, all independently of my observing them or not observing them. (Scanlon, 1988, pp. 220f.)

Exactly what more we need is the question that we probe in 3.3, where we discuss the naive theory of space and time; in 3.4, where we turn to an analysis of (human) beings; and in 3.5, where the nature of rules is discussed. In 3.6 we summarize some of the basic mathematical apparatus pertaining to groups and (operator) semigroups that we will employ in subsequent chapters to capture the notion of rule-like regularity, and in 3.7 we discuss the standard logical theory of natural language semantics known as Montague grammar (MG). In 3.8 we summarize some general criteria of adequacy for a theory of meaning and discuss to what extent MG meets them.

We emphasize at the outset that, unlike our philosophical predecessors, in particular Aristotle and Kant, on whose work we rely very heavily, we do not approach the prolepsis with the goal of trying to understand the essential features of space, time, or moral behavior. Rather, our goal is to delineate the *minimal* formal theory that makes it possible to discuss such weighty issues in an axiomatic fashion. In this regard, the present work fits well into the formal approach to analytic philosophy originating with Leibniz and Spinoza, and owes a great deal to the 20th-century work on the subject by Russell, Carnap, and Montague. Compared with the work of these giants, who attempted to subsume both everyday and scientific language use under the same theory, our goals are more modest, concentrating entirely on everyday language at the expense of having anything to say about the foundations of scientific theories except, of course, the theory of language.

3.1 Understanding

We begin with a story from the *New York Times*, September 29 1973, selected by McCarthy (1976) “as a candidate for a target for a natural language understander. The story is about a real world event, and therefore the intentions of the author are less relevant for answering questions than for made up stories. The main goal of this discussion is to say what a person who has understood the story knows about the event. This seems to me to be preliminary to making programs that can understand.”

A 61-year old furniture salesman was pushed down the shaft of a freight elevator yesterday in his downtown Brooklyn store by two robbers while a third attempted to crush him with the elevator car because they were dissatisfied with the \$1,200 they had forced him to give them.

The buffer springs at the bottom of the shaft prevented the car from crushing the salesman, John J. Hug, after he was pushed from the first floor to the basement. The car stopped about 12 inches above him as he flattened himself at the bottom of the pit.

Mr. Hug was pinned in the shaft for about half an hour until his cries attracted the attention of a porter. The store at 340 Livingston Street is part of the Seaman’s Quality Furniture chain.

Mr. Hug was removed by members of the Police Emergency Squad and taken to Long Island College Hospital. He was badly shaken, but after being treated for scrapes of his left arm and for a spinal injury was released and went home. He lives at 62-01 69th Lane, Maspeth, Queens.

He has worked for seven years at the store, on the corner of Nevins Street, and this was the fourth time he had been held up in the store. The last time was about one year ago, when his right arm was slashed by a knife-wielding robber.

To quote McCarthy further: “An intelligent person or program should be able to answer the following questions based on the information in the story:

1. Who was in the store when the events began? Probably Mr. Hug alone. Although the robbers might have been waiting for him, but if so, this would have probably been stated. What did the porter say to the robbers? Nothing, because the robbers left before he came.
2. Who was in the store during the attempt to kill Mr. Hug? Mr. Hug and the robbers.
3. Who had the money at the end? The robbers.
4. Is Mr. Hug alive today? Yes, unless something else has happened to him.
5. How did Mr. Hug get hurt? Probably when he hit the bottom of the shaft.
6. Where is Mr. Hug’s home? (A question whose answer requires a literal understanding of only one sentence of the story.) Does Mr. Hug live in Brooklyn? No, he lives in Queens.
7. What are the names and addresses of the robbers? This information is not available.

8. Was Mr. Hug conscious after the robbers left? Yes, he cried out and his cries were heard.
9. What would have happened if Mr. Hug had not flattened himself at the bottom of the pit? What would have happened if there were no buffer springs? Mr. Hug would have been crushed.
10. Did Mr. Hug want to be crushed? No.
11. Did the robbers tell Mr. Hug their names? No.”

McCarthy goes on to ask several more questions that we will deal with later, but this is already sufficient to see one of his main points: we would like to test the understanding that a computer has in the same way we test the comprehension abilities of schoolchildren, by asking (natural language) questions and expecting (natural language) answers that reflect their ability not just to paraphrase but also to integrate knowledge gained from the text with their pre-existing knowledge and to draw inferences. We will discuss a problem set that is designed to specifically probe this ability in Section 7.1.

A full solution will therefore rely not just on natural language parsing and generation abilities, but also on a method, or several methods, of drawing inferences from various sets of axioms. For the time being, we will use FOL as our scheme of representing knowledge and drawing inferences, not because we believe it to be ideal for this task (in fact, McCarthy developed several arguments in this paper and elsewhere to show that it is not), but rather because it happens to be the most widely used formalism in knowledge representation. In fact, FOL is something of a continental divide among the frameworks of logic we could use. From a mathematical perspective, FOL is a small system, considering that the language of set theory requires only one binary relation, \in , and it is evident both from the Peano and the ZF axioms that we will need all well-formed formulas (or at least the fragment that has no atomic sentence lying in the scope of more than three quantifiers, see Tarski and Givant (1987)) to do arithmetic. Therefore, those who believe that mathematics is but a small, clean, well-organized segment of natural language will search for the appropriate semantics somewhere upwards of FOL – this is the MG tradition we will discuss in Section 3.7, where higher-order intensional logic is viewed as essential. There has already been significant work on trying to restrict the power of the Turing-complete higher-order intensional apparatus to FOL (Blackburn and Bos, 2005), and in Chapter 5 we take this further, moving to formalisms that fall at the low end of the complexity scale, well below FOL. At that point, much of what mathematical logic offers is not applicable, and the algebraic methods introduced in Section 3.6 have more traction.



Fig. 3.1. Where is the shaft? What buffer springs?

Let us begin by considering what a highly logical and intelligent person, but not one familiar with modern life, say Mr. George Boole, would have made of the above story. In a critical respect, understanding what happened requires understanding the geometry of the *freight elevator*, with the *shaft* and the *buffer springs* at the bottom. For Mr. Boole, whose concept of the freight elevator, complete with [ropes and pulleys](#), is illustrated in Fig. 3.1, the story makes little sense.



The contemporary reader has no problem responding to questions similar to Question 9 above: What if Mr. Hug had fallen on top of the buffer springs? He would have been crushed by the downcoming elevator. What if there were no buffer springs? He would have been crushed because the elevator bottom could have come all the way down. Notice that the last sentence, about Mr. Hug's right arm getting slashed by a knife-wielding robber, would have caused no problem to Mr. Boole, even though the syntax is just as tricky (really, it is the knife that has the appropriate geometry for slashing, not the robber), because the knowledge required to understand this part about *robbers wielding knives* and *knives slashing bodies* was already at his disposal.

We thus see that the problem of understanding the story can be divided into at least two major parts: first, a generic reasoning capability about shapes, rigid bodies (elevators, knives), soft bodies (arms, torsos), and their interactions, and second, the rest. The first part, which is also responsible for our understanding of who was in the store, out of the store, at the elevator door, under the elevator, on the first floor, in the basement, and so on, will be called *naive space–time geometry*. This is a highly sophisticated body of knowledge, and clearly one that is to a significant extent already at the disposal of the crocodile that knows how to entrap its prey in the shallows. To make use of this knowledge, the semantics must link objects to geometrical shapes in a manner that is so seamless as to be nearly invisible: we need to invoke persons from different ages and cultures to realize that the knowledge that the elevator bottom cannot touch the shaft bottom because of the positioning of the buffer springs or that Queens is not part of Brooklyn is something that is not inborn.

We defer discussing naive space–time geometry to Section 3.3, but note here that this theory, having been under immense selectional pressure for billions of years, is highly sophisticated, and a good understanding of Euclidean space–time and Newtonian physics is neither necessary nor sufficient for our goal of supporting commonsensical inferencing of the kind McCarthy's questions are probing, even if we steer clear of quantum phenomena, electromagnetism, and speeds approaching that of light. The key element of prolepsis we need to take home from the above discussion is that there are *objects*, and that objects can have *attributes* such as *hardness* and *shape*. Many of these attributes, such as *color*, *smell*, or *function*, will have their own naive theories, and we can easily imagine stories whose comprehension relies on these naive theories just as heavily as the comprehension of Mr. Hug's story relies on naive geometry.

Exercise \rightarrow **3.1** Study the logic of great detectives like Sherlock Holmes or Miss Marple. What kind of reasoning do they employ? Does finding solutions to differential equations play a role in this process?

Question 10 asks whether Mr. Hug wanted to be crushed. This question presumes the existence of needs and wants in people’s heads; in fact it presumes a whole theory akin to naive geometry that we will call *naive psychology*, which we defer to Section 3.4. When we answer this question in the negative we rely not just on sympathetic understanding (since we would not want to get crushed, we presume other people would not want to either), or on broad axiomatic descriptions of behavior (crushing is bodily harm, people normally avoid bodily harm), but also on textual evidence: Mr. Hug *flattened himself* at the bottom of the shaft, an act that can only be explained as self-defense.

Exercise[†] 3.2 Write down the steps of reasoning starting from the premiss that Mr. Hug flattened himself at the bottom of the shaft and leading to the conclusion that he didn’t want to get crushed. What axioms do you need to invoke?

3.2 The minimal theory

As the preceding discussion makes clear, part of our goal must be to develop a theory capable of handling the kind of commonsensical inferences that people routinely, automatically, and generally subconsciously make when answering simple questions about simple stories. We may have more ambitious goals, demanding the same sort of understanding in regard to comic book stories, theatrical performances, or movies, which have an increasingly powerful visual element at the expense of a straight (linear) narrative and can rely on non-linguistic genre conventions. We may even look for a general theory of comprehension that can take real-life sequences of events (in the limiting case, just the uninterpreted sense data) as input and provide the means for answering questions about these. Since this would bring in a whole range of foundational questions concerning what constitutes an event, and an equally, if not more, complex set of pattern recognition issues, we stay with the less ambitious goal of dealing with purely linguistic input, where each event is signaled by a main verb. We still have the problem of (0) syntax-driven analysis, extracting information from natural language expressions (both from the text to be understood and from the queries), and the task of natural language generation, converting the results of the inference process back to natural language. Our theory also has to provide (1) a means of representing everyday knowledge, such as the naive geometry and psychology introduced in Section 3.1; (2) a means of representing the information communicated by the text to be understood; and (3) an inference mechanism that lets us combine the two to yield additional knowledge.

In our example, (1) would contain axioms like *Getting crushed is suffering bodily harm*, *Animals don’t want to suffer bodily harm*, and *Humans are animals*; (2) would contain statements such as *Mr. Hug is a human*, *Mr. Hug flattened himself* and the inference mechanism (3) would lead to conclusions such as *Mr. Hug didn’t want to get crushed*. To make this work in FOL we need to take several technical steps. First, we will need complex predicates such as `getting_crushed` and `suffering_bodily_harm` so that we can say $\forall x \text{ getting_crushed}(x) \rightarrow \text{suffering_bodily_harm}(x)$. This leads directly to the question of whether we

have yet more complex predicates such as `want_to_suffer_bodily_harm` or whether *want* is a higher-order operator on predicates. Taking this latter option has some disadvantages; in particular, it will be very hard to guarantee that the system stays first order in the face of statements like *Everything a mother wants for herself she wants even more strongly for her children*. But taking the former option has even more undesirable consequences in terms of psychological reality: once `want_to_have_ice_cream` and `want_to_get_a_pay_raise` are all complex predicates stored in the system, one needs to develop a whole theory of how such predicates are acquired by the language learner, machine or human. This is where the issue of compositionality, already broached in the Preface, comes in: the only reasonable hypothesis is that *wanting to have ice cream* is composed of the meanings of `want` and `have_ice_cream`, and similarly for *wanting to get a pay raise* and (not) *wanting to get crushed*. Looked from this perspective, *want* is a strange kind of operator, in that $x \text{ WANT } y$ may be true, and z may necessarily follow from y , yet $x \text{ WANT } z$ may be completely false: for example $x \text{ WANT to_smoke}$ does not imply $x \text{ WANT to_risk_cancer}$. Even more strongly, y and z may be strictly equivalent, such as the Morning Star and the Evening Star (although the ancient Greeks, before Pythagoras, were not aware that the two were actually the same object, which we call the planet Venus today), yet a person wanting to see one may not necessarily want to see the other. This problem of *opacity* is very important for the standard model-theoretic view of semantics, but will be less important for us for reasons discussed in Chapters 3.9 and 5.6.

The final piece of the prolepsis we have to consider is the issue of *valuation*. Natural language statements are not just true or false, analytic or synthetic, a priori or a posteriori, but also good or bad, cheerful or sad, attractive or repulsive, honorable or shameful, and we have no problem rating them on Osgood-style scales (Section 2.7). It is often said that such valuations are always subjective (can only be made with reference to an individual who considers them being that way), while truth and falsity are objective properties of propositions, requiring no reference to individuals. On closer inspection this view appears untenable: clearly the majority of the propositions one encounters will be true or false only relative to a given system of axioms, and even seemingly unshakeable truisms like $2 + 2 = 4$ depend on tacit assumptions such as the Peano axioms. Perhaps logical tautologies are an exception, but the choice of logic matters, and in many important cases it is undecidable whether a given formula constitutes a tautology. Things are not that simple in the domain of ‘empirical facts’ either: is Pluto a planet? Overall, our only known theory of truth makes this notion highly dependent either on some theory (set of axioms) or on some (structured or unstructured) set of models. Here we will take a mechanistic view of individuals, treating them as no different from the set of valuations they hold, or, what is the same, we take a rather animistic view of theories, regarding them as Platonic individuals. Either way, we assume that *some* valuation, or at the very least, the potential to be endowed with some valuation, is present in all propositions.

At minimum, a valuation is some function that a theory may impose on propositions, with a *value* being a member of a binary opposition such as true/false, good/bad, plea-

surable/painful etc. We may wish for a far richer notion of value, for example when we discuss probability we may take the value from \mathbb{R} rather than \mathbb{B} . Since our goal is to minimize prolepsis, we will not assume \mathbb{R} as a universal concept that is part of our naturally endowed (innate) system of thought. In this respect, we also part ways with Kant, for whom Euclidean space and time are part of the prolepsis. In Section 3.3 we will begin to develop our own naive theory of space and time, but in the prolepsis we refrain from positing even the existence, let alone the precise characteristics, of space/time. Complex valuation strategies will be further discussed in Section 7.1.

Whether a simpler theory of valuation than binary is possible is not clear at the outset, but clearly mapping everything on the same element will not create a nontrivial partial order. Thus in the prolepsis we need to assume the existence of at least two things. That this is not a self-evident assumption was already clear to Plotinus, who saw the difficulties of creating Many from One on the same level as creating something from nothing (*Enneads* V.2). Here we make no such claim as actually *creating* two things, we simply restrict attention to models that have at least two elements, since there is not much we can, or need to, say about models with only one element.



3.3 Space and time

To round out the picture emerging from the preceding considerations, we need to populate our universe. We assume *objects*, and we assume that at least some of these objects are *agents*, capable of purposive behavior. There may also be blind forces of nature, such as winds or fires, which are capable of acting (bringing about changes in the states of other objects and themselves), but we do not at the outset assume that such acts are ‘free’ or ‘voluntary’. Our primary interest is with agents endowed with sensory apparatus, as opposed to ‘blind’, and with goals, i.e. purposive behavior, as opposed to involuntary actions. Here we begin by cataloging the basic predicates one needs to be able to investigate such objects. Syntactically, we will distinguish unary and binary predicates, the former typeset in `typewriter font`, the latter sometimes in `SMALL CAPS` (see Section 6.4). The unaries will be prefixed to variables, the binaries will be infix, following *Subject Verb Object* (SVO) order. For ease of reading, time and again we will find it expedient to add person and number suffixes, typeset in normal font.



We begin with animals, because this is a domain whose naive theory is well understood by every reader. First we will have the *dog*, which is `four-legged, animal, hairy, barks, bites, faithful, and inferior`; and the *fox*, which is `four-legged, animal, hairy, red, clever`.

Exercise^o 3.3 Define *horse* and *donkey* in a similar fashion.

We make purposely very little distinction between an individual dog, the species *Canis lupus familiaris*, and the set of dogs in the world. (One important technical device, the potential set of dogs in all possible worlds, will be discussed in Section 3.7.) What we are primarily interested in is the everyday, commonsensical notion of what the word *dog*

means, and this is perhaps best approximated by the Platonic idea of the dog as a ‘typical’ individual, which has all the essential properties of dogs, and only those properties.

Exercise → 3.4 Is the Platonic dog male or female?

We steer clear of the scientific theory of dogs. We are not interested in the issue of whether dogs are truly a species or just a subspecies, and how this (sub)species is to be defined, for example, in terms of its DNA. Clearly, people could talk about dogs and all dog-related activities for millennia without the benefit of a scientific theory, and, equally clearly, the scientific theory falls short of explaining even the simplest facts of language use, for example that comparing someone to a dog is an insult, while comparing them to a tiger is not. We also disown the philosophy of [essentialism](#) in that what we say about essential features applies to *words*, not the things they name. It would actually make sense, from both the scientific and the philosophical standpoint, to see their DNA as descriptive of the essence of doghood, but this is not what language use takes as essential, and it is the facts of language use that we subject to an essentialist analysis here.

Let us now consider *mule*, which is animal, cross between horses and donkeys, stubborn. Since being a ‘cross between horses and donkeys’ is the most important, possibly the only, defining property of a mule, we will need a theory that can express this notion as a combination of more primitive notions. Before turning to this, we note that not every combination of concepts is expected to yield a new concept with a practical use.

Exercise ° 3.5 Define some concepts that are not in use. Define some concepts that have no use.

Conversely, not every concept is defined by the conjunction of other concepts, primitive or derived. What the *mule* example shows is that definitions will require some mechanism that goes beyond conjunction. We take this mechanism to be *function application*, so that expressions like cross between horses and donkeys can be further analyzed as donkey FATHER mule and horse MOTHER mule. The key observation is not so much the fact that functions are used as the fact that they are used in equational definition: whatever x a mule is, it is such a thing that donkey is the FATHER of x . This would be easy to express in FOL by writing $\forall x \text{ mule}(x) \exists y, z \text{ horse}(y) \wedge \text{female}(y) \wedge \text{donkey}(z) \wedge \text{male}(z) \wedge \text{parent}(x, y) \wedge \text{parent}(x, z)$, but for the same reasons as above, we do not assume variables in the prolepsis: the formal method for avoiding them will be described in Section 4.6.

In order to have functions and function application of some sort, but no variables, we take a particularly tame version of functions, [finite state transducers \(FSTs\)](#).

Definition 3.1 An FST is given by a finite set of *states* S , a finite set Σ of *inputs*, and a finite set Γ of *outputs*. An FST is capable of changing state in a manner that is determined by table lookup based on its current state and current input alone. (The state change may, but need not, be fully synchronized with consuming input or producing output, a matter we return to in Chapter 4.)

Phil



One way to think of FSTs is simply as a model of living beings characterized by a very simple but robust sensory apparatus that is capable of distinguishing one of finitely many inputs from Σ , independent of what state the FST is in, and a very simple effector system that is capable of producing signals from Γ . Note that FSTs are not intended as fully realistic, direct models of (human) beings. For example, we know by introspection that humans are capable of getting into states where their sensory system shuts down, something the FST model has no provisions for. Also, it will require significant modeling work to be able to claim that FSTs can exhibit goal-directed behavior, can have needs, desires, intentions, and so on. Before turning to these and similar issues of naive psychology, let us complete the basic theory of space and time. By assuming FSTs, we assume the existence of *state spaces* composed of elementary states (deterministic or nondeterministic), which we will call *loci*. To fully define an FST we need to specify what input or series of inputs will move the FST from one locus to another, and also to specify what output, if any, is produced by such a move. Looking at this from the other side, any given FST defines a state space with some characteristics, and we can use FSTs to give a specific meaning to a rather simple, discrete sense of space.

Example 3.1 The Useful Pot (Fig. 3.2). There are only two states, which we call `burst_balloon_out` and `burst_balloon_in`, and we number them 0 and 1, respectively. There are two inputs, which we call `Eeyore_put_burst_balloon_in` and `Eeyore_take_burst_balloon_out` and abbreviate as p and t , respectively. p has the effect of moving the locus from 0 to 1 and t moves it from 1 to 0. In state 1, p has no effect and in state 0 t has no effect. There are no output signals.

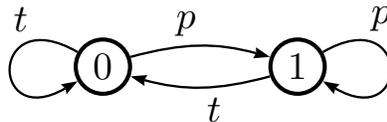


Fig. 3.2. The Useful Pot

For the moment, we ignore the facts that it is not just burst balloons that could be stored in Useful Pots, it is not just Eeyore who can put these in or take them out, and that putting something in a pot has effects beyond filling the pot, such as rendering the object invisible from the outside.

Exercise[†] 3.6 Build a more realistic model of pots, with provisions for filling and emptying them gradually. Try not to rely on complex notions like volume integrals which are unlikely to be proleptic.

In Section 3.6 we will return to the research program of using state spaces to define not just primitive spatial relations like IN and OUT but also more complex ones. We note that FSTs have broad implications for the modeling of time as well. For explicit models,

it is easy to build FSTs corresponding to the weekly cycle of days, the daily cycle of hours, and so forth.

Exercise^o 3.7 Build a calendar model that has a leap year every four years, except that every hundred years the leap year is omitted, but every four hundred years the leap year that would ordinarily be omitted is nevertheless retained. In combination with subsidiary day models for months, an hour model for days, a minute model for an hour, and a second model for a minute, how many states does the resulting overall calendar model have? How precise is this model?

Actually, our main interest is not so much in explicit models of time as in the model already implicit in the FST conception. The mere fact that input signals are received one after the other, never in parallel, implies that time is conceptualized in a discrete, sequential fashion. In particular, there is an elementary succession step between state changes we call a *transition*. In a **Mealy machine**, which is the definition we will use throughout this book, both input and output are synchronized with the transition, while in a **Moore machine** only inputs are tied to transitions, outputs are tied to states. An interesting case, discussed further in Chapter 4, is the Hidden Markov Model (HMM), where moves between states have probabilities attached to them, but outputs are tied to states. Strictly speaking, HMMs have no input as such – when we conceptualize them as Moore machines, the only inputs are *time ticks*, which permit (or force) the machine to move into a new state and emit a new output.

Exercise^o 3.8 Each Mealy (Moore) machine computes a relation between input and output strings. Is there a relation that is computable by Mealy (Moore) machines but not by Moore (Mealy) machines?

The key temporal notion in the prolepsis is not so much the idea of *time* itself as the idea of a *process*. It seems that humans (and in all likelihood, all mammals) are endowed with a perceptual mechanism that inevitably makes them perceive certain sensory inputs as processes. Try as we might, we cannot perceive the flight of the arrow as a series of states: what we see is a continuous process. The compulsion to do so is so strong that even truly discrete sequences of inputs, such as frames of a movie, will be perceived as continuous, as long as the frame rate is reasonably high, say 20/sec.

That naive physical processes (not just motion, but all changes of state such as the ripening of fruits) are continuous is hardly debatable. Zeno noted that to sustain a continuous model of processes by a discrete model of time is not trivial, for if time is composed of instances, when does the arrow move? Not in an instance, since any instance is static, and not between instances, since if time is exhaustively composed of instances as a line is exhaustively composed of points, then there is nothing between instances. The standard theory, beginning with Dedekind, Cantor, and Weierstrass, offers a sophisticated theory of the *time line*, which is indeed composed of instances the same way a line is composed of points. The real line \mathbb{R} does double duty, supporting both time instances and spatial instances, and the motion of the arrow, conceptualized as an $\mathbb{R} \rightarrow \mathbb{R}$ function, is indeed composed of time instances where the arrow stays at a fixed spatial point, yet makes possible the measurement of continuous motion, including the measurement



of speed. This is no doubt one of the crowning achievements of mathematical physics, but to say that it is part of the prolepsis would really be, to borrow a phrase from Smith and Casati (1994), the ‘shamefacedly counter-commonsensical set-theoretic translation’ of the primitives.

The arrow paradox, in our analysis, is due primarily to another part of the naive concept of time, the assumption of a *next* instance. Neither under the standard ϵ, δ notion of continuity nor in nonstandard analysis (which is in many ways closer to the naive picture, though the mathematical foundations are even more counterintuitive) is there such a thing as a next time instance. On the contrary, time in these theories is infinitely divisible; there will be another time instance (in fact, there will be infinitely many) between any two instances. This is analogous to the apparent infinite divisibility of fluids like water. As Feynman (1965) has it:

Suppose we have a drop of water, a quarter of an inch on the side. If we look at it very closely we see nothing but water – smooth, continuous water. Even if we magnify it with the best optical microscope available – roughly two thousand times – then the water drop will be roughly forty feet across, about as big as a large room, and if we looked rather closely, we would *still* see relatively smooth water [...] Look still more closely at the water material itself, magnifying it two thousand times again. Now the drop of water extends about fifteen miles across, and if we look very closely at it we see a kind of teeming, something which no longer has a smooth appearance – it looks something like a crowd at a football game as seen from a very great distance.



The foundational theory needed to sustain this kind of infinite divisibility, [mereology](#), rests on a continuous notion of *part of* that is to be contrasted with the standard, discrete notion. For typical solid objects, be they rigid or elastic, we have parts that no longer fulfill the definition of the whole: the head of a horse is no longer a horse, a small enough chip of a granite rock is no longer a rock but dust, and so on. The standard theory of sets formalizes this by means of the [Axiom of Foundation](#), which implies that there is no infinite descending *part of* chain. The continuous case, where part of water is such that we can take even smaller parts, is best stated with the aid of the [Anti-Foundation Axiom](#) (AFA). We emphasize again that we are not interested in the scientific theory of water which includes H₂O molecules that no longer have parts that can be called water – on the contrary, our interest is in the naive theory of water (and of time and space), which permits arbitrary divisibility. Since in mereology we can still have well-founded sets while in standard set theory we cannot have non-well-founded sets, it appears that for set-theoretical foundations we are better off with mereology than with standard set theory.

Our way out of the arrow paradox will also do away with the requirement that instances are point-like: in nonstandard terms, we will assume that instances are already infinitesimally small intervals. We also do away with the assumption that such instances form an exhaustive partitioning of time; again in modern terminology, we take them to

be just discrete samples from a continuous process. It is only perceptual blurring together of these samples that we must declare proleptic; the rest is learnable. In Chapter 4 and beyond we will broaden this narrow foundation to a fuller theory of *states of affairs* and *actions* connecting these, but as long as the details of the temporal signature are irrelevant we will simply call both of these ‘matters’.

One notion that we will come across later is that of *fluents*, logical variables that are tied to time. Is this berry edible? Well, there is a time of the year when it is; the rest of the time it is either not there or not ripe. Often we are faced with the question of how well a set of behavior patterns will serve a particular goal or set of goals. In logic, our first impulse is to model such questions as implications, automatically moving the time condition into the premiss, ‘if August, edible’. Fluents offer a more natural way of handling such cases, even if we have a discrete underlying time, rather than \mathbb{R} as is common in the *event calculus*.



3.4 Psychology

By the same token by which (naive) physical states of a system can be modeled by FSTs, we can use FSTs to model mental states such as being happy or angry. We will explore such naive models in greater detail in Section 3.6; here we focus on the proleptic aspects. The key idea is to equate nondeterminism, in the technical sense used in computer science since Rabin and Scott (1959), with *free will*. This latter term, coming with a great deal of baggage from psychology and philosophy, is not amenable to a simple, unified analysis because different authors mean different, often strikingly different things by it, but our goal, as usual, is to reconstruct the everyday, commonsensical meaning. There are two rather strong theses here: first, that nondeterministic behavior merits the name free will, and second, that there is nothing in the notion of free will that is not amenable to analysis in terms of nondeterministic automata.



As for the first thesis, free will occupies such a pivotal position in philosophy because it has been realized at least since Aristotle (Nicomachean Ethics, Bk. 3) that only those acts where conscious choice is possible are subject to moral judgment. Thus when we say that automata are capable of nondeterministic behavior, we are claiming, among other things, that they can satisfy at least some of the conditions necessary for moral judgment and virtuous conduct and thus, according to Plato and Aristotle, happiness (eudaimonia), a matter we shall return to at the end of this book in Chapter 9.

As for the second thesis, this clearly depends on one’s notion of free will. To clear away some of the conceptual underbrush surrounding the issue, we shall use the example of an individual wishing to go from station A to station B in a city with a subway system. Even if the subway is running entirely deterministically on a prescribed schedule, our person can still make some elementary choices such as getting on a train or not, so in the end she can still get anywhere she wants to. Her free will is limited only in that the travel *time* from A to B, even with full knowledge of the schedule and an ability to make optimal choices, will depend on facts outside her control. Our conclusions from this

small example are twofold: first, that assuming free will is not the same as assuming total control, and second, that free will in larger things requires the ability to make elementary choices.



The much-discussed Conway-Kochen [Free Will Theorem](#) asserts that under reasonable conditions elementary choices can be pushed down to the quantum level: if people can have nondeterministic behavior so can elementary particles. There is a logically equivalent theorem which appears trivial: if each part of a system is deterministic, the entire system will be deterministic. The value of the Conway-Kochen Theorem is in proving this assuming axioms that fit well with quantum physics, but this is tangential to our enterprise in that quantum physics in general, and the three axioms used by Conway and Kochen in particular, is obviously beyond the prolepsis. Here we may as well assume directly that (i) there exists some nondeterminism in the world and that (ii) such nondeterminism can be amplified to human behavior.

An elaborate defense of assumption (ii) is made by Penrose (1989), but we believe that exhibiting the exact mechanism whereby quantum-level indeterminacy can lead to free will is not really necessary, because proprioception of free will is an empirical given. We are absolutely confident, based on primary sensory data, that boiling water will burn our skin. If the complete causal chain from heated nerve endings to the subjective sensation of burning pain could be exhibited, this would have far-reaching implications for example for the design of painkillers, implications that the naive theory lacks, so in this sense the detailed theory is superior to the un-analyzed statement. But our confidence in the original statement is already absolute, so for the purpose of accepting it as an axiom (for example in order to derive elementary guidance for behavior like ‘don’t immerse your hand in boiling water’) the details of the causal chain are irrelevant. Since our confidence in the existence of free will comes from primary (macroscopic) sensory data, we are willing to bite the bullet and assume that elementary particles can exhibit free will.



More precisely, particles are assumed to have *Willkür* or ‘power of (arbitrary) choice’, a notion that is viewed by Kant (1793) as secondary in his analysis of *Wille* which is purposive choice, the elementary building block of ethical behavior. Trains go in all directions, and we have a choice in picking the one that takes us closer to our destination. What the Kantian analysis demands is a *being* (Kant calls him *Man*, but neither male gender nor human genetic material seems essential to the analysis) capable of stating rules, performing actions, and being capable of judging whether a given action fits or violates a given rule. To formalize Kant’s analysis, we assume that sensory feedback is instantaneous, and that both motor action and drawing conclusions take positive time. We will not consider exactly how much time, not because the question is uninteresting, but because the issue is irrelevant to an understanding of Kant’s moral philosophy, which neglects the issue of resource bounds.

We assume that the *mind* of a being has a large but finite state space and, further, that some (but not all) states come intrinsically marked with *pain* or *pleasure*. An early rule of behavior is pain avoidance: if action X leads from a painful state S to a painless state S’, while action Y leads from S to another painful state S’’, a being subject

to this rule will prefer action X to action Y. A rule of pleasure seeking can be similarly formulated. We do not assume that these rules are hard and fast; it may well be that a being considers inflicting pain on itself to be good for some higher reason.

Exercise[†] 3.9 Refine the analysis to include degrees of pain and pleasure.

The pain/pleasure valuation is largely fixed. A human being may have the power to acquire new tastes, and make similar small modifications around the edges, but key values, such as the fact that harming or destroying sensors and effectors is painful, can not be changed. Instead, individuals are capable of assigning their own valuation to the states, i.e. a partial mapping from states to *another* linear order, which we will call *value* and assume that it has at least three gradations, positive, neutral, and negative. Beings can be purely hedonistic (assigning high value to states iff they are pleasurable) but they need not be – they have the freedom to assign high value to painful states. Nor is there a requirement of consistency: it may well be the case that state S is inevitably followed by state S' and our being values S positively and S' negatively or the other way around. A being is also free not to avail itself of the valuation mechanism at all, assigning no value to any state or, what is the same, assigning the same value to all states. Further valuations, such as ranking states in terms of esthetic value ('X is more beautiful than Y') or any other consideration (X serves the national interest better than Y, X contributes more to global warming than Y, etc. etc.) are all possible, and we make no requirement that these be consonant, i.e. we admit the possibility that $X > Y$ on some scale and $X < Y$ on some other scale.

Now we are in a position to recapitulate Kant's starting point: beings, as defined above, are free to select a valuation and to bind themselves to it in the sense that for every starting state S and alternative continuations X and Y, if they value X over Y they will follow X *unconditionally*. It is customary in the philosophical literature (though not in the writings of Kant especially) to call such an unconditional binding an *obligation* and speak of its force using the modal *ought*. Thus one is obliged not to say falsehoods, one ought to keep one's promises, etc. By undertaking obligations, beings reduce their free will to those remaining cases where X and Y are valued (by them) equally. This model includes as a special case *adeontic* beings, who can be said to unconditionally adhere to the empty valuation.

Another point that appears crucial to naive psychology is the concept of the *self*. For our purposes it will be sufficient to assume that the *mind* of beings contains some concept of the being itself, a concept we may as well call *self*. This self is just another FST, and since there is no requirement for it to be strictly isomorphic to the being (our model of the self can be, and often is, quite imperfect), there is no infinite regress or *homunculus argument*. It is worth emphasizing that our goals are considerably more modest than those of philosophers and cognitive scientists who wish to account for a broad range of human psychology – our primary focus is with linguistic expressions that either contain the morpheme *self* or give some overt indication of our state of mind, a matter we shall return to in Section 5.6.



Just as we are incapable of perceiving a movie as being composed of still pictures, we are normally incapable of perceiving ourselves in the world without presuming some soul or homunculus in which our consciousness is lodged. And, just as in the case of boiling water burning our skin, we are not required to exhibit a full scientific theory of how this may come about. Again, this is not to deny that for many purposes such a detailed theory would be superior to the un-analyzed statement, but our confidence in the original statement is already absolute, so for the purpose of accepting it as an axiom (for example, in order to derive elementary guidance for behavior) the details of the causal chain are irrelevant. Our confidence in the existence of self comes from primary sensory data, and assuming that *being HAS self* entails no loss of generality. These views are, of course, very close to those of Aristotle and Locke.

3.5 Rules

On a very large evolutionary timescale even the prolepsis may be learnable, if not by an individual, at least by a species. Here we are concerned with learning *given* the prolepsis, that is, learning particular FSTs (objects), valuations (partial mappings from objects to objects), and *rules*, to which we shall turn shortly. In practice, human beings take many years to develop a valuation. Kant abstracts away from this developmental process, concentrating on an idealized being prior to the moment of its undertaking a set of obligations. What makes Richard determined to prove a villain, and another person committed to helping the poor? Can there be principles that pure reason compels us to embrace, or are there irreducible rules of morality whose only support is divine decree, requiring an act of faith on the part of an otherwise rational being? Kant casts this as an epistemological problem: if there are some candidate rules, for example, ‘I ought to work for the benefit of others’ or ‘I ought to work for my own benefit’, how can we know which is ultimately the right one? Is it sufficient that our heart tells us, rather clearly, that the altruistic maxim is more noble than the egotistic one?

The question is much harder than it first appears. As Bayles (1968) notes, “It would seem that [egoism] would often result in severe competition between people, since each person would be out to get the most good for himself, and this might involve his depriving others. However, serious defenders of egoism, for example Hobbes and Spinoza, have generally held that upon a rational examination of the human situation it appears one best promotes his own interest by co-operating with others.” We do not deny the role of direct intuition concerning right and wrong, just as we do not deny the role of sensory input concerning hot and cold. However, when Kant wants to develop a theory of right and wrong, it is not his goal to account for all human intuitions concerning morality, just as a physicist interested in thermodynamics will leave many interesting observations about human perception of hot and cold to the study of psychology and physiology. It is of course desirable for the theory to be broadly consistent at least with the strongest human sensations, but there is room for discrepancies, and there is nothing wrong with either correcting human perception (for example, that ‘burning’ from dry ice

is really not burning but freezing) or employing subsidiary theories specifically aimed at resolving the remaining discrepancies (see Chapter 9 for further discussion).

As for the epistemological problem, there are essentially three approaches to learning about anything. First, we may simply accept the word of others, an approach we will call *learning by tradition*. Second, we may wish to make observations and conduct experiments, an approach we will call *learning by induction*. Third, we may use our reasoning facilities, an approach we will call *learning by deduction*. Kant of course is extremely familiar with Scripture and church doctrine, vastly more so than modern-day scientists or philosophers, and there is little doubt that his own heuristics for arriving at the particular maxims he advocates are to some extent guided by critical analysis of revealed teachings. He is also a giant of induction, having, by the time of writing *Religion*, already provided an incredibly broad and deep analysis of space, time, and causation in the *Critiques*. Yet when Martin Luther says “Reason is the greatest enemy that faith has; it never comes to the aid of spiritual things, but – more frequently than not – struggles against the divine Word, treating with contempt all that emanates from God” this could not be farther from Kant’s method, which is primarily deductive, seeking to justify religious doctrine by force of reason wherever possible. When he demonstrates that a particular doctrine such as that of original sin can *not* be arrived at by reason alone, Kant takes this as a justification of learning by tradition, rather than as a step toward demolishing the doctrine in question.

Aristotle’s theory of learning relies very heavily on his distinction between objects, which have physical weight and geometrical extent, or, in short, *corporeality* on the one hand, and Platonic forms, which lack corporeality, on the other. While the object itself is a composition of form and matter, to learn about an object the two must be separated, and it is only the form that is transmitted to the learner. A being is potentially a knower because she has the power to receive such forms. Let us for a moment compare this to the modern, model-theoretic account of learning, in which acquiring the meaning of a concept like *dog* requires knowing of each object, not just in this world but in every possible world, whether it is a dog or not. It is clear that in a resource-bound world the Aristotelian account has a far greater chance of success, in that it requires only the transmission of a finite (and, as our analysis in Section 3.3 makes clear, rather small) amount of information.

Aristotle makes a clear distinction between forms, as discussed above, and sensory inputs, which he considers corporeal. Our perception faculties (called *phantasia* originally) generate *phantasms*, which give rise to knowledge (or, rather, beliefs) of an inferior sort: we may see a snake in the garden, but upon closer examination we may find that there is only a twisted rope there. In fact, Aristotle considers the whole sensory apparatus corporeal, and takes it to be responsible for knowing any particular physical object, and takes intellectual cognition to be responsible for learning concepts. Here we are less concerned with the corporeal/incorporeal distinction, if only because computers have provided us with a convenient example where the distinction makes no difference. For Aristotle, Kant, or any other philosopher before the advent of computers, something



like the [Euclidean Algorithm](#) was incorporeal: it is an idea whose constituent parts, including the inputs and the outputs, are also ideas. Today, we may still see a particular embodiment of this idea as some kind of patterns of holes on punchcards, or as patterns of electron flow among circuit elements, but we have come to accept that the corporeal versions are largely immaterial to an understanding of what is happening – for pragmatic reasons, we are all Platonists now.



Given the ease with which we move between isomorphic versions of the same idea, it is clear that the Aristotelian view of communication, best articulated in [Locke's *Essay* \(1690\)](#), will serve rather well in the prolepsis. We will assume that beings have minds, and that their minds contain ideas, more complex thoughts that are formed from elementary concepts. Semantic generation is simply the encoding of these ideas in natural language; parsing is the decoding. We take communication to be a process of 'telementation', whereby ideas from one person's head move to the head of the other person. Our contemporary notions of what constitutes linguistic structure, beyond the mere succession of words, are considerably more sophisticated than those of Locke, but at the proleptic level we need no more than the idea that sounds can carry information. That humans are semantic devices, seeking meaning in all kinds of sensory input, is hardly debatable: in fact a great deal of early schooling consists in making it clear that certain repeated patterns of input lack information and the semantic urge must be directed elsewhere. What is less well known, but is quite extensively supported in the ethological literature (see [Alcorta and Sosis \(2007\)](#) for a modern review), is that animals already exhibit ritualistic behavior that can only be explained by assuming a mechanism that builds internal models that embody causative correlations between sensory inputs and desirable internal states.



Turning to rules, we will not start with sophisticated ethical maxims of the sort Kant discusses, but rather with elementary propositions like 'hot water will cause burning pain'. This is not quite a *rule* in the sense we are interested in, but something even stronger, a *law of nature*: it is strict and exceptionless, and lies entirely outside the sphere of human (individual or social) ability to change. When we say it is exceptionless, we need the implicit qualification 'under normal conditions' to cover cases like congenital analgesia, but this is a qualification we need to add to practically every statement – we return to this issue in [Section 3.6](#). One rule we can derive from the proposition is 'don't get into contact with hot water'. We will not assume that the rule is learned by tradition, though there is no doubt that a very significant part of our knowledge is culturally transmitted. Past sixth grade we all know that the boiling point of water on Mount Kilimanjaro is below 80 °C, yet few of us have gone there and performed the experiment. We will also not assume that the rule is learned by induction, though again there is no doubt that a very significant part of our knowledge is acquired inductively, both at the individual and at the cultural level. In this particular case, however, if the rule is acquired inductively it is likely acquired based on a single instance, rather than by the more laborious process of data mining or rule induction. How the prolepsis supports [one-shot learning](#), especially for (grammatical) rules, is a nontrivial issue, and we refrain from



speculating on the matter until a significant body of rules has been collected, sufficient at least for dealing with the Winograd schemas discussed in Section 7.1.

Since we have eliminated, at least for the sake of this example, all other approaches, we are left with the deductive task of *deriving* this rule from more elementary precepts such as if *X causes Y* and *Y has negative value* then *avoid X*. It is quite clear how to provide at least a rough analysis of causation with automata: if state *X* is deterministically followed by state *Y*, we can say that *Y* was CAUSED by *X*. Similarly, the idea of *avoiding* some state or set of states and the dual idea of *choosing* some state or set of states also lend themselves to a natural formulation in terms of automata exercising free will on the set of nondeterministic states available to them at any given moment. Together with the notion of valuation we have already discussed in Section 3.2, we can see the beginning of how a naive calculus of behavior can be built up, a matter we will address in detail in Section 3.6. We emphasize that such a calculus is not part of the prolepsis, just as the naive theory of soft and rigid bodies lies outside this domain: our interest is in constructively learning such theories built on a narrow proleptic foundation that only contains the idea that objects can have attributes.

The last missing piece of the prolepsis is therefore the deductive ability itself, the ability that can derive both the specific rule *avoid contact with hot water* and the general piece of wisdom ‘don’t do that then’. To get to the specific rule, we will need yet another axiom, that of *locality*, which forbids action at a distance. For our purposes, locality can be stated as follows: for object A to exert influence on object B, A must be in contact with B. To derive the specific rule from the general law, we need some pattern recognition ability to realize that *hot water* can stand in the “A” slot and *skin* can stand in the “B” slot, and, further, some modus ponens-like rule of deduction that licenses the conclusion from the premiss. Let us begin with two binary predicates CAUSE and CONTACT and a unary predicate change. With FOL-like notation, locality could be stated as $A \text{ CAUSE change } B \Rightarrow A \text{ CONTACT } B$. (Here \Rightarrow is some implicational primitive which, together with the variables, is used only for notational convenience – we defer the formal theory to Chapter 4.) Note that it is not the change in B that is in contact with A, acknowledging the possibility of delayed changes, but rather B itself, and note that it is not just the generic substance *water* that needs to be referred to but rather the more specific *hot water*. Some relevant deductive steps can be formulated as follows:



1. CAUSE change \Rightarrow INFLUENCE
2. INFLUENCE \Rightarrow CONTACT
3. hot water CAUSE pain
4. pain bad \Rightarrow hot water CONTACT bad

Readers will no doubt have noticed that in the above proto-formulas we have ignored a lot of things that FOL pays close attention to: for example, when we said $A \text{ CAUSE change } B \text{ implies } A \text{ CONTACT } B$ what we really meant was $A \text{ CAUSE (change } B)$, while in 1 above we really have (CAUSE change) in the premiss, with parenthetization going the other way. Perhaps even more importantly, we have replaced application of the

valuation function, $v(\text{pain}) = \text{bad}$, by simple juxtaposition, pain bad or, perhaps, pain IS bad . We will sort out these details in Chapter 7.

3.6 Regularities



From high school, we are all familiar with the idea of analyzing large and complex machinery such as [gantry cranes](#) in terms of *simple machines* such as levers, pulleys, screws, and wedges, yet it seems a stretch to call these primitive units ‘machines’ and it requires [special talent](#) to consider a geometrical object, the inclined plane, as a simple machine in and of itself. When we study rules, we generally do this with the goal of understanding rule-governed behavior, and have in mind complex examples such as the rules of grammar. Here we look at the simplest kind of rules, perhaps not even deserving the name ‘rule’, because they may lack some feature one may consider essential – to avoid terminological problems, we will call these *regularities* or simply *patterns*. Even among patterns, we avoid the complexity of 2D patterns such as found on leopards or zebras, and concentrate on linear sequences, where progression (conventionally from left to right) can be taken as temporal succession.



Clearly, a simple pattern such as $\text{day-night-day-night-day-night} \dots$ is not just a regularity, but a law of nature in the sense in which we used this term earlier: it lies entirely outside the sphere of human (individual or social) ability to change it, it is exceptionless, and strict. A purist may object to all three terms of this definition. First, it is [easy to imagine](#) a society capable of controlling this matter and, in general, we are quite capable of imagining worlds with different laws of nature. But these kind of ‘possible worlds’ are inferior to the actual world we live in in that we rely (and can continue to do so indefinitely) on the actual laws of this world for our survival, and agents of the kind discussed in Section 3.1 that employ laws of nature in their reasoning will have an evolutionary advantage over those agents that do not (or that rely on the wrong laws).

Second, our purist may object that even if we accept the laws as they are, the $\text{day-night-day-night}$ pattern is not exceptionless, since eclipses break it up time and again. This is a much more serious objection, inasmuch as history leaves little doubt that those who understand the higher pattern of eclipses (or at least have court astronomers that do) have a significant advantage over the barbarians. As the *Analects* (V.14) makes clear, the magnitude of the problem was already clear in antiquity: “Before he could put into practice something he had heard, the only thing Tzu-lu feared was that he should be told something further”. Here we follow the general practice of grammar and admit regularities as laws even if they have exceptions, i.e. if there exist other regularities that may override them.

A particularly clear case is offered by explicitly quantified statements such as *everyone must pay the fee*. An empirical study (Kornai, 2010b) of over 6,000 universally quantified expressions gleaned from 300 issues (45m words) of an American newspaper, the *San Jose Mercury News*, has not revealed a single unambiguously exceptionless case of the kind we discussed in Section 2.5. Since having exceptions is the norm, while

systems of logic are built to sustain only exceptionless generalizations, we need some mechanism to handle the additional workload. The method with the most widespread use, allowing a set of exceptions of [measure zero](#), is problematic in that it only shifts the difficulty to the definition of the measure used. In fact, when we say that *everyone must pay the fee (except senior citizens)* this does not imply that the probability of admitting a non-payer is zero – on the contrary, the exceptions will come from a non-null set. A better approach, and the one we shall follow here, is to take such sentences to express [generic](#) truths, true of an entire genus but not necessarily of each individual, as in *hunters tell tall tales*. This will to some extent blunt the force of our implicational system, since we can no longer conclude *Joe must pay the fee* from *everyone must pay the fee*, but this is a small price to pay, in that the implication actually fails whenever there is an overriding clause (for example, that state employees are exempt).

Finally, our purist may also be critical of the strictness of a pattern. Once allowance is made for eclipses, days and nights may indeed follow each other in regular succession, but the process is not at all exact: sometimes the days are longer, sometimes the nights are, and the symmetry one would expect between days and nights is rarely manifest. We take this objection to rest on a simple misunderstanding of what ‘strict’ means here: it is precisely because the law does not aim at exactness that it can remain strict. Should we attempt to replace it by a complex formula that takes the time of year, longitude, and latitude into account, we would have greater precision, but no stricter explanation of the basic pattern. What makes a person *obese*? Insurance companies might agree that payment for medical treatment may be justified if and only if the weight of a person (expressed in pounds) divided by the square of their height (expressed in inches) exceeds 0.04267, but this is hardly a definition of obesity that makes sense outside a very limited healthcare context, and even there its applicability is dubious, as it is easy to imagine some committee of learned doctors and actuaries moving the threshold to 0.0393. We will return to this matter in Section 3.7, but we note here in advance that we will favor a definitional style where *obese* is defined as ‘very fat, overweight’ in accordance with the everyday meaning.

To build the basic apparatus we will need for handling rules in general, and laws of nature in particular, we need to refresh our knowledge of the basics about relations, semigroups, and operator semigroups. We will provide a series of definitions and basic theorems that the reader should be familiar with, but we omit most of the proofs. Until now, at least for functions of a single argument, we have followed the convention of analysis whereby the function precedes the argument: $f(t)$, $F(\phi)$, and so on. From here on we will follow the convention of algebra, writing the argument first and the function afterwards. This has the great advantage that parentheses can be omitted for successive function applications, since tfg can only mean applying g to the result of applying f to t , or, what is the same, applying the composite function fg to t . Needless to say, the same convention can be used to deal with relation application: if P and Q are some binary relations over some base set U , and $B \subset U$ is some (not necessarily singleton) set, BP will denote all those elements u of U for which $\langle b, u \rangle \in P$ holds for some $b \in B$, and



BPQ will denote all those elements u of U for which there exist some $b \in B$ and $c \in U$ such that $\langle b, c \rangle \in P$ and $\langle c, u \rangle \in Q$ both hold. We begin with the notion of a **binary relation**, which we define as including the base set(s).

Definition 3.2 A *binary relation* R is a subset of the **Cartesian product** $A \times B$ of two sets, and, conversely, every such subset is considered a relation over **domain** A and **codomain** B .

Even though the notation $R : A \rightarrow B$ is often reserved for **functions**, i.e. relations that satisfy $\langle a, x \rangle \in R, \langle a, y \rangle \in R \Rightarrow x = y$, here we will use this notation for relations as well because it makes the domain and codomain explicit. If $A' \subset A$ and $B' \subset B$, we may consider $R' = R \cap (A' \times B')$ to be the *restriction* of R and R to be an *extension* of R' . The Cartesian product is a naturally associative operation (even though the usual set-theoretical definition of the ordered pair $\langle a, b \rangle$ as $\{\{a\}, \{a, b\}\}$ does not make this clear), and we will make no distinction between $\langle a, \langle b, c \rangle \rangle$ and $\langle \langle a, b \rangle, c \rangle$ and will treat them both as $\langle a, b, c \rangle$.

Definition 3.3 Given some relations $R \subset A \times B$ and $S \subset B \times C$ we define their *product* $T = RS \subset A \times C$ as containing all pairs $\langle a, c \rangle$ and only those pairs for which there exists some b such that $\langle a, b \rangle \in R$ and $\langle b, c \rangle \in S$.

Since relation composition is associative, binary relations over some set X define a **semigroup**, which we will call the (full) *relational monoid* over X and denote by $\mathbf{FR}(X)$. (Recall that a semigroup is a **monoid** if it has an **identity element**.)

Exercise^o 3.10 Prove that a semigroup S can always be embedded in a monoid, i.e. there exists a homomorphism $\phi : S \rightarrow M$ into a monoid M such that $S\phi \subset M$ is isomorphic to S .

Any set of relations over X closed under composition will be a semigroup (or monoid, if the identity is included) – these are called *relational semigroups* or (*monoids*). In fact, these are the *only* semigroups (or monoids) one needs to consider, in that any arbitrary semigroup (or monoid) will be isomorphic to a relational semigroup (or monoid respectively). This is the semigroup version of Cayley's Theorem.

Cayley's Theorem Every semigroup S is isomorphic to a semigroup of relations T . If S has an identity, the corresponding relation in T is the identity (=) relation.

Proof Either S has an identity element, or we can extend it to a monoid S' by adjoining one. For each element s of S' , we define a relation T_s over S' by those pairs $\langle a, b \rangle$ for which $as = b$ holds in S' . Since semigroup multiplication is an operation yielding a unique result, every time we have $\langle a, x \rangle \in T_s$ and $\langle a, y \rangle \in T_s$, it follows that $x = y$, i.e. T_s is not just a relation, it is a *function* from S' to S' , and the image of a under T_s , denoted aT_s , is simply as . Further, $aT_xT_y = aT_{xy}$, so the mapping ϕ that assigns T_s to s is a homomorphism. Since S' has an identity e , and $eT_s = es = s$, it follows that for $x \neq y$ we have $T_x \neq T_y$, i.e. ϕ is **injective**, and therefore invertible on its range. ■

When the relations in S are functions, the semigroup is called a **transformation semigroup**. The monoid of all transformations over some set will be called the *full transformation monoid* and will be denoted $\mathbf{FT}(X)$. As the proof above makes clear, every

semigroup is isomorphic to a subsemigroup of some $\mathbf{FT}(X)$. In particular, $\mathbf{FR}(X)$ is also representable as a transformation semigroup, but there is a price to pay in that the size of the representational base grows superexponentially.

Exercise^o 3.11 If X has n elements, how many elements does $\mathbf{FR}(X)$ have? Embed $\mathbf{FR}(X)$ in some $\mathbf{FT}(Y)$ using the method of the above proof. How many elements does Y have?

When the relations in S are invertible functions, the semigroup is called a *permutation semigroup*. The monoid of all permutations over some set will be called the *symmetrical group* and will be denoted S_n . Permutation semigroups are always embeddable in groups, so we should speak simply of *permutation groups*. However, it is not the case that every semigroup can be embedded in a group, as the following simple example shows.

Example 3.2 *A transformation semigroup over two elements.* Let $0,1$ be two elements, with $I = \{\langle 0,0 \rangle, \langle 1,1 \rangle\}$ the identity transformation, $P = \{\langle 0,0 \rangle\}$ and $Q = \{\langle 1,1 \rangle\}$ two distinct nonidentity elements, and Z the empty relation. As a moment's thought will show, $PP = P, QQ = Q, PQ = QP = Z$, and the zero Z and identity I behave as expected: $PZ = ZP = QZ = ZQ = IZ = ZI = ZZ = Z, IP = PI = P, IQ = QI = Q, II = I, IZ = ZI = Z$. Thus these four elements are closed under multiplication, and form a (commutative) monoid M . Suppose we wish to embed M in some larger group G where Z would have an inverse W : this would mean that in the group, $P(ZW) = PI = P, Q(ZW) = QI = Q$ and, by associativity, $P(ZW) = (PZ)W = ZW$ and $Q(ZW) = (QZ)W = ZW$. Thus, by transitivity we have $P = Q$, a contradiction.

One critical issue highlighted by this example is that unlike groups, semigroups may lack the so-called **cancellation property**: in a semigroup, if $AX = BX$, it does not follow that $A = B$, while in a group this conclusion always holds. A simple transformation lacking the cancellation property is one that maps everything onto a single point: this is called the *reset*, and it is a remarkable fact about semigroups that they **can be built up from permutations and resets**. Groups arise naturally in the context of studying the **automorphisms** of a structure, while semigroups arise from studying its **endomorphisms**.

By generalizing Example 3.1, we obtain a particularly important case of transformation semigroups attached to algebraic systems. We describe a *state machine* (also called a *semiautomaton*) as a finite collection of states Q influenced by a finite set of inputs Σ (see Definition 3.1). By ‘influence’ we mean simply that each $\sigma \in \Sigma$ is a partial function (transformation) of Q : we write the result of applying the transformation σ to q as $q\sigma$. Since it is often more convenient to work with total functions, we can add a special *sink state* s to Q and define $p\sigma$ as $s \in Q'$ whenever $p\sigma$ was undefined in Q , and of course extend all $\sigma \in \Sigma$ by $s\sigma = s$. The state machine is thus fully defined by a finite data structure, denoted T and called the *transition table* (and also called the *transition matrix*), whose element at the intersection of the p -th row and σ -th column is given by $p\sigma$. Clearly, each state machine $\langle Q, \Sigma, T \rangle$ has a semigroup S associated to it, and, conversely, each finite transformation semigroup S can be associated to a state machine



whose states are given by the base that S acts on and whose alphabet Σ is made up from the elements of S .

After these preparations, we are ready to offer a simple formal model of regularities: we will say that a regularity is whenever some input changes (or leaves unchanged) one or more states. (In Chapter 4, outputs will also be considered.) In the day–night example, there are only two states, Day and Night, and only one input, *next*, and the laws of nature that we are after can be simply stated as $Dn = N$ and $Nn = D$.

Exercise^o 3.12 Refine the day–night cycle so as to include transitional periods dawn and dusk. Do you need extra operators beyond ‘next’?

Exercise^o 3.13 State the birth–life–death succession by means of an automaton. How is the life–death part different from saying *all men are mortal*? How would you phrase the birth–life part by means of universal quantification?

Exercise[†] 3.13 (continued) Study the major systems of [eschatology](#) and state them by state machines.

It is not at all obvious that all regularities can be expressed as transitions in appropriately chosen state machines, just as it was not at all obvious at the outset that all of classical physics could be expressed by partial differential equations or that all of classical mathematics could be expressed in the language of set theory. In fact, modern science provides a rich storehouse of laws of nature that would be hard, if not impossible, to express in this primitive language; take, for example, *the atomic weight of carbon is 12.0107 daltons*. We emphasize that the impact of such statements on natural language semantics is minimal – innumerate and scientifically uneducated people are perfectly capable of using natural language to convey their thoughts and feelings, and it is the everyday use of ordinary language that we wish to understand.

A far more pertinent set of examples is furnished by the Sung philosophers’ notion of *pattern* or *principle*: a thing must have a rule to which it should conform. All things have principles; for example, fire is hot, and a tree flowers in the spring and fades in the autumn. That the ruler is superior to the minister is a constant principle of the Empire (these examples are from Graham (1958)). Another philosophical precursor is Leibniz, whose [monads](#) have a “blueprint”, i.e. a complete concept or law of the series that lists all of its states (Bobro, 2013). We begin to address the issue of how to formulate these and similar patterns in Chapter 4.

3.7 The standard theory

As we have seen in Chapter 2, a simple logical theory expressed in first order predicate calculus already contains three highly structured parts: a language of formulas L , a collection of models \mathcal{M} , and an interpretation relation $\text{int}: L \rightarrow \mathcal{M}$ between the two. (In addition to these, there is also a less visible component, the proof theory, which describes what syntactic manipulations of the formulas preserve the truth defined by the interpretation function, but we can safely disregard this for the moment.) One might expect linguistic theory to follow the same architecture, using L to contain all well-formed



(grammatical) strings and only these strings of some natural language such as English, \mathcal{M} , the collection of models, to capture the world that is being talked about, and int to map elements of the language onto their meanings.

Historically, it was this simple ‘naive’ picture of natural language semantics that drove the abstraction process leading Tarski (1956) (Polish original 1933, German translation 1935) to model-theoretic semantics, but the standard theory of linguistic semantics, originally proposed by Montague (1970) and Montague (1973), represents a considerable departure from this architecture. On the left side, we do not find not L , natural language, but D , *disambiguated language*, a theoretical construct that contains not just the well-formed expressions of language but also their constituents and derivation histories. On the right side, we do not find real-world objects or even formal objects (models), but formulas F of a particular logic calculus that we will discuss shortly. The full picture of the standard theory, called *Montague grammar*, is composed of the first two or three arrows in Fig. 3.3, with the primary attention focused on the translation homomorphism t , with the models \mathcal{M} being reasonably standard set-theoretical constructs (except for an internal time parameter that temporal semantics often relies on), and the grounding g in the real world completely left out.

$$L \xrightarrow{d} D \xrightarrow{t} F \xrightarrow{I} \mathcal{M} \xrightarrow{g} W$$

Fig. 3.3. Information objects associated with MG

The disambiguation mapping d is an elegant technical device that helps a great deal in simplifying subsequent stages of the mapping. Unfortunately, scholars in the MG tradition have spent little effort on building grammatical models of natural language that could serve as a starting point for disambiguation in the sense Montague urged, and the use of d in semantics is more a promissory note than an actual algorithmic method. The problem is not so much that the pioneering examples from *Every man loves a woman such that she loves him* to *John seeks a unicorn and Mary seeks it* could hardly be regarded as examples of ordinary language, as the alarming lack of progress in this regard – best-of-breed implementations still cover only a few dozen constructions. Another part of the theory that has remained, for the past forty years, largely unspecified, is the mapping g that would *ground* elements of the mathematical model structure in reality. For a mathematical theory, such as the theory of groups, there is no need for g as such, in that there are no groups “in the world”. All objects in mathematics that have group structure (for example, the symmetries of some geometrical figure) can be built directly from sets (since a symmetry is a function, and functions are sets), so restricting attention to model structures that are sets is entirely sufficient.

The problem arises with non-mathematical concepts such as, say, colors or dresses. Unfortunately, there are no red or green sets, and the idea of speaking about the set (or some set) of dresses is fraught with difficulties. Since a version of [naive set theory](#) is often used to speak of the ‘set of red things’, we will spend some time here analyzing this

Phil





notion. At the beginning of this chapter we assumed, together with Scanlon (1988) and the whole *realist* tradition of philosophy, the objective existence of everyday things, and the subjective existence of sensations such as red color. As long as there are things, and as long as there are perceptual qualia, speaking of the set of red things seems to come for free. Unfortunately, from the fact that there are red things it does not follow that the set of red things can be formed. In axiomatic set theory this step is justified by the *comprehension* axiom scheme, but the axiom pertains to sets of the theoretical kind, not actual sets of things in the real world. To say that sets of real things will obviously satisfy the axioms of set theory is like saying that real triangles (for example those whose vertices are faraway points like stars and whose edges are traced by rays of light) will have their angles sum to 180° – there is no guarantee built into the structure of the real world, the structure of set theory, or the structure of the grounding map g that would guarantee this. As a matter of fact, light rays are a reasonable approximation of (geodesic) lines, stars are reasonably point-like on an interstellar scale, yet the angles do not add up to 180° – the real world happens to be non-Euclidean, and for all we know, naive sets in reality may not satisfy the comprehension axiom scheme.



Be that as it may, the intuitive picture behind MG is often presented in terms of naive set theory: suppose *Jones* is a name, and we denote the universe of individuals by P (by individual we mean not just individual persons, but also individual dresses, individual blog postings, etc. etc.); then the *extension* of ‘Jones’ satisfies $(\text{Jones})t \in P$. For one-place predicates such as *dress* or *red*, t will yield sets of individuals, i.e. a member of 2^P , the set of dresses and the set of red things, respectively, and the same goes for compound predicates such as *red dress* or *wears a red dress*. Some predicates are *intersective* in the sense that $(\text{red dress})t$ is expected to be $(\text{red})t \cap (\text{dress})t$, and if x wears y and $y = a$ red dress it will follow that x wears a red dress. However, other predicates do not follow this simple intersective pattern: if Jones is a student at Springfield High, it follows that her classmates are also Springfield High students. However, if she goes to college, and becomes a former Springfield High student, it is no longer the case that her classmates are also former Springfield High students. A similar puzzle surrounds the statements *the temperature is thirty* and *the temperature is rising*: both may be true at the same place and the same time, yet it does not follow that thirty is rising.



To handle these and similar problem cases of *opacity* (see Section 3.2), MG shifts attention from the value that t would take in a single model (a matter we shall return to in Section 7.2) to the value it takes in all models. Instead of simply collecting model structures in \mathcal{M} we add a base class I , elements of which are used to *index* the models. This could be a technically ambitious undertaking even at the foundational level, since, as we discussed in Section 2.5, theories that have infinite models have models at every cardinality, so I would be a proper class (bigger than any set). As a workaround, intensional theories come with model structures called frames which use a fixed I and an I -indexed set of models. Instead of looking at the value $(x)t$, called the *extension* of x in some model, we look at the total collection of values given by the function $T : I \rightarrow M_i$, called the *intension*. Take some property such as *red* whose extension in a model is sim-

ply the set of red things there. The problem with identifying the *meaning* of ‘red’ with this set (red)*t* is that if I decide to paint my red barn white, the extension changes, while it is hard to believe that the meaning of ‘red’ has also changed. Using intensions offers a way out of this quandary: the meaning of ‘red’ is the intension of red, an indexed family $R_i \subset M_i$, which remains unchanged by my painting the barn white. What changes, under this conception, is the model (also called a *possible world*) or, equivalently, the index we are at, but the overall family is unchanged.

A simple application of this idea would be to use a single time parameter τ to index models: we think of possible worlds as the actual world at different time instances. (This is not how Montague actually handled time, but MG is a rich family of related theories with many alternative analyses, and the idea of using times as possible world indexes is standard.) With this conception, the intension of *the temperature in Paris* is a function from I (the set of times) to the reals, and the extension is a single numerical value at any given time. This solves the puzzle of ‘thirty is rising’ quite nicely, since in one sentence we talk of *the temperature* as a function, and in the other as a value, so transitivity cannot be invoked.

Another technical device that plays an important role in MG is a relation $A \subset \mathcal{M} \times \mathcal{M}$ called the *accessibility* relation. This is used to define two important modalities, *possibly* and *necessarily*, denoting them by \diamond and \square , respectively (and writing them, following the tradition of logic, to the left of the predicate they apply to). Returning to our examples from Section 2.4, when we say *ice is cold* what we mean is translated into MG by assigning the extension B to *ice* (things that are ice) and the extension C to *cold* (things that are cold): the sentence will be true if $B \subset C$. When we say *cancer has no cure* we again have two sets, K for illnesses that are cancer, and H for illnesses that have a cure – the sentence is true if $K \cap H = \emptyset$. The difference is that $B \subset C$ is true at every index. We write this as \square *ice is cold* and paraphrase it as *ice is necessarily cold*, and we write \diamond *cancer has no cure* and paraphrase it as *cancer possibly has no cure*. More formally, we extend the semantic definition of \models introduced in Section 2.5 by the following clause: $W \models \square p$ iff for all V satisfying WAV we have $V \models p$. In other words, we consider a proposition p necessary in a given model W if it is true in every model V accessible from W . Once necessity is taken care of, it is easy to define $\diamond p$ as $\neg \square \neg p$ and, conversely, if we had defined possibility first, necessity would follow as ‘not possibly not’ – the two notions are dual. The primary advantage of formulating possibility and necessity through the alternative relation (Kripke, 1959) is that this clarifies the status of several plausible rules concerning the modal operators. For example, to demand a rule of deduction $\square p \rightarrow p$ (if something is necessarily true, it is true) is to say that A is reflexive, and to demand $\square p \rightarrow \square \square p$ is to say that S is transitive.

The complexity that modality and intensionality (not to speak of *higher order*, a feature of MG we will not discuss here) add to the logical calculus has to be weighed against the actual power of these techniques to resolve the issues that led to their introduction. With modality, the main problem is that we are forced to enlarge the apparatus at every level to accommodate a notion, necessity, that plays at best a tangential role in



natural language. Just as there is a significant gap between the actually attested generic use of quantifiers and the ‘episodic’ readings that are formalized in MG, there is a similar gap between the use of *necessary* in everyday statements such as *water and food are necessary for survival* and technical statements such as *water necessarily boils at a hundred degrees centigrade*. The problem is not so much that both statements require further qualifications (really, water and food are not necessary for the continued survival of iambic pentameter, and it is only under normal atmospheric pressure that water will boil at a hundred degrees) as the difference in argument structure: real language expressions have the form *x is necessary for y* while the formal necessity operator deals with *it is necessarily the case that x*.

Exercise \rightarrow **3.14** We call a modal calculus *Euclidean* if possible things are necessarily possible: $\Diamond p \rightarrow \Box \Diamond p$. What condition(s) will the accessibility relation have to satisfy to guarantee that proofs that rely on the Euclidean property are actually sound? Give an example of an accessibility relation that gives rise to a nontrivial model system, but fails to be Euclidean.

We have a reasonably good idea about what is necessary for some matter (action or state of affairs) to come about; in fact much of our everyday encyclopedic knowledge can be recast in terms involving the binary NECESSARY_FOR relation: we say a dry and rust-free surface is necessary for paint to adhere properly, regular exercise is necessary to avoid obesity, and so forth. It is easy to test such statements empirically: one only needs to apply paint to a wet or rusty surface, neglect to exercise, etc. and watch the results. But with the unary \Box operator our ideas of what constitutes a valid test are much weaker. For example, we know that in bridge a bid of four spades is necessarily higher than a bid of four diamonds, for otherwise the game would not be what we call bridge. But as soon as we leave the definitional realm we are no longer in a position to say that something is necessarily so. For example, we know that in water molecules, the angle between the two hydrogen atoms is a constant, 104.5° , a number that does not change as we heat the water, put it under pressure, or add salt or other chemicals, and we are tempted to say that the angle in question is *necessarily* obtuse. Yet our knowledge of [polymorphism](#) is not quite robust enough to exclude a polymorph of H_2O where the angle between the two hydrogen atoms is acute, say 88° . Call such a hypothetical substance acute angle water or aater. Upon observing aater in an experiment or in nature we must either conclude that $\Box_{\text{water}} \text{O-angle obtuse}$ was false, or retreat to the definitional realm and say that aater is not really water – what was defined as H_2O now needs to be redefined as ‘ H_2O with obtuse angle at the O vertex’. This rhetorical move, known as the [No true Scotsman](#) fallacy, is effective (to the limited extent that it is) precisely because in the definitional realm, things defined to be so are indeed necessarily so.

Exercise \rightarrow **3.15** We call a modal system *normal* if provable things are necessarily true: if A is a theorem, so is $\Box A$. Is distributivity of \Box , i.e. $\Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$, a theorem in a normal system? Is it necessarily a theorem?

The techniques applied for \Box and \Diamond remain largely applicable to other, seemingly quite unrelated cases such as **epistemic** and **deontic** modality, but with the same fundamental problems: we have very few truth-conditional guidelines when such statements are valid. When we say *Peter's work is really excellent, he deserves a raise* the issue is not so much that we have trouble defining 'really excellent', for we have similar trouble defining almost every adjective (we will return to this issue in Chapter 5), but rather the 'deserving' part: he may never get a raise even if deserving and he may get a raise even if undeserving. We would gain no traction over the matter by simply inspecting what happens in (reachable) possible worlds even if we knew how to carry out such an inspection. The same difficulty pervades modal theories of knowledge: we may not know something even though we have compelling evidence for it, and we may know, or think we know, something that is not supported by evidence.

Another, more subtle difficulty arises in combining modalities: having different kinds of operators, each with their dedicated accessibility relation, leads to a combinatorially intractable system of model structures. For this reason we will pursue a lexical approach here, where the relevant properties are directly encoded in the keywords *necessary*, *know*, *must* etc. because this is considerably simpler to set up and maintain. How this is done will be discussed in greater detail in Section 7.3.

3.8 Desiderata

In general, representational theories of meaning, not just MG or the rival theories discussed in this book, but all theories that assign some representation xR to each linguistic expression x , are expected to meet some requirements. First, we expect the map $x \mapsto xR$ to be a **computable function** of x . (Since people are actually capable of assigning meanings to such expressions in real time, it is not an unreasonable demand on the theory for R to be computationally simple, say linear or at most **polynomial**, but we leave this matter to one side.) Trivial as this requirement may be, it already excludes the attractive **direct** theory of reference that would say that the meaning of a name is the person or thing named by it, for the following reasons. For one thing, we don't actually have a naive theory of sets that would let us express the grounding function g of Fig. 3.3 – we only have axiomatic set theory, and once g is a mapping from models (appropriately structured sets) to the world *conceived as a set* we have accomplished nothing over and above the theory that excludes this last step. For another, even assuming we can somehow sneak in the real world (or better yet, all kinds of possible worlds) among the models, direct reference predicts that the word *Shakespeare* refers to the unique historical person **William Shakespeare**. But if this is so, who is *the Polish Shakespeare* that the “*Looking for the Polish Shakespeare*” *Contest for Young Playwrights* wants to find? Clearly, not some British subject born in Stratford-upon-Avon but a brilliant playwright who is a Polish national. Unfortunately, direct reference means there is no Polish Shakespeare, there never was one, and there never will be one, not even in a parallel universe, so the expression is predicted to be nonsense, which is contrary to actual usage. This brings



us to a cardinal methodological point that we have already urged in the introduction. Philosophers and logicians may take a hard line and argue that, well, since Shakespeare was actually non-Polish, the expression *Polish Shakespeare* is indeed nonsensical. But to the extent that we are interested in building a workable semantics for natural language expressions, we simply cannot ignore the fact that the organizers named their contest the way they did, fully expecting people to understand the construction *Polish Shakespeare* to evoke the idea of ‘brilliant Polish playwright’, and we cannot ignore the fact that these expectations are fully met; it is exactly this meaning that people attribute to the phrase.

Second, we may require our semantic theory to account for synonymy: if two expressions x and y mean the same thing, we may demand $xR = yR$ and, conversely, whenever $xR = yR$, we may want to conclude that x and y are synonymous. The theory that we will explore in Chapter 4 and beyond actually fails this test: it is commonly agreed that *Rottweilers* are different from *St. Bernards* yet in our theory both are represented as dog . Rather than criticizing common usage, we simply say that it includes a great deal of encyclopedic knowledge, a matter we shall explore in Chapters 4 and 5. MG also fails this simple test, because there are many expressions such as *square circle* and *triangular circle* which are logically inconsistent. Since these have no models, they are missing from every possible world, meaning that their intension R is the function that assigns the empty set to every index. Since they have the same intension, we are forced to conclude that they mean the same thing, which is empirically false. One may argue that such a failure is not truly damning, since it is only bizarre examples of nonexistent objects that cause problems. In fact, once these examples are at hand, it is trivial to leverage them to existent examples: if object A under the blanket is a hat or a triangular circle, and object B is a hat or a square circle, the intensions are the same: $(\text{hat or square circle})T \equiv (\text{hat or triangular circle})T$, a function that is not identically empty as long as hats exist. More important, there are many objects (such as an integer $n \geq 3$ such that $x^n + y^n = z^n$ has solutions in positive integers x, y, z) whose existence is not so easy to determine. In fact, there are many seemingly reasonable things whose existence is still not known, and the fact that they turn out to be necessarily nonexistent does not mean they are all the same. Altogether, the algebraic theory and MG fail this test for the opposite reasons: we refuse to burden the algebraic theory with certain kinds of knowledge, while MG insists on incorporating all mathematical knowledge (see Section 4.1 for further discussion).

Third, we may require our semantic theory to account for implications. There are many ways to state this requirement formally, the most ambitious one calling for a sound and complete proof theory of the kind discussed in Section 2.6. This requirement could be relaxed in various ways, and for certain theories (broadly, those capable of expressing a weak form of arithmetic known as *Robinson’s Q*) in fact it must be relaxed – this is the celebrated *Gödel incompleteness theorem*. However, it is not at all clear that a semantic theory that focuses on natural language must be capable of handling arithmetic (as Chomsky (1965) observed, syntax requires no counting), so completeness may still be an attainable goal. Soundness is more problematic, in that commonsense reasoning



about objects, people, and natural phenomena often invokes inference rules that are not sound. Consider, for example, the following rule: *if A' is part of A and B' is the same part of B and A is bigger than B, then A' is bigger than B'*. Let us call this the Rule of Proportional Size, RPS. A specific instance would be that children's feet are smaller than adults' feet since children are smaller than adults. RPS is statistically true, but not entirely sound: we can well imagine, for example, a bigger building with smaller rooms. Nevertheless, we feel comfortable with these rules, because they work most of the time, and when they don't, a specific failure mode can always be found: we will claim that the small building with the larger rooms, or the large building with the smaller rooms, is somehow not fully proportional, or that there are more rooms in the big building, etc. Also, such rules *are* statistically true, and they often come from inverting or otherwise generalizing rules which are sound, for example, the rule that if we build A from bigger parts A' then the parts B' that B is built from, A will be bigger than B. (This follows from our general notion of *size*, which includes additivity.) Once we do away with the soundness requirement for inference rules, we are no longer restricted to the handful of rules which are actually sound. We permit our rule base to evolve: for example, the very first version of RPS may just say that big things have big parts (so that children's legs also come out smaller than adults' arms, something that will trigger a lot of counterexamples and thus efforts at rule revision); the restriction on it being the same part may only come later. Importantly, the old rule doesn't go away just because we have a better new rule. What happens is that the new rule gets priority in the domain it was devised for, but the old rule is still considered applicable elsewhere.

Fourth, we may require the theory to provide a means of linking up meanings across languages, serving as a translation pivot. The direct use of Montague's IL for this purpose was explored in the 1970s (Hauenschild, Huckert, and Maier, 1979; Landsbergen, 1982), but these attempts faltered for a variety of reasons, chief among them the inability to extend Montague's original grammar fragment to a wider coverage. Because MG concentrates on the compositional aspects of meaning at the expense of word meaning, the kind of logical form it uses abstracts away from over 85% of the information content of sentences, which makes it less than ideal for the translation pivot role. In contrast, the algebraic theory of lexemes is eminently suitable for translation, in no small part because we make translational equivalence criterial in the definition of meaning. For example, *chrome*₁ 'hard and shiny metal' is translated in Hungarian as *króm*, while *chrome*₂ 'eye-catching but ultimately useless ornamentation, especially for cars and software' is translated as *ciráda*. How this method can be carried through in large multilingual lexica will be discussed in Chapter 6.

Fifth, we require the theory of lexical semantics to connect to a theory of the meaning of larger (non-lexicalized) constructions including, but not necessarily limited to, sentential syntax and semantics. MG uses word-specific axioms, known as *meaning postulates*, to describe word meaning, an approach made very powerful by the fact that there is nothing in the theory that limits the expressive power of the axioms. For example, standard unary predicates classify objects into two categories: *blueX* will hold iff *X*

appears to the naked eye as having the color blue. We can trivially extend this to compound Booleans such as *blue or green*. For a fixed temporal parameter t , we may call something *examined* if somebody examined it before t . Thus we can define the predicate *grue* to mean ‘(green and examined) or (blue and not examined)’, and similarly *bleen* to mean (blue and examined) or (green and not examined)’. We may feel that *blue*, *green*, and *examined* are primitive in some way that *grue* and *bleen* are not, yet this is nowhere captured in the system, as can be seen from the following.

Exercise[→] 3.16 Given *grue*, *bleen*, *examined*, define *blue* and *green*. Are these definitions simpler, more complicated, or just as complex as the converse definitions given above?

As we shall see in Chapters 4 and 5, the algebraic theory of lexical semantics meets this criterion maximally, as it uses the same objects, machines, for representing meaning from the smallest morpheme to the largest construction (but not beyond, as [communicative dynamics](#) is left untreated).

Finally, we list responsiveness to philosophical puzzles as a criterion of adequacy, though it must be said that accounting for the vast range of empirical facts observable in everyday language use seems to us a great deal more important than accounting for such puzzles. Be that as it may, since MG starts out with the goal of accounting for opacity, we may as well ask to what extent the enterprise succeeds in doing this. The results are surprisingly mixed: key cases like *The reporter is looking for the oldest person in Asia in 2011* v. *The reporter is looking for Chiyono Hasegawa* are still largely unresolved. In some weak sense we may consider possible worlds where the two notions of *the oldest person in Asia in 2011* and *Chiyono Hasegawa* are not coextensive, but this is purely speculative as we don’t have the means to exhibit a counterexample. In [deterministic](#) models (which are unrealistic from a psychological standpoint, as we discussed in Section 3.4), it would be impossible to furnish counterexamples. But even if the world is not truly deterministic, there are cases where such counterexamples cannot be exhibited at all, not even in principle, as we shall see in Chapter 5.

Exercise[→] 3.17 Assume time is discrete, and that the only possible worlds are the ones actually obtaining at some time t . Let the accessibility relation $R_{i,j}$ mean that evidence from the previous i time instances is available and we will have the chance to conduct experiments in j future instances ($i, j \geq 0$). Define $\Box p$ to be true at t iff it is true at all instances $t - i, t - i + 1, \dots, t + j - 1, t + j$ (with i, j fixed), and $\Diamond p$ to be true iff it is true at at least one of them. Which of the following deductive rules are sound?

- (D) $\Box p \rightarrow \Diamond p$
- (M) $\Box p \rightarrow p$
- (4) $\Box p \rightarrow \Box \Box p$
- (A) $\Box p \rightarrow \Box \Diamond p$
- (5) $\Diamond p \rightarrow \Box \Diamond p$
- (CD) $\Diamond p \rightarrow \Box p$

($\Box M$) $\Box (\Box p \rightarrow p)$

(C4) $\Box \Box p \rightarrow \Box p$

(C) $\Diamond \Box p \rightarrow \Box \Diamond p$

Exercise[→] 3.18 Assume time is discrete and cyclic: there is a constant T such that the world at time t is identical to the world at $t + T$ for every t . How do the results of Exercise 3.17 change?

Exercise[→] 3.19 Assume time is discrete but a limited number of choices are possible: at every t there are exactly P worlds w^0, w^1, \dots, w^{P-1} , and the accessibility relation $R_{1,1}^k$ means that for every time t and world w^a evidence from the previous instance is available from worlds w^b for $|a - b| \leq k$, and experiments will be possible in the same worlds at $t + 1$. How do the results of Exercise 3.17 change? Do not assume that worlds w_t^b are accessible from w_t^a except when $a = b$.

Exercise[†] 3.20 Explore the analogous systems where time is continuous.

3.9 Continuous vector space models

It is evident from the foregoing, in particular from Fig. 3.3, that the technical apparatus of the classical model is painfully complex, with center stage taken by the set of formulas F that belong to a higher order intensional calculus IL (Gallin, 1975) that is known to be Turing-complete: any problem from any domain whatsoever can be reformulated as a problem of satisfaction in IL. That the machinery is not at all specific to linguistic semantics is not a fatal flaw (compare [partial differential equations](#), which find many uses in physics, chemistry, and biology without being specific to either of these fields), and many insights can be gained by grappling with IL (Hobbs and Rosenschein, 1978; Lapierre, 1994), but it certainly gives license to explore other broad families of formalisms besides those of logic. Here we will use vectors instead of formulas, and in the next chapter we will use graph- and automata-theoretic notions to capture semantics.



Linguistics has a long tradition of expressing meaning by means of vectors. The standard model of lexical decomposition (Katz and Fodor, 1963) divides lexical meaning into a systematic component, presumed to be shared across languages, which captures aspects of the meaning in terms of discrete (usually binary) features such as *male/female*, *human/animal*, etc., and an accidental component called the *distinguisher*. Different but related meanings are collected together in a single tree – Fig. 3.4 gives an example.

This representation has several advantages: for example, *bachelor*₃ ‘holder of a BA or BSc degree’ neatly escapes being *male* by definition. Certainly the idea of structure sharing across different senses of a word has a lot to recommend it, but it is not obvious what, if anything, we wish to share across *chrome*₁ ‘hard and shiny metal’ and *chrome*₂ ‘eye-catching but ultimately useless ornamentation, especially for cars and software’. To the contemporary speaker the etymological relationship is still transparent, as it was only half a century ago that *chrome*₁-plated bumpers, hubcaps, window frames, and door

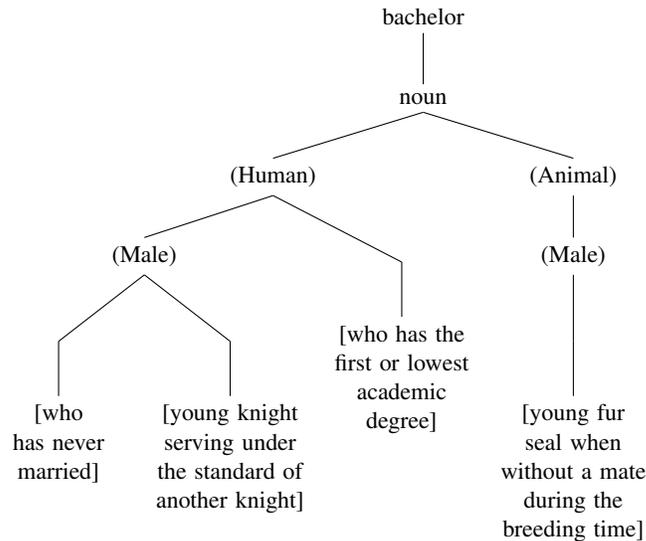


Fig. 3.4. Decomposition of lexical items into features

handles were a primary means of adding *chrome*₂ to cars, but clearly the etymology is non-causal, especially as many other ornamental additions such as tailfins were also in broad use in the automotive industry. We defer the issue of structure sharing until after a full discussion of the continuous case, but note here that the issue of related meanings sharing the same form, handled by the disambiguation mapping d before we come to the formulas in Fig. 3.3, remains just as important if we use vectors instead of formulas.

While the idea of representing meanings by extralogical means such as vectors from some finite-dimensional vector space over \mathbb{R} takes some getting used to, the practice is by now widespread (see our discussion in Section 2.7), and the proof of the pudding is in the eating. Here our primary issue is whether the desiderata laid out in the previous section can be met by systems that rely on continuous vector space (CVS) representations. First, we expect xR , the vector assigned to expression x , to be a computable function of x . This will be accomplished by storing aR for each atomic expression a : as there are only finitely many atomic expressions (morphemes), storing the entire function R is feasible. R is called an *embedding* in this context, since it acts to place each morpheme or word (and, as we shall see, the meanings of complex expressions as well) within the Euclidean geometry of \mathbb{R}^n . With 10^5 words and $n = 100$ (these values are typical), it takes 40 MB uncompressed to store a word embedding with one 32-bit floating-point number per dimension. For full computability, we also need to deal with complex expressions: a typical method is to simply add up the vectors assigned to the component words. This corresponds to a *bag of words* approach, which is already known to be quite effective in tasks like *information retrieval*. We will see more sophisticated methods later on, but all remain computable in polynomial time. Computing the embedding itself is typically done on the basis of distributional similarity, for example by the method described in

Comp



Section 2.7: we encode in a very high-dimensional vector what words a given word x cooccurs with within a window centered on x , and apply standard [dimension reduction](#) techniques to bring n down to a manageable size. While the reduction process itself may not have polynomial guarantees, it is performed offline, before the semantics is used in natural language processing (Collobert et al., 2011).



Our second desideratum, accounting for synonymy, is met in principle: if two words or larger expressions are synonymous, ideally they are embedded in \mathbb{R}^n at the same point. In fact, vectors can do even better: if u is *more* synonymous to v than to w we can expect the distance between uR and vR to be *less* than that between uR and wR . The relevant distance may not be Euclidean: a typical choice is the angular (cosine) distance. The direction of the vectors is of far greater importance than their length, so many embeddings are best conceptualized as placing the expressions on the unit sphere. In the logical framework, there is no way to talk about meanings being more or less synonymous: either a formula is equivalent to another or it is not, a strictly 0–1 decision. Yet there is a clear pre-theoretical sense of *hare* being more synonymous to *rabbit* than to *ox*, or *hurt* being more synonymous to *maim* than to *praise*, – with vector representations we can capture this idea in a manner amenable to empirical testing.

This entails a significant shift in perspective. Logically, *thirteen plus seventeen* is strictly synonymous with *thirty*, while the vectors for these expressions may not line up too well. This shift, as we shall see, is caused primarily by different empirical facts in the two domains. In programming languages, wherever a single arithmetic expression like ‘30’ is legitimate, a complex expression like ‘13+17’ is just as good. In natural language, the two are not so neatly interchangeable, not just in fixed expressions like *Thirty days hath September*, which would come out rather odd as *?Thirteen plus seventeen days hath September*, but also in ordinary conversation. Consider *How many people are you expecting for the party? ?Oh, about thirteen plus seventeen. Or ?You know, thirteen plus seventeen is the last birthday you’ll really enjoy, after that it’s all downhill.* (Here and elsewhere we follow the notational convention in linguistics of prefixing an asterisk to ungrammatical, and a question mark to questionable/strange utterances.) Because in English *thirteen plus seventeen* has a very different distribution from *thirty*, the vectors assigned to these expressions will be quite different, since R is generally computed based on distributional similarity as discussed above.

Currently our third desideratum, accounting for implications, is not well met by CVS models. Certain simple implications, such as *John used pliers* \Rightarrow *John used a tool*, look quite amenable to vectorial techniques in that we expect the vectors for ‘tool’ and ‘pliers’ to line up quite well. But the converse implication *John didn’t use a tool (to fix the faucet)* \Rightarrow *John didn’t use pliers (to fix the faucet)*, while equally valid, poses a far greater challenge, in that negation and Booleans are not trivial to capture in this framework – a good empirical test of various proposals is available in the form of the [Recognizing Textual Entailment](#) (RTE) shared tasks. In fact, this requires the same shift in perspective as discussed above, since the natural language Booleans are not the same as the logical Booleans we studied in Chapter 2 – we will return to this matter in Section 7.3.



Compare *Male or female, I will hire the first competent person* to *Tall or short, I will hire the first competent person*. While logically equivalent, the first statement declares that the speaker is opposed to gender-based discrimination, while the second declares this about height-based discrimination. We may consider this a case of the opacity phenomenon discussed in Section 3.2 and Section 3.7 above, because the diagnostics are the same: substituting equivalent expressions u and v in some larger context C_D leads to expressions CuD and CvD which are no longer equivalent. The standard logical tool for handling opacity, intensionality, is not sufficient for handling these cases, since the ordinary English sense of *male or female* (which is not particularly sensitive to transgendered, chromosome-irregular, etc. people) is the same *everybody* as that of *tall or short* in every possible world. Since such cases of *hyperintensionality* can only be handled by abandoning the standard theory that Pollard (2008) called ‘The Peacable Kingdom of Natural Language Semantics’ (see also Section 5.6), it is of particular interest to see whether vector-based semantics can deal with them.

To the extent expressions like *tall* and *short* both contain a significant component of *height*, i.e. the scalar products ($tallR$, $heightR$) and ($shortR$, $heightR$) are large compared with the product of the lengths of these vectors, the sum $tallR+orR+shortR$ will also have a significant *height* component, and similarly *male or female* will have a significant *gender* component. This goes a long way toward explaining why one statement is about height-based and the other is about gender-based discrimination and not the other way around. Many fascinating questions remain, for example why *male or female* sounds considerably better than *?female or male* (Bolinger, 1962), but the mechanism is clearly capable of expressing basic facts about natural language that the truth-conditional approach is ill-equipped to deal with.

Can CVS models serve as a translation pivot? This is unlikely if R is computed from monolingual distribution data as described above, but there are many ways to compute embeddings, and if we base the computation on [parallel texts](#) useful results may be obtained. Another approach is to relate embeddings for different languages by linear transformations (Mikolov, Le, and Sutskever, 2013; Makrai, [in preparation](#)). How vectors can serve to represent sentential meaning is also an area in the early stages of exploration, and it will be sufficient to list here some of the main approaches.

The oldest one, due to Smolensky (1990) and originally proposed in the context of neural networks, is to use [tensor products](#) to encode variable binding. While in Section 3.3 we have already discussed the reasons for not permitting variables and variable binding in the prolepsis, the phenomena generally handled by these tools are still with us: for example, *Dick shot Harry* means something very different from *Harry shot Dick* but this is not reflected in a representation that is simply the sum of the three vectors $DickR$, $HarryR$, and $shotR$. Another phenomenon routinely handled by variables is the ‘binding’ of [anaphors](#): compare *First John insulted Mary, then he ridiculed her* to *First John insulted Mary, then she ridiculed him* and *First John insulted Bill, then he ridiculed him*. In the first two, we can resolve the ambiguity by relying on the gender of the pronouns, but in the last sentence both meanings are available. We will see many examples



in Section 7.1. The variable-free technique for handling such cases is discussed in Section 4.6.

Let us see how tensor products can be used to encode such distinctions. First we set up some attribute–value relations (in accordance with linguistic terminology, Smolensky talks about *slots* and *fillers* rather than attributes and values). In our case we need two slots: we can call these *agent* and *patient*, *subject* and *object*, *shooter* and *victim*, or *nominative* and *accusative*, or simply number them ‘1’ and ‘2’. The names of these slots are irrelevant, and indeed different grammatical traditions use different names; what matters is that in one case *DickR* fills the first slot and *HarryR* the second, and in the other case it is the other way around. To encode this distinction, we build the vector space $V \otimes V \otimes V$ from the original V , and consider $HarryR \otimes shotR \otimes DickR$. While tensor products are formally commutative (in the sense of $U \otimes V$ being isomorphic to $V \otimes U$), in any canonical basis fixed in advance the vectors $HarryR \otimes shotR \otimes DickR$ and $DickR \otimes shotR \otimes Harry$ are not the same, and in fact we can reconstruct the fillers, and their ordering, uniquely from each of these.

A second approach, due originally to Plate (1995), is to use circular convolution \circledast instead of the tensor product \otimes : for two vectors $\mathbf{x} = [x_1, x_2, \dots, x_n]$ and $\mathbf{y} = [y_1, y_2, \dots, y_n]$, the k -th component of $\mathbf{x} \circledast \mathbf{y}$ is defined as $\sum x_i y_{k-i}$, where $k - i$ is computed modulo n . This has the advantage of preserving the dimension of the input vectors, so that for *Dick shot Harry*, when represented as the sum of *shooter(Dick)R* and *victim(Harry)R*, both of these vectors will preserve the original dimension of *shooterR*, *victimR*, *DickR*, and *HarryR*. In the tensor system, we would have to make the uncomfortable choice between a representation that relies on a two-argument function (a member of $V \otimes V \otimes V$) and one that relies on the sum of two one-argument functions (a member of $V \otimes V$), or apply some ‘squishing’ function that reduces the higher dimensions.

The last approach to be discussed here is characterized precisely by such squishing operators: a typical example is the recursive auto-associative memory (RAAM) of Pollack (1990), where sequences of vectors $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k$ are reduced by a three-layer network that outputs, for each pair of n -dimensional vectors \mathbf{x}, \mathbf{y} a squished vector $\mathbf{s} = (a, b)S = h(A\mathbf{x} + B\mathbf{y})$, where A and B are $2n$ by n matrices. In the original RAAM neural net model, the matrices encode connection strengths and h is some sigmoidal activation function; in the linear version (Voegtlin and Dominey, 2005), h does not appear (it is the identity function). Starting with the zero vector as \mathbf{r}_0 , we first form $\mathbf{s}_1 = (\mathbf{r}_0, \mathbf{r}_1)S$, next we form $\mathbf{s}_2 = (\mathbf{s}_1, \mathbf{r}_2)S$, and in general recursively call the squisher S using the previously output \mathbf{s}_i and the next \mathbf{r}_i as its input.

Let us now return to the idea of structure sharing. Clearly, there is something in common between *bachelor*₁ ‘unmarried man’ and *bachelor*₄ ‘fur seal without a mate’, namely the inability to produce (legitimate) offspring. As noted in Kornai (2009), dictionary definitions often reflect highly outdated world-views, and catch up very slowly:

Thus, to go from the historical meaning of Hungarian *kocsi* ‘coach, horse-driven carriage’ to its current meaning ‘(motor) car’ what is needed is the prevalence

of the motor variety among ‘wheeled contrivances capable of carrying several people on roads’. A 17th century Hungarian would no doubt find the notion of a horseless coach just as puzzling as the notion of flying machines or same-sex marriages. The key issue in readjusting the lexicon, it appears, is not counterfactuality as much as rarity: as long as cloning remains a rare medical technique we won’t have to say ‘a womb-born human’.



But what do offsprings have to do with obtaining a *bachelor*₃’s degree? This seems to be somewhat related to being an apprentice, as is *bachelor*₂ ‘young knight without his own banner’. For an abstract structuralist like [Roman Jakobson](#), these four senses of *bachelor* all go back to one thing, being ‘unfulfilled in a traditional male role’. One striking claim of CVS representations is that the vectors *bachelor*_i all point in roughly the same direction (Mikolov, Yih, and Zweig, [2013](#)).

3.10 Further reading



While we have used one of [John McCarthy](#)’s examples, the question-answering perspective on semantics is characteristic not only of his thinking on the subject, but in fact of the whole field of artificial intelligence, and we find similar discussions throughout the work of the more psychologically inclined AI practitioners like [Roger Schank](#) as well. The roots of this approach in fact go back to the early psychological study of reading, in particular Thorndike ([1917](#)). In linguistics we commonly find a narrower definition of semantics, restricting attention to the study of truth-conditional implications and equivalences and delegating every other aspect of meaning to *pragmatics* ([Gazdar, 1979](#)). In subsequent chapters we will discuss several issues, such as the resolution of pronouns and indexicals, incomplete utterances, speech acts, implicature, discourse particles, that can be thought of as belonging more to pragmatics than to (truth-conditional) semantics, but we will simply speak of semantics rather than ‘semantics plus pragmatics’ all along. How the boundary between semantics (narrowly construed) and pragmatics should be drawn will be discussed in [Section 5.5](#).



The importance of Aristotle in the investigation of common sense reasoning can hardly be overstated. Modern exponents of (neo)scholastic thinking such as [Adler](#) would go as far as claiming that Aristotle *defines* common sense, that an introduction to Aristotle ([Adler, 1978](#)) is an introduction to common sense. From our perspective, Aristotle actually presents a highly coherent and nontrivial theory that goes far beyond what we would want to call ‘naive’ or ‘commonsensical’. This applies particularly to a central tenet of his ontology, that objects are composed of material substance and ideal form. One aspect of this view, namely that location is just one of the qualities an object has, similar to its shape, weight, or color, is related to the theory of fluents, for which see, for example, [Lambalgen and Hamm \(2005\)](#).

The model presented here owes a great deal to Aristotle, but it is not intended as a faithful reconstruction of his views and it is somewhat hard to place in the subsequent

scholastic tradition. In particular, it is almost trivial to cast the theory presented here as a hard-line realist theory that ascribes independent existence to abstract notions like ‘four-legged’, ‘beautiful’, or ‘from’. Yet when we define, for example, *dog* it is not the essential properties of dogs that we are after (clearly, these would very much include the wolf/dog genome) but rather the essential properties presupposed by the *word* ‘dog’ such as inferiority, which is not inscribed in the genome the same way as being hairy or four-legged is, but is a matter of cultural convention.

For a highly detailed formulation of the naive theories of space, time, emotions, planning, and other aspects of the naive theory, some of which we will touch on in subsequent chapters, see Gordon and Hobbs (2017). The idea that naive psychology can be modeled with automata is quite well established – for a thorough discussion, see Nelson (1982). The key observation, that nondeterministic automata can be said to have free will, goes back to the foundational paper on nondeterministic algorithms, Floyd (1967). In contemporary linguistics, it is perhaps Roy Harris who is most critical of the telementation theory; in fact he coined the word *just* to criticize the idea in three volumes, *The Language Makers* (Harris, 1980), *The Language Myth*, (Harris, 1981), and *The Language Machine* (Harris, 1987). As is often the case, the best way to study an idea is to learn about it from its opponents. We stay with the mainstream view that sounds can, and do, carry information.

It is rather unlikely that the Neo-Confucian notion of *li* (pattern, principle) or the Leibnizian concept of monads could be entirely faithfully reconstructed in terms of FSA, and in fact both of these are embedded in conceptual systems that go well beyond the ‘naive’ or ‘commonsensical’ views that we wish to formulate. That said, specific references to *li* are nearly always to the kind of regularities we recognize as commonsensical, and any theory of semantics should have the means at its disposal to be able to state them formally. This is particularly true of the kind of disembodied use of higher forces that is taken for granted in Chinese metaphysics: “Thus to say that ‘High heaven shook with anger’ by no means implies that there is a man up above who shakes with anger; it is simply that the principle (*li*) is like this [that is, that crime deserves anger].” (Graham, 1958)

As for monads, Kornai (2015) reconstructs these as cyclic FSA whose states are what Leibniz calls *perceptions* and whose transitions are automatically triggered by the next time tick; Leibniz calls this *entelechy*. This goes some way toward explaining key aspects of *Monadology*, in particular, the lack of inputs and outputs (see §7 of the *Monadology*) and the need for universal harmony (see §59), but again, our goal is not to fully reconstruct Leibniz’s system in contemporary terms but to provide a formal system that can account for perceptions, causality, and the like.

The best systematic introduction to Montague’s pioneering work is provided by Gallin (1975), who eliminates the minor inconsistencies across Montague’s original papers (which are collected in Thomason (1974)). More contemporary introductions include Dowty, Wall, and Peters (1981) and Gamut (1991). Modal logic goes back to Aristotle (*Organon*, Books 2–3) and was intensely studied in the *Middle Ages*. The



modern theory begins with Carnap (1946) and Carnap (1947), with the now standard treatment codified by Kripke (1963). The normal property, sometimes called the ‘Rule of Necessitation’, actually goes back to St. Thomas Aquinas, for whom things are necessary because God wills them so, but even the omnipotent God is bound by the rules of logic. For a thorough introduction to modern modal logic, see Lemmon, Scott, and Segerberg (1977) or Hughes and Cresswell (1996). For normal systems, the best source remains Hughes and Cresswell (1984), even though most of the material about normal systems is incorporated into Hughes and Cresswell (1996). We return to necessitation in Section 7.3.

The idea that word meanings emerge as the Boolean atoms of partitions created by different languages goes back to Apresjan (1965). “Bleen” and “grue” were invented by Goodman (1955); see also Swinburne (1968). For a detailed discussion of meaning postulates in MG, see Zimmermann (1999).

For a relatively recent survey of CVS semantics, see Clark (2015). A more detailed understanding of how meaning vectors line up in Euclidean space is still work in progress: we call attention to Levy and Goldberg (2014), who describe the popular `word2vec` embeddings of Mikolov et al. (2013) as implicit matrix factorization, and Arora et al. (2015), who relate vector length to log frequency and word cooccurrence to scalar product.

Graphs and Machines

Contents

4.1 Abstract finite computation	87
4.2 Formal syntax	94
4.3 The smallest machines	102
4.4 Graph and machine operations	104
4.5 Lexemes	107
4.6 Inner syntax	112
4.7 Further reading	119
4.8 Appendix: defining words	120

In the 1960s and 1970s, [flowcharts](#) were widely used for describing the structure of computer programs. In this chapter we generalize these information objects in two directions: instead of graphs we will use *hypergraphs*, and instead of finite state automata we will use a more algebraic formulation, the *machines* introduced by Eilenberg (1974) and now often called ‘Eilenberg machines’ or [X-machines](#).

We use hypergraphs and machines to model both the elementary linguistic structures known as [morphemes](#) and more complex ones such as words, phrases, sentences, and texts. In later chapters will also use these to deal with implication, knowledge representation, and in fact all issues of semantics broached so far, but here we concentrate on the central property of machines that makes them useful for semantics, the decoupling of the inner and the outer syntax.

In [4.1](#) we begin by defining simpler models of computation, gradually building up to hyperedge replacement graph grammars (Drewes, Kreowski, and Habel, 1997) and machines. In [4.2](#) we present the basic building blocks of the formal theory we will use in describing outer syntax, the syntactic congruence and the syntactic monoid. In [4.3](#) we look at the smallest machines, where the base set has 0, 1, or 2 elements, and explain how the relational structure that machines come equipped with can be used to encode the inner syntax. In [4.4](#) we begin to define operations on hypergraphs and machines, and in [4.5](#) we introduce *lexemes*, which are rather simple machines aimed at capturing the notion of morphemes and larger dictionary units. In [4.6](#) we describe how the effects of



variable binding can be obtained without variables. We illustrate the central techniques on the vocabulary of English, and present a defining vocabulary in 4.8.

4.1 Abstract finite computation



There are two broad classes of computer models: those that assume an infinite storage facility, such as the tape of a [Turing machine](#), and those that assume only finite storage. There are many intermediary classes, such as [linear bounded automata](#), whose working memory is set to be proportional to the size of the input, but our interest will be in strictly finite devices. We begin by refreshing our knowledge of some basic definitions we have already used informally in Section 3.6.

Definition 4.1 A *semiautomaton* over an alphabet Σ is given by a finite set of states S and some transitions $T \subset S \times \Sigma \times S$, i.e. a finite directed graph whose vertices are collected in S and whose edges, collected in T , are labeled by symbols from Σ .

This definition is permissive in regard to different edges starting at the same vertex being labeled by the same symbol, in regard to there not being edges out of, or into, some of the vertices, and in regard to there being multiple edges, bearing different labels, from and to the same vertices, except that edges that share all three parameters $\langle \text{start}, \text{label}, \text{end} \rangle$ are collapsed (not counted with multiplicity). If different edges starting at the same vertex are never labeled by the same symbol, we call the semiautomaton *deterministic*.

Definition 4.2 Given some (not necessarily finite) set X , its binary relations (all subsets of the Cartesian product $X \times X$) can be collected together in $\mathbf{FR}(X) = 2^{X \times X}$. Using relational composition as the product operation and the identity relation as the identity, this set becomes a monoid, called the (full) *relational monoid* over X . It is often convenient to treat relations as multi-valued partial functions from X to itself, and we retain the arrow notation $X \rightarrow X$ for these. Single-valued relations (partial functions) are called *transformations* of X and are collected together in the monoid $\mathbf{FT}(X) \subset 2^{X \times X}$. This set, using relation composition for the binary operation and the identity for the unary operation as before, forms a submonoid of $\mathbf{FT}(X)$, called the (full) *transformation monoid* over X .

Certain submonoids (sometimes just subsemigroups) of the relation monoid are of particular interest. For each letter $\sigma \in \Sigma$, a semiautomaton defines a relation T_σ on its base set S : we say $\langle a, b \rangle \in T_\sigma$ iff the triple $\langle a, \sigma, b \rangle$ is in T . If the semiautomaton is deterministic, these relations are transformations over S ; in the general (nondeterministic) case, they are just relations of S . It is possible, and often advantageous, to look at this matter from the standpoint of relations: if S is some (not necessarily finite) set and T_1, \dots, T_k are relations of this set, finite compositions of these can be equated with strings over the finite alphabet composed of the symbols T_1, \dots, T_k . If the underlying set X is finite, the relational monoid $\Phi \leq 2^{X \times X}$ generated by the T_i is itself guaranteed to be finite.

Exercise[◦] 4.1 Provide an example of an infinite base set S with a finite set of transformations T_1, \dots, T_k which lead to a finite transformation monoid. Provide a finite set of transformations T'_1, \dots, T'_k which lead to an infinite transformation monoid. For a finite set X with n members, how many elements will $\mathbf{FR}(X)$ and $\mathbf{FT}(X)$ have?

Definition 4.3 A *finite state automaton* (FSA) is a semiautomaton with two distinguished subsets $I \subset S$ and $F \subset S$, called the *initial* and the *final* states, respectively. (The final states are also called *terminal* or *accepting* states.)

We expect the reader to be rather familiar with FSA and [regular expressions](#), and concentrate only on some aspects of the theory less commonly taught. The key point is that each FSA defines a [formal language](#) as containing those and only those strings of Σ^* which transform at least one initial state of the machine into at least one terminal state. In linguistics it is customary to replace the nondirectional idea of ‘defining’ a language by more specific terms that relate to the operational *mode* of the automaton. As in computer jargon, where it is common to collect different but closely related operational characteristics together, such as in the ‘color mode’ of Photoshop or the ‘EBCDIC/ASCII mode’ of IBM mainframes, here we speak of the *generation* or *analysis* mode of an FSA.

For a more general example, we will use the Pythagorean theorem. This can be used in *checking mode*: given a triangle, we can measure its three sides and use the theorem to verify if it is indeed a right triangle. It can also be used in *construction mode*: if we wish to create a right angle, a string with 12 evenly placed knots can be laid out to form a triangle with sides 3, 4, and 5. Finally, the theorem can be used in *computation mode*, or, rather, three distinct computation modes: if we are given two of the three sides a, b, c of a right triangle the third one can be computed.

With FOL, it is not hard to capture these modes. The Pythagorean Theorem itself can be stated as *triangle, right, a, b, c* $\Rightarrow a^2 + b^2 = c^2$. We are suppressing some of the geometrical complexities by having a on the left-hand side abbreviate the statement `triangle HAS side, side HAS length, length EQ a`; and on the right-hand side it stands simply for the length of the side. We are further suppressing some difficulties in guaranteeing that the `side` that appears in `triangle HAS side` is actually the same side that appears in `side HAS length`, not because the issue is trivial, but because it has no bearing on the issue of modes. As we shall see, machines are more complex than FSA (in fact, they come complete with a little FSA built in) and, accordingly, will have many more operational modes.

Exercise[◦] 4.2 Prove that for each finite subset L of Σ^* there exists an FSA \mathcal{L} that defines it. Find an infinite language R that can be generated by FSA. Find an infinite language P that cannot be so generated.

While the notion of initial and terminal states is so simple as to require no additional formalism, we will nevertheless reformulate these in terms of mappings, because the general idea will come in handy in a moment. According to the von Neumann definition of ordinals, $\mathbf{1}$ is defined as the singleton set $\{\emptyset\}$ containing the empty set as its only member. We will continue to use 1 to denote the identity morphism, but boldface $\mathbf{1}$ will stand for any arbitrary singleton set – this should lead to no confusion, as we will



be interested only in mappings to and from this set. Given a semiautomaton $\langle \Sigma, T \rangle$, we define an FSA by adding an *initial map* $\alpha : \mathbf{1} \rightarrow S$ and a (partial) *terminal map* $\omega : S \rightarrow \mathbf{1}$. Instead of the subset I we will use the range (codomain) of α , and instead of T we will use the domain of ω . Now we can say that the string $\sigma_1 \dots \sigma_k$ is *accepted* or *generated* by the FSA iff $\langle 1, 1 \rangle \in \alpha \sigma_1 \dots \sigma_k \omega$.

Notation 4.1 To some extent we are departing from the notational conventions of computer science, in that we will not find it necessary to always list every defining item of a structure. For example, we can denote the same FSA \mathcal{A} by $\langle \Sigma, S, T, I, F \rangle$ or by $\langle S, T \rangle$ or $\langle S, \alpha, \omega \rangle$, depending on which part of the structure we need to emphasize. We will also follow Eilenberg in passing over the obvious mappings between relational tuples such as $\langle \langle a, b \rangle, c \rangle$ and $\langle a, \langle b, c \rangle \rangle$, treating both as being identical to $\langle a, b, c \rangle$. The **empty string** will be denoted by λ .



After these preparations, we are ready to define the key notion of this section and, indeed, of the whole book. Informally, a *machine* is a finite state automaton whose alphabet has been mapped onto the relational monoid of a set X that need not be the state set. We have seen above how symbols of the alphabet act on the states of a (semi)automaton. Here we assume the existence of a different set X , some of whose relations $\phi \subset X \times X$ are invoked by letters of the alphabet. Our interest will be in the transformation monoid Φ generated by the ϕ that appear in the range of this mapping.

Definition 4.4 A *machine* with an alphabet Σ over a base set X is given by an *input set* Y , an *output set* Z , a relation $\alpha : Y \rightarrow X$ called the *input code*, a relation $\omega : X \rightarrow Z$ called the *output code*, a finite state automaton $\langle S, T, I, F \rangle$ over Σ called the *control FSA*, and a mapping M of each $\sigma \in \Sigma$ to some $\phi \in \Phi \leq 2^{X \times X}$.

Comp

Since machines will play a critical role in formulating the abstract algebraic (as opposed to the logic- or vector-based) theory of semantics, some anticipatory remarks are in order. Readers primarily interested in obtaining a better feel for machines as computing devices should pay attention to Section 4.3, where several simple examples are provided. This is deferred until after Section 4.2 only because there are even simpler devices, the FSA defined in Definition 4.4, and the FSTs defined in Definition 3.3, which are already capable of doing much work, and we discuss these first.

Ling



Traditional linguistics puts the emphasis on words, and all grammars written before the 1960s devoted the bulk of the effort to phonology (of which we will have little to say in this book) and **morphology**, a subject we will turn to in Section 5.2. Starting with Chomsky (1957) and Chomsky (1965), generative grammar has to a remarkable extent succeeded in inverting this emphasis and concentrating the effort on syntax. In this book, given the information-theoretic reasons spelled out at the beginning in Section 1.3, we revert to the traditional view and spend more effort on crafting the model of individual words (lexemes, see Section 4.5), and their network, the lexicon; see Chapter 6. We keep syntax simple by distinguishing *outer syntax* or phenogrammar (Chapter 5) from *inner syntax*, which can be performed by machines or even simpler FST/FSA notions (see Section 4.6). But before we can turn to any of these, we introduce a more static notion, better suited for combinatorics than algebra, that of a hypergraph.

Definition 4.5 An (edge-labeled, finite) *hypergraph* with an alphabet (label set) Σ , a (finite) vertex set V , and (finite) hyperedge set E is defined by a mapping $\text{att}:E \rightarrow V^*$ that assigns a sequence of pairwise distinct attachment nodes $\text{att}(e)$ to each $e \in E$ and a mapping $\text{lab}:E \rightarrow \Sigma$ that labels each hyperedge. The size of the sequence $\text{att}(e)$ is called the *type* or *arity* of the label $\text{lab}(e)$. As machines come with input and output mappings, hypergraphs come with a sequence of pairwise distinct *external nodes* denoted ‘ext’. This sequence may be empty, a choice that makes the more standard notion of [hypergraphs](#) a special case of our definition.



Our main interest will be in replacing a hyperedge e by some hypergraph H so that only the attachment nodes of e are kept and these are fused with the external nodes of H , respecting the ordering of $\text{att}(e)$ and $\text{ext}(H)$. Intuitively, att nodes of hyperedges correspond to both input and output in function application, so that a function f that depends on *two* arguments and produces one output will correspond to a hyperedge with *three* att nodes, say 0 for the output, and 1 and 2 for the inputs (in this order) to f . In terms of logic formulas this means that constants, and only these, correspond to hyperedges that have exactly one att node. We return to the issue of connecting machines and hypergraphs in Section 5.8, where we introduce *valuations* (mappings to small ordered sets of values).

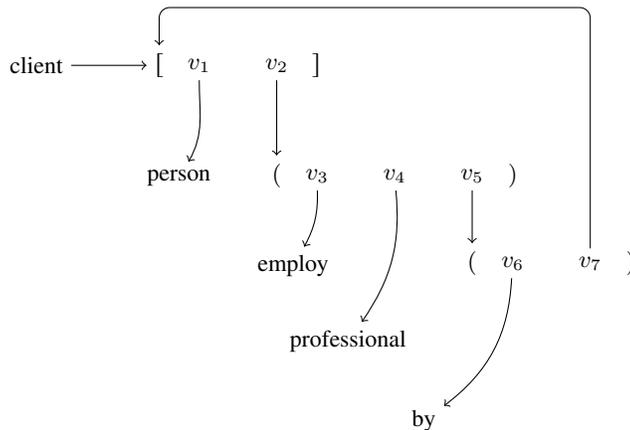


Fig. 4.1. Defining *client*

Quillian (1969) was the first to define words using a hypergraph where each word corresponds to a hyperedge. Here we consider the word *client*. To present the definition depicted in Fig. 4.1 as a hypergraph, let us number the vertices v_i from left to right, and top to bottom. The entire definition of *client* consists in only two statements, the arrow from v_1 to *person*, and an arrow from v_2 to a more complex three-vertex hyperedge e composed of v_3, v_4 , and v_5 . Anticipating matters somewhat, we will say that a client IS_A person such that e holds. e in turn is a relation of *employment* (v_3) between some

professional (v_4) and some hyperedge f composed of by (v_6) and v_7 , where v_7 points back to *client*.

Webster's Third defines *client* as ‘a person who engages the professional advice or services of another’ and Quillian encodes in his system ‘a person who employs a professional’. Both of these definitions aim at distinguishing *client* from *employer*, ‘a person who employs another’, by somehow making it clear that the person being employed or, better yet, the services provided by that person for the client, must be of a professional nature.

Exercise^o 4.3 Suppose a summer resort employs a tennis pro to teach the children of the paying guests to play better tennis. Who is her employer? Who are her clients: the children, their parents, or the resort? The resort also hires a lawyer fallen on hard times to mow the lawn. Is the resort now his client?

Perhaps the most challenging aspect of definitions of this nature is their apparent circularity. The *client* hyperedge points to *employ* (e), *employ* points to a *by*-phrase f , while f points back to *client*. As we briefly discussed in Section 3.8, by decoupling the knowledge representation from the details of English syntax, we can easily circumvent the problem that was created here by the use of the [passive voice](#). Instead of the indirection through f , we will say directly that `client EMPLOY professional` or, better yet, `client EMPLOY service, service IS_A professional`.



Ling

As a rule, we will not spend a great deal of energy on trying to decide whether *professional* is a noun or an adjective, even though the English paraphrases *Jack is a professional* and *Jack is professional* would differ somewhat. Such differences can be sufficient for distinguishing different senses of the same word; compare *Jack is blond* ‘has blond hair’ to *Jack is a blond* ‘fulfills the criteria for the stereotype’. But this is a peculiarity of English syntax, and as such it has no place in the semantics. There is a somewhat related distinction often made in logic between *constants* and *variables* (see our discussion in Section 2.4), which is often seen as necessitating a distinction between $P \text{ IS_A } Q$, taken to mean $\forall x, x \in P \Rightarrow x \in Q$ or $P \subset Q$, and $c \text{ INSTANCE_OF } P$, referring to $c \in P$. For our purposes, this is a distinction without a difference, required in classical logic because strict typing is necessary to fend off [Russell's paradox](#), but not at all relevant to the logic of generics and prototypes that drives the natural language semantics that we discussed in Section 3.7. In the system we will present here, this particular graph no longer has a cycle, because we will know from the edge labels 0 (`IS_A`), 1 (`subject`), and 2 (`object`) that the client is the person who does the employing.



Comp

To obtain the graph depicted in Fig. 4.2 from the definition of *client* as given in *Webster's Third*, one needs to perform a tremendous amount of work. First, we need to parse the definition, a job we leave in the hands of the [Stanford Parser](#). (This may come as something of a disappointment to readers who wish to understand the many intricate ways syntax and semantics affect each other, but see our remarks in Section 3.7.) The parser returns a *parse tree*, in this case the one depicted in Fig. 4.3



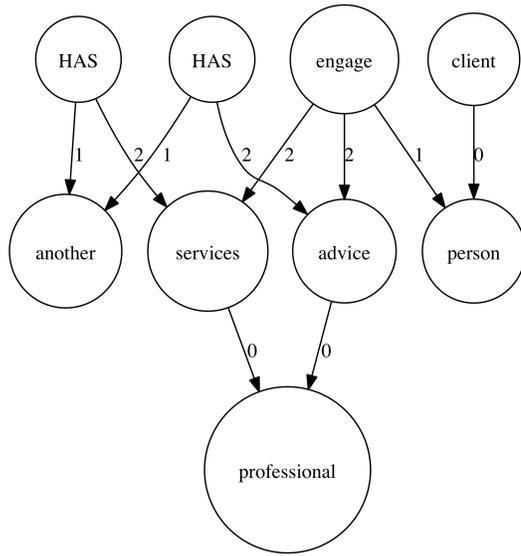


Fig. 4.2. client graph obtained from definition in Webster’s Third

```

(NP
 (NP (DT a) (NN person))
 (SBAR (WHNP (WP who))
 (S (VP (VBZ engages)
 (NP (NP (DT the) (JJ professional) (NN advice))
 (CC or)
 (NP (NP (NNS services))
 (PP (IN of)
 (NP (DT another))))))))))
  
```

Fig. 4.3. Parse tree of the definition

The parser actually makes an error, attaching the prepositional phrase *of another* to *services*, rather than to the entire noun phrase *professional advice or services*. Since **PP attachment** is still problematic for the current generation of parsers (recall our example in Section 1.1 about the man on the hill with the telescope), the 4lang system has to fix it in the mix during the second stage, where both *services* (actually, the singular *service* rather than the plural *services*) and *advice* are linked as *professional* things that *another* HAS.



At the price of introducing edge labels, we have eliminated circularity from this definition and from many others (though not necessarily from all of them), but we have obviously not eliminated it from the entire system of definitions. Here we see *client* defined in terms of *person*, *engage*, *professional*, *another*, *service*, *advice*, and conceptual primitives such as HAS, AND, OR, plus three different types of node labels 0, 1, 2. What happens if the definition of *professional* makes reference to clients? We

discuss this matter in Section 4.5 below, but the idea should be clear already: this is no more detrimental than saying that x and y are defined by $x = 3y + 1$ and $y = 2x - 3$.



One significant difference between the system presented here and widely used systems of knowledge representation such as [Freebase](#) or [DBpedia](#) is the size of the link inventory: we only have 0 (attribution, IS_A), 1 (subject), and 2 (object), while fact collections rely on tens of thousands of different links such as *StarredIn* to express such facts as ‘Morena Baccarin starred in *Gotham*’. That the profligate use of link types is a significant problem has been known since Woods (1975), and here we part with the KR tradition and follow the tradition of grammar instead. Aside from IS_A (which actually can be eliminated at the cost of making the system far less compact; see Section 4.5), our labels are called NOM and ACC in case grammar (Fillmore, 1977), 1 and 2 in relational grammar (Perlmutter, 1983), AGENT and PATIENT in linking theory (Ostler, 1979), and nsubj and dobj in Universal Dependencies (see Section 5.4). The label names themselves are irrelevant, what matters is that these elements are not part of the lexicon in the same way as the words are. We defer the matter of treating *phrasal verbs* such as ‘star in’ until linear order is discussed in Section 5.3.

4.2 Formal syntax

When we have some elements (conventionally called *letters*, though in typical cases we think of word-sized units) and we are interested in studying how the elements can occur together (‘syn’) in an order (‘taxis’), we speak of *syntax*. This is a well-studied subject, and there is simply no way we can present, within the bounds of this volume, even the basic material covered in introductory courses. In this section we will concentrate on introducing the formal apparatus that will be indispensable for dealing with syntactic phenomena in natural language. Some specific issues, such as the treatment of idioms, are covered, but many others, such as agreement or headedness, are deferred to Section 5.3.

The formal theory begins with a set Σ where we collect the letters, and a set of strings L where we collect the permissible linear arrangements (strings) of the letters. If all arrangements are permissible, we obtain the free monoid Σ^* with unit element λ , the empty string, but if some arrangements are not permitted the language in question will be a proper subset of Σ^* . In these cases, strings outside L are called *ungrammatical* and are generally noted in linguistics by a prefixed * as in **John tea drinks* as opposed to the permissible (grammatical) order *John drinks tea*.

Example 4.1 Our first language, L_1 , will have three kinds of symbols: g or good symbols give rise to grammatical strings; b or bad symbols will render any string ungrammatical; and n or neutral symbols are such that leaving them out will preserve the (un)grammaticality of any string. (We can think of such n elements as [filled pauses](#).) Thus the grammatical strings are $\{g, n\}^+$ and the ungrammatical ones are $\Sigma^*b\Sigma^*$. As usual, there is no direct evidence bearing on the grammaticality of the empty string λ , and the definition of L_1 given above leaves the matter open: one could argue $\lambda \in L_1$



based on the fact that it contains no b , but one could also argue $\lambda \notin L_1$ based on the fact that it contains no g either. For the sake of definiteness, we denote the first choice by L_1 and the second by L_1^0 .

It is evident that in such a simple case all syntactic information about some string of letters such as $xyzy$ can be gleaned from knowing whether the elements $x, y,$ and z that appear in it are themselves good, neutral, or bad. In fact, the basic alphabet Σ may even be infinite; what matters is just the mapping $\Sigma \rightarrow \Gamma = \{g, n, b\}$ that tells us whether a given letter from Σ is good, neutral, or bad. In more complex cases, this alone may not be sufficient, since good elements, such as *tea*, *drinks*, and *John*, may need to appear in a permissible order for the result to be considered grammatical. Yet the basic idea of sorting the letters into various categories works very well; we just need to be more careful about establishing the categories.

Definition 4.6 Given a language $L \subset \Sigma^*$, the *syntactic congruence* \sim_L induced by L on Σ^* is defined to hold between two strings $\beta, \delta \in \Sigma^*$ iff for all strings $\alpha, \gamma \in \Sigma^*$ we have $\alpha\beta\gamma \in L \Leftrightarrow \alpha\delta\gamma \in L$, that is, if substituting δ for β in any context will not change the membership status (called in linguistics the **grammaticality**) of the string. If we set α or γ to be the empty string, we obtain the definition of the *right* or *left* syntactic congruence respectively. For any string β , the set of contexts α, γ that make $\alpha\beta\gamma \in L$ is called the *distribution* of β . Thus, two strings β and δ will be syntactically congruent iff their distributions are identical. Since \sim_L is a congruence, we can form the usual quotient structure (see Section 2.2).



Definition 4.7 Given a language $L \subset \Sigma^*$, the *syntactic monoid* Σ^*/L is formed by taking the equivalence classes of \sim_L as its elements and defining the product of two classes as the class of any two representatives.

It is clear that for the language L_1 of Example 4.1 we have $g \sim n$, and we require only two classes: $[\lambda]$, the equivalence class of good strings; and $[b]$, the equivalence class of bad strings. Denoting these by 1 and 0, respectively, the multiplication table of the syntactic monoid will be as shown in Table 4.1.

	1	0
1	1	0
0	0	0

Table 4.1. Multiplication in $\{g, n, b\}^*/L_1$

Turning to the language L_1^0 of Example 4.1, we note first that the equivalence class $[\lambda]$ of the empty string is neither 1 nor 0. Even though λ is defined to be ungrammatical, it is in a different congruence class from b , because $g\lambda = g \in L_1^0$ while $gb \notin L_1^0$. Similar reasoning proves that λ is not in the congruence class of good strings either, since in the empty context (using λ for both α and γ in Definition 4.6) we have $\lambda\lambda\lambda = \lambda \notin L_1^0$ but we have $\lambda g\lambda = g \in L_1^0$. Thus in the syntactic monoid of L_1^0 we have at least one more class, which we will denote by e , and the multiplication table shown in Table 4.2.

	e	1	0
e	e	1	0
1	1	1	0
0	0	0	0

Table 4.2. Multiplication in $\{g, n, b\}^*/L_1^0$

Exercise^o 4.4 Verify that Table 4.2 is the correct multiplication table for Σ^*/L_1^0 .

In a monoid, by definition, we always have an identity element e . If we have nothing else, this is the *trivial monoid* M_1 . Since the row and column defining left and right multiplication by e are completely predictable, omitting them from the multiplication table can cause no confusion, except for the following: we must *know* whether the convention to omit has already been applied or not. Table 4.2 is not the same as Table 4.1, corresponding to the fact that the monoid Σ^*/L_1 is not isomorphic to the monoid Σ^*/L_1^0 , the former having two, the latter three distinct members. Over two elements (counting the identity) there are only two monoids, the one given in Table 4.1 (where the identity element of the monoid is denoted by 1), and another one given in Table 4.3.

	1	t
1	1	t
t	t	1

Table 4.3. Multiplication in C_2

As the reader will notice, Table 4.3 differs from Table 4.1 in two respects: first, that the element 0 has been renamed t , and second, that $0 \cdot 0$ was 0 while $t \cdot t$ is 1. In other words, t is invertible; in fact $t^{-1} = t$, and C_2 is not just a semigroup, but the familiar cyclic group of order 2. M_2 , on the other hand, is a true semigroup; not only does 0 fail to be invertible, but also we cannot in fact embed M_2 in any larger group.

Exercise^o 4.5 Why?

The first question that naturally arises from the preceding considerations is what language, if any, will give rise to C_2 as its syntactic monoid? As a moment's thought will show, $(aa)^*$ over a one-letter alphabet will suffice, and adding neutral elements n_1, n_2, \dots will only grow the alphabet size without changing the syntactic congruence.

Exercise^o 4.6 What language, if any, will give rise to the cyclic group over k elements, C_k , as its syntactic monoid?

Example 4.2 As our next examples, consider $\Sigma = \{a, b\}$ and the languages $(ab)^*$ and $(aa)^*$. Both require a two-state FSA with $S = \{0, 1\}$, $I = \{0\}$, $T = \{1\}$, and an edge $\langle 0, a, 1 \rangle$. The difference is that in the $(ab)^*$ automaton the reverse edge is $\langle 1, b, 0 \rangle$, while in the $(aa)^*$ automaton it is $\langle 1, a, 0 \rangle$ (Fig. 4.4).



Fig. 4.4. Automata accepting $(ab)^*$ and $(aa)^*$

Exercise^o 4.7 Prove that the automata defined above indeed define the languages $(ab)^*$ and $(aa)^*$. Prove that these are the minimal automata to do so. Compute the syntactic congruence and syntactic monoid associated with these languages.

Notation 4.2 In the study of FSA and FSTs, it is convenient to denote initial states by little arrows leading to them and final states by little arrows leading out. Whether the 1 that is the source of the arrows denoting the initial state or the 1 that terminates the arrows denoting the final state should be added to the “minimal” state count is debatable, but in the weighted case, especially when the weights are interpreted as probabilities, it simplifies matters a great deal to have an initial state with λ transitions leading to the initial states, and a final ‘sink’ or ‘no return’ state (Fig. 4.5). In computer science this state is often suppressed by convention, yet it needs to be added back if one wishes to identify machine states with the elements of the syntactic monoid as suggested above.

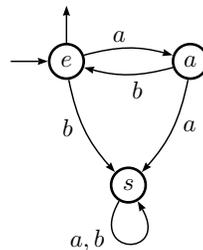


Fig. 4.5. Automaton accepting $(ab)^*$ (sink state included)

Exercise^o 4.8 Over the one-letter alphabet $\{a\}$, consider the [POSIX extended regular expression](#) `/^a?$(aa+?)\1+$/`, which will match a^p iff p is prime. Can you write a non-extended regular expression (without using a backtrack variable) to perform the same task?



By now we have at hand most of the machinery we will need to do formal syntax. As we have seen, there are several information objects (we use this term broadly, to include both algebraic structures and the [data structures](#) familiar from computer science) that play a role: these are summarized in Fig. .



We begin with some *expression* \mathcal{E} such as a regular expression used to define regular languages, or a rule system written in the TwoLC formalism (Karttunen and Beesley,

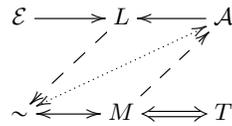


Fig. 4.6. Information objects associated with languages

2003) used to define regular transductions (rational relations). These are highly compact representations, capable of defining with a few hundred symbols automata or transducers containing millions of states. The relationship between expressions \mathcal{E} and languages/relations L is unidirectional in the following sense: given expressions of restricted complexity such as regexps or TwoLC grammars, we have quite effective algorithmic means to generate the languages or to establish whether a particular string is covered by these, while the converse is spectacularly untrue: given a regular language or relation defined, for example, by an automaton or transducer, it is an extremely hard problem to find a simple regexp or TwoLC grammar that defines it. Some cases of this problem are candidates for being cryptographic [one-way functions](#).



The same asymmetry is true of the relationship between regular languages or relations and their automata (or transducers). Providing the automaton or transducer \mathcal{A} is perhaps the easiest way to capture a finite state language or regular relation, respectively, in a compact manner, paving the way for very efficient enumeration and membership-testing algorithms. The converse problem of learning the (minimal deterministic) automaton given a language is very hard; for a survey, see Angluin and Smith (1983). The dashed arrow from the language L to the syntactic congruence \sim is unidirectional in a different sense: a language uniquely defines a syntactic congruence, but the converse is not true: different languages may define the same congruence. In particular, we have the following result.

Exercise^o 4.9 The syntactic monoids of an arbitrary language $L \subset \Sigma^*$ and its Boolean complement will always be isomorphic.

Broadly speaking, the syntactic congruence provides information about the semiautomaton, with the initial and accepting states left unspecified, rather than the fully specified automaton. (This is only broadly true; in reality the automaton depends more on the right congruence than on the full congruence, a matter that can be rectified by considering *reversible* languages, for which a string $a_1 a_2 \dots a_{n-1} a_n \in L$ iff $a_n a_{n-1} \dots a_2 a_1 \in L$. Since restricting ourselves to reversible languages is not very satisfactory, we return to this matter in Section 4.4.)

Exercise^o 4.10 Connect the dots in the dotted arrow in Fig. 4.6 by proving that the equivalence classes of the right congruence \sim'_L can be identified with the states of the minimal deterministic finite automaton \mathcal{A} defining the language L .

The relationship depicted in the bottom row of Fig. 4.6 between the syntactic congruence \sim and the syntactic monoid M is one long familiar from algebra, since M is obtained from factoring the free monoid Σ^* by \sim . For the relationship between M and

its multiplication table T we use an even stronger arrow, because these are mutually definitional: a monoid is *given* by its multiplication table and the table is trivially computed if the monoid is given. Algebraic identity (isomorphism) of two monoids M and M' is the re-lettering of their multiplication tables T and T' and, conversely, any such re-lettering leaves the algebraic identity of the defined monoids unchanged.

The factor monoid Σ^*/L (sometimes denoted Σ^*/\sim_L) provides in a compact form all the syntactic information we have about L . We illustrate this with a series of increasingly complex examples.

Exercise[→] 4.11 Given an arbitrary finite group G , what language, if any, will give rise to G as its syntactic monoid?

We now briefly turn to describing all monoids over three elements. Ignoring the row and column corresponding to the identity, we have two other elements a and b , and the four slots in the multiplication table can each be filled by three elements e, a , or b , so the maximum number of semigroups to consider would be $3^4 = 81$. The *actual* number of semigroups will of course be smaller, for two reasons: first, because not every potential multiplication table will work (associativity may not hold), and second, because different tables can correspond to the same (isomorphic) semigroups.

Exercise[→] 4.12 Describe all semigroups with exactly three elements (counting the identity).

As Fig. 4.6 made clear, the study of the syntax of a language L is intimately linked to the study of the syntactic monoid M obtained by factoring Σ^* by the syntactic congruence (see Exercise 2.9 on page 21). Of particular interest are the *lexical categories* and *lexical subcategories* of L . Here the naming traditions of mathematics and linguistics differ significantly and we need to disambiguate. What mathematicians call *letters* of the alphabet are generally called the *lexicon* in syntax. Accordingly, mathematicians prefer to talk of *strings* composed of letters, while syntacticians prefer to talk of *sentences* composed of *words* or *lexemes*. The distinction is readily reflected in the cardinality of Σ : typical mathematical examples involve only one, two, or a handful of letters, while the cases of syntactic interest often involve tens of thousands of different words. Since we have so many words, the relational footprint of the syntactic congruence on the set of words, $C_L = \sim_L \cap \Sigma \times \Sigma$, is already of great practical interest.

Exercise[◦] 4.13 Prove that C_L is an equivalence relation over Σ . Is it an equivalence relation over Σ^* ? Is it a congruence relation?

Definition 4.8 We call the equivalence classes of C_L *lexical categories*.

Those familiar with linguistic terminology will know that, in linguistics, these classes are called *subcategories* rather than categories (see Sections 5.4 and 6.3 for further discussion). Here we will stay with the distributionally inspired terminology, using ‘lexical category’ to designate collections of those words that share an identical, not just similar, distribution, especially as there is little chance of confusion with mathematical *categories*. From a technical standpoint it is also important that we use words, rather than



morphemes, since the within-word distributions of two morphemes are rarely identical. Thus when we say that *hat* and *coat* have identical distributions we ignore, by definition, the fact that there is a profession of *hatters* but there are no ⁰*coaters* (see Section 5.2 for further discussion of the raised ⁰ notation, signifying *accidental gaps*). By the same token, in the analysis leading to C_L an important simplification is effected by removing set phrases from L , a move justified by the fact that set phrases are listed in the lexicon.

To see how this affects the monoid, we divide English, taken as a formal language E , into two disjoint parts S and I , where S is ‘simplified’ English without the idioms, and I is the language of idioms. The question is now twofold: first, to what extent will \sim_E differ from \sim_S , and second, to what extent does the individual membership of words in the congruence classes change? Since idioms are generally frozen expressions that started out life as understandable, the change between the two congruences is negligible. Consider *John and Mary are at loggerheads*. It is a mystery to those not familiar with the idiom what these loggerheads are, and consulting the dictionary offers little help, as we find the following:

1. A loggerhead turtle.
2. An iron tool consisting of a long handle with a bulbous end, used when heated to melt tar or warm liquids.
3. *Nautical* A post on a whaleboat used to secure the harpoon rope.
4. *Informal*
 - a) A blockhead; a dolt.
 - b) A disproportionately large head.

Unless the dictionary has a specific entry saying *to be at loggerheads* means ‘to be engaged in a dispute’, the reader will never get a hint of the idiomatic meaning. Yet the overall system of syntactic categories is undisturbed by this, since the sentence has the same form as *The balls are at rest* or *The kids are at school* and we simply assign *loggerheads* to the same category of stative nouns that, for example, *school* belongs to. As for our second question, \sim_E and \sim_S may be the same, but the membership of individual words in individual classes can sometimes change as a result of taking idioms into account. Consider *John and Mary tripped the light fantastic*. In normal usage, *trip* is intransitive; one can say *We tripped (on acid)* but one cannot say **We tripped the Sahara* (as opposed to *We travelled the Sahara*). In general, lexicographers will choose to ignore the idiom and assign *trip* to the intransitive class, both because doing otherwise would create the mistaken impression that *trip* freely occurs with an object and because the opposite choice would not bring us any closer to the goal of describing *to trip the light fantastic*, an expression whose meaning ‘to dance’ is completely unpredictable from the meaning of its parts.

Ling With categories and subcategories comes the notion of *strict subcategorization*. Traditional grammar finds it convenient to lump together in a single category such as *conjunction* words that have only superficially similar functions and hugely different distributions: consider the ‘coordinating conjunction’ *or* and the ‘subordinating conjunction’

that. It is evident that *that* can rarely be replaced by *or* and *or* can practically never be replaced by *that* in any sentence in a grammaticality-preserving fashion. One noticeable difference between these two elements is that they have different arity: *or* requires two arguments of the same category (consider *S or NP constructions like **China is industrializing rapidly or John*) while *that* requires only one, typically a tensed sentence (as opposed to tenseless; compare *It is not surprising that Mary met/meets John* to . . . **that Mary meet John*). In such situations we speak of a word subcategorizing for its argument(s), a notion particularly helpful for verbs, which we prefer to lump together in a major category V, but we must at the same time recognize that some of them fall into distinct subcategories; compare **John appointed/renounced*, verbs that demand some object, with *John ate*, which does not, even though conceptually it is evident that there is no act of eating that doesn't involve some food object, just as there is never an act of appointing without an appointee.

Another linguistic notion that we will rely on is that of a *selectional restriction*. It is not enough to say that a verb like *elapse* requires a subject, since English verbs require a subject with such intensity that if no semantic subject is available a **dummy** must be provided, as in *It was raining*. Rather, to distinguish *elapse* from verbs in general we must say that it *selects for* not just any subject, but a temporal phrase; compare *Three months elapsed* with **The anvil elapsed*. The terminology is rather ambiguous in terms of the direction: is the temporal phrase the enabler that makes it possible to choose the verb *elapse*, or is it the choice of the verb that forces us to use a temporal phrase? Is this even a matter of grammar? In general, abstract nouns are rather distinct from physical objects, and it is hard to assign those adjectives to the former that are common to the latter; consider ?*The idea is green with orange stripes* or ?*The proof pulsated for a long time*.

Selectional restrictions are more typical of objects than subjects, and it is not at all trivial to pin down their scope based on a single language. For example, the Hungarian verb *fájlal* 'feel pain' is felicitous both when it is used for a body part *János fájlalja a lábát* 'John feels pain in his leg', and for external objects, *János fájlalja a bizottság döntését* 'John regrets the decision of the committee', while in English one could hardly regret one's leg or feel pain in the decision. Armed with a bit more lexicographic knowledge, we will return to these issues in Sections 5.7 and 6.3 but we emphasize here that selectional restrictions are of a different, softer character than strict subcategorization. Wilks (1978) argued as follows:

Mr. Wilson said that the line taken by the Shadow Cabinet, that a Scottish Assembly should be given no executive powers, would lead to the break up of the United Kingdom. (The Times: February 5, 1976) [. . .] anyone setting out to write down the selection restrictions for the objects of the verb take would not want to write then in such a way that lines could be said to be taken [. . .] Whether or not we want to call such usage "metaphorical," it is the norm in ordinary everyday language use, and cannot be relegated to the realm of the exceptional, or the odd, and so dealt with by considerations of "performance" in the sense of (Chomsky,



1965). On the contrary it is, I shall argue, central to our language capabilities, and any theory of language must have something concrete to say about it. Even if the newspaper usages above are “extended,” I would suggest that anyone who could not grasp these extensions could not be said to understand English properly.

In Section 6.4 we will discuss how to turn this into a methodological principle, *monosemy*, that claims ‘one word, one meaning’, putting the burden of proof on those who assume several meanings where a single ‘extensible’ one could do the job. Here we flesh out some criteria that may help distinguish the two ranges of phenomena. First, strict subcategorization has typecasting power; cooccurrence restrictions do not. For example, if we read that *A sekki elapsed* we know that this must refer to some time period even if we don’t know the details of the *sekki* system. In contrast, while it is true that physical objects are given far more often than abstract ones, there is no need to typecast *executive powers* into some physical object for the example above to make sense. Second, violations of strict subcategorization are perceived as violations of grammar, while violations of selectional restrictions are seen as problems with the belief system of the speaker. To quote McCawley (1970):



While some linguists might suggest that a person who says things like *My toothbrush is alive and is trying to kill me* observes different selectional restrictions than normal people do, it is pointless to do so, since the difference in ‘selectional restriction’ will correspond exactly to a difference in belief about one’s relationship with inanimate objects. A person who utters this sentence should be referred to a psychiatric clinic, not to a remedial English course.

Third, and this is not necessarily a criterion distinct from the second, strict subcategorization can generally be stated in terms of well-established and often ‘grammaticalized’ general features such as case, gender, tense, aspect, etc.; selectional restrictions rely on more specific features, generally restricted to small sets of words, often a single word. While selectional restrictions can easily be suspended in fairytales or science fiction, or even in simple attitudinal contexts such as *John believes that . . . , I dreamed that . . . ,* or *Nobody in his right mind would claim that . . . ,* subcategorization (for example, for aspectual marking) is not so easy to circumvent: compare *I dreamed that Max knew the answer* with **I dreamed that Max was knowing the answer*.

4.3 The smallest machines

Here we survey the simplest machines in order of increasing base complexity. To this end, we need another piece of terminology: the *behavior* of a machine is defined in terms of the strings its control FSA generates: if $\sigma = \sigma_1 \dots \sigma_k$ is such a string and $\sigma M = \phi_1 \dots \phi_k$ is the sequence of relations that M maps this to, the behavior associated with this string is the relational composition $\alpha\sigma_1 \dots \sigma_k\omega$, and the behavior of the machine is

the union of all such relations corresponding to all strings given by the control FSA, and only those strings.

If the base X is empty, it has no relations, so the only FSA that can act on it is the null graph (no states and no transitions). This is called the NULL machine. If X is a singleton, the only relations it can have are the identity 1 and the empty relation 0, which combine in the expected manner: $0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0, 1 \cdot 1 = 1$. Note that the identity relation 1 corresponds to the empty string λ . Since $1^n = 1$, the behavior of the machine can only take four forms, depending on whether it contains 0, 1, both, or neither, the last case being indistinguishable from the NULL machine over any size of base. If the behavior is given by the empty string alone, we will call the machine 1 with the usual abuse of notation, independent of the size of the base set. If the behavior is given by the empty relation alone, we will call the machine 0, again independent of the size of the base set. Slightly more complex is the machine that contains both 0 and 1, which is rightly thought of as the *union* of 0 and 1, giving us the first example of an operation on machines, a subject we will turn to in Section 4.4.

To fix the notation, in Table 4.4 we present the multiplication table of the semigroup R_2 that contains all relations over two elements (for ease of typesetting, the rows and columns corresponding to 0 and 1 are omitted). The remaining elements are denoted $a, b, d, u, p, q, n, p', q', a', b', d', u', t$ – the prime is also used to denote an involution over the 16 elements which is *not* a semigroup homomorphism (but does satisfy $x'' = x$). Under this mapping, $0' = t$ and $1' = n$; the rest follows from the naming conventions.

	a	b	d	u	p	q	n	p'	q'	a'	b'	d'	u'	t
a	a	0	d	0	a	q	d	d	0	d	q	a	q	q
b	0	b	0	u	u	0	u	b	q'	q'	u	q'	b	q'
d	0	d	0	a	a	0	a	d	q	q	a	q	d	q
u	u	0	b	0	u	q'	b	b	0	b	q'	u	q'	q'
p	p	0	p'	0	p	t	p'	p'	0	p'	t	p	t	t
q	a	d	d	a	a	q	q	d	q	q	q	q	q	q
n	u	d	b	a	p	q'	1	p'	q	u'	d'	b'	a'	t
p'	0	p'	0	p	p	0	p	p'	t	t	p	t	p'	t
q'	u	b	b	u	u	q'	q'	b	q'	q'	q'	q'	q'	q'
a'	u	p'	b	p	p	q'	d'	p'	t	t	d'	t	a'	t
b'	p	d	p'	a	p	t	u'	p'	q	u'	t	b'	t	t
d'	p	b	p'	u	p	t	a'	p'	q'	a'	t	d'	t	t
u'	a	p'	d	p	p	q	b'	p'	t	t	b'	t	u'	t
t	p	p'	p'	p	p	t	t	p'	t	t	t	t	t	t

Table 4.4. Multiplication in R_2

To specify an arbitrary machine over a two-element base we need to select an *alphabet* Σ , a mapping $M : \Sigma \rightarrow \{0, 1, a, \dots, t\}$, an FSA that generates some language over (the semigroup closure of) this alphabet, and input and output mappings α and ω . Because any string of alphabetic letters reduces to a single element according to the semigroup

multiplication, the actual behavior of the machine is given by selecting one of the 2^{16} subsets of the alphabet $\{0, 1, a, \dots, t\}$. Therefore, over a two-element base there can be no more than 65,536, and in general over an n -element base no more than 2^{n^2} non-isomorphic machines (ignoring input and output mappings), since over n elements there will be n^2 ordered pairs and thus 2^{n^2} relations.

While in principle the number of nonisomorphic machines could grow faster than exponentially in n , for our purposes the cardinality of the base can be limited to three, so the largest machine we need to countenance will have its alphabet size limited to 512. This is still very large, but the upper bound is very crude in that not all conceivable relations over three elements will actually be used, and in Chapter 5 we will discuss principled methods for deriving further bounds on these.

Exercise^o 4.14 Are FSA special cases of machines? Why or why not?

Exercise* 4.15 Are FSTs special cases of machines? Why or why not?

In contrast to the outer syntax discussed in Section 4.2, the *inner* syntax of the machine is simply given by stipulation, as a relational monoid Φ of some set X . There may be some ‘inner language’ for which X can be the state set of the defining automaton, but we are given both more and less than this. We are given more, since we are given the full congruence while the automata states correspond just to right congruence, and we are given less, since the syntactic monoid does not uniquely determine the language it came from. First and foremost, any bijection of an alphabet Σ to some other alphabet Γ will change the language but will leave the syntactic monoid intact up to isomorphism, so knowing the monoid offers no clues about the ‘true names’ of things. Second, the syntactic monoid only conveys information about the semiautomaton, often leaving the precise choice of initial and final states up for grabs. Yet this skeletal apparatus is sufficient to deal with the central issues of inner syntax, slot filling and argument sharing.

Example 4.3 Consider polynomials in several variables x, y, z, w, \dots . In addition to the usual arithmetic operations we have several functional operations, most importantly *substitution*. For example, by substituting $y + z$ in place of x in $wx + wx^2$ we obtain $wy + wz + wy^2 + wz^2 + 2wyz$. We can even substitute $y + z$ in place of w in the result to obtain $(y + z)^2 + (y + z)^3$, and actually the order of these substitutions is immaterial. Every time we substitute a polynomial having variables v_1, \dots, v_k in place of w_0 in some polynomial having variables w_0, \dots, w_r , we obtain a polynomial having variables $v_1, \dots, v_k, w_1, \dots, w_r$. It is only by substituting elements of the underlying ring that we can actually decrease the number of variables.

One way to look at this is through the automorphism group of these polynomials. By substituting x for z and y for w we can see that the polynomials $x + y^2$ and $z + w^2$ are, as long as these variables are not present elsewhere, exactly the same.

4.4 Graph and machine operations

Given two or three machines, we can imagine many operations that could be used to define new ones. In this book we make the bold claim that everything in semantics is

machines: words are machines, sentences are machines, model structures are machines, parsing, inferencing, generating, and all other tasks must be done by machines. To make good on this claim, we must make sure that even the operations we define on machines must be doable by machines. Thus, it is not enough for us to say that such and such semantic phenomenon corresponds, say, to a substitution of one machine in another, we must be able to show how the data structure corresponding to the output machine can be obtained *by machine means* from the data structures corresponding to the input machines. It is at this point that the lack of deeper mathematical apparatus in the prolepsis becomes useful. As we argued in Chapter 3, all we need are FSTs, their valuations, and some kind of implicational primitive \Rightarrow we could translate as ‘normally implies’.

As far as representations are concerned, machines could be dispensed with in favor of hypergraphs (and, as we shall see, hypergraphs of a rather limited kind), but these are static data structures, in themselves incapable of computation, even of the primitive kind of computation that FSA and FSTs perform. Here we will introduce context-free hypergraph grammars in analogy with context-free string rewriting, but our goal with these will be more limited, aiming at describing the form of the permissible structures, as opposed to the actual process of obtaining them. Before we begin discussing how computations are to be handled in the machine world, let us review some of the basic definitions and facts concerning FSTs.

Definition 4.9 A finite state transducer (FST), also known as a **Mealy machine**, is given by an *input alphabet* Σ , an *output alphabet* Γ , a state space S , and a *transition relation* $T \subset S \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \times S$.



In effect, an FST operates on pairs of input and output symbols the same way as a semiatomaton operates on unique symbols, and indeed it is customary to extend the definition with initial and accepting subsets of states. In Mealy machines, the convention is to decouple the input and the output by saying that upon receiving the input the Mealy machine moves to another state and subsequently produces the output (but this output can be dependent on the input symbol the machine last consumed).

The key part of the definition, which renders the exact microsynchrony of the input and output irrelevant, is the ability of FSTs to take λ moves. FSTs are inherently non-deterministic in that, in any given state, we can permit the machine to read no input (or, what is the same, read the empty string λ) but nevertheless move to another state and/or output further symbols. If T contains no tuples of the form $\langle s_1, \lambda, \gamma, s_2 \rangle$ we say it has no λ input, and further, if it has no two tuples equal on the first and second coordinates we say it is *deterministic*.

For any assignment of initial and accepting states, an FST naturally gives rise to a relation $R \subset \Sigma^* \times \Gamma^*$: we say that two strings $\sigma_1 \dots \sigma_k$ and $\gamma_1 \dots \gamma_l$ of not necessarily equal length are related iff there is a sequence of states s_0, \dots, s_r starting in an initial and ending in an accepting state such that both $\sigma_1 \dots \sigma_k$ and $\gamma_1 \dots \gamma_l$ can be parsed into r parts l_i and r_i , respectively (with interpolation of λ s as needed), so that each quadruple $\langle s_i, l_i, r_i, s_{i+1} \rangle \in T$ for $0 \leq i < r$. By definition, FSTs offer a means of characterizing certain string relations by a finite amount of data, and deterministic FSTs characterize

(partial) string functions. Yet there are string relations, even functions, which have a finite characterization but no FST representation. Consider the *square* relation over the one-letter alphabet $\{a\}$ given by all pairs of the form $\langle a^n, a^{n^2} \rangle$, and only those pairs, for $n \in \mathbb{N}$.

Exercise[→] 4.16 Define a Turing machine that returns for every input string a^n the string b if $s = \sqrt{n}$ is not an integer, and returns a^s if it is. Is this Turing machine linear bounded (requiring no more than a constant times the tape length of the input)?

String relations computable by FSTs are called *rational* or *regular*. It is important to know that these differ significantly from regular expressions (or rational sets) over strings, by not being closed under complementation or intersection. Consider a transducer T with one non-sink state and a loop which reads a and writes b , thus accepting the relation $\{\langle a^n, b^n \rangle : n \in \mathbb{N}\}$. As a moment's thought will show, the set of strings $a^n : b^n$ (where \cdot is some center marker) is not definable by an FSA.

Exercise[→] 4.17 Construct an FST defining the relation $\{\langle a^n, b^n c^* \rangle : n \in \mathbb{N}\}$ and an FST defining the relation $\{\langle a^n, b^* c^n \rangle : n \in \mathbb{N}\}$. What is the intersection of these two relations? Is it FST-definable?

Theorem 4.1 (Eilenberg) Given an FST $\langle \Sigma, \Gamma, S, T \rangle$ computing the regular relation $R \subset \Sigma^* \times \Gamma^*$, there exists a machine computing the same relation.

Proof We make the following choices: both the alphabet Σ' of the machine and the base set X are defined as $\Gamma_\Lambda \times \Sigma_\Lambda$, where Γ_Λ is the discrete direct sum of Γ with a special symbol Λ that we use to encode the empty string λ , and similarly for Σ_Λ (we need to make sure that Γ and Σ are disjoint, including the two Λ symbols, but we will not burden the notation with this). The image of $\langle \gamma, \sigma \rangle$ under the machine mapping M is given by $\langle \langle \lambda, \sigma \rangle, \langle \gamma, \lambda \rangle \rangle$. We define the input mapping $\alpha : \Sigma \rightarrow \Gamma_\Lambda \times \Sigma_\Lambda$ by $\sigma\alpha = \langle \lambda, \sigma \rangle$ and the output mapping $\omega : \Gamma_\Lambda \times \Sigma_\Lambda \rightarrow \Gamma$ by $\langle \gamma, \lambda \rangle\omega = \gamma$, and $\langle \gamma, \sigma \rangle\omega$ is undefined for $\sigma \neq \lambda$. The transition table T' of the machine will be based on the transition table T of the FST: if $\langle s, \sigma, \gamma, t \rangle \in T$, we put the transition $\langle s, \langle \gamma, \sigma \rangle, t \rangle$ in T' ■

This theorem is considerably weaker than the one actually proven by Eilenberg (1974: Section 10.3), who considers rational relations over arbitrary monoids, not just over free ones. Yet this is all that we need to demonstrate that FSTs, which we argued in Chapter 3 were proleptic, are covered by the theory.

Comp



In the machine-based system developed here, hypergraphs appear only in the base of the machines, but we can also devise a system based entirely on hypergraphs. For this we need to generalize to hypergraphs the well-known [context-free grammar \(CFG\)](#) mechanism. The reader already familiar with string CFGs (we will provide a formal definition in Section 5.1) will no doubt be somewhat surprised by the definition of hyperedge replacement, which makes a hyperedge e replaceable by some hypergraph H only if their types match.

Definition 4.10 For a hyperedge e in a hypergraph H and replacement hypergraph B , the hypergraph $H[e/B]$ (read: H with B substituted for e) is formed by deleting e from

H except for its attachment nodes $\text{att}(e)$ and adding in B by fusing the external nodes $\text{ext}(B)$ to these in order; $\text{ext}(H)$ remains unchanged.

It is then up to us how much of what we would ordinarily consider ‘context’ is encoded in the att/ext nodes. Consider a polygon-shaped hypergraph H with k vertices, all external, and hyperedges e_1, e_2, \dots, e_k each containing all these vertices as attachment nodes, all with cyclic ordering mod k , starting at $1, 2, \dots, k$.

Exercise^o 4.18 With rewriting rules $e_i \rightarrow e_{i+1}$ ($i < k$) and $e_k \rightarrow e_1$, let us first use the first rule and form $H[e_1/e_2]$. Is this isomorphic to H ? Why or why not? What, if anything, is changed by adding an extra vertex v_0 and a classical edge that runs from v_0 to v_1 ?

Exercise[→] 4.19 A *clock* mod k is a rewriting system acting on some data structure such that the current state of the system uniquely determines, mod k , the number of steps taken to get there from the starting state. A *semi-clock* will reach each of its k states exactly once, a full clock infinitely many times. Can you design a (context-free, deterministic) string/hypergraph rewriting system that acts as a (semi)clock? Can you base your design on FSA, FSTs, or machines?

Exercise[→] 4.20 Given some (semi)clocks mod p, q, r , where p, q, r are pairwise relative primes, design a (semi)clock mod pqr .

4.5 Lexemes

Here we introduce a special class of machines, called *lexemes*, which are intended as models of dictionary entries. How these related to the lexemes of lexicography is a question we defer to Chapter 6. While the definition is not hard (lexemes will be machines over one-, two-, or three-element directly linked base sets, the elements themselves being lists of pointers to other objects, typically machines), we will need quite a bit of discussion to motivate it, especially for readers not familiar with modern lexicography and knowledge representation. The collection of lexemes for any given language will be called the *lexicon* of the language, and we assume the reader to have at least passing familiarity with (monolingual) *dictionaries* produced by (teams of) professional lexicographers.

In algebra, we pay little attention to how we name things: the same construction, the cyclic group over 7 elements, could be called C_7 or Z_7 ; the same theorem could be named after Perron or Frobenius; and so on. In sharp contrast to this, linguistic objects already come equipped with *forms*, both written and spoken; in fact, it is part of the definition of the [linguistic sign](#) that it is composed of a form and a meaning. Further, it has been argued at least since [Plato’s Cratylus](#) that the connection between the form and the meaning is a matter of convention; that which we call a rose would by any other name smell as sweet. That the form cannot be derived from the meaning is evident from the fact that different languages uses different forms to express the same meaning. In the other direction, many aspects of meaning can be derived from an analysis of





the form: for example, *tendovaginitis* is an inflammation (-itis) of the sheath (vagina) surrounding the tendon. However, there are minimal (atomic) signs, called **morphemes** in linguistics, which cannot be decomposed further, and for these the relation between form and meaning is set entirely by convention (with the possible exception of **onomatopoeic** words).

In $\langle \text{form, meaning} \rangle$ pairs, the atomicity of one side does not imply the atomicity of the other. Consider *quicksilver*, composed of two forms, *quick* and *silver*. The compound form refers to a single unique element, mercury, which is chemically just as unanalyzable as silver, named by a single form. As with all adjective–noun compounds of this type, the name suggests a naive theory of mercury being a particular (quick, i.e. lively; cf. ‘the quick and the dead’) kind of silver, a theory that did not survive the transition from alchemy to chemistry. For the converse, consider *brass*, an atomic form defining a compound entity. Here we will be less concerned with the cases where it is the scientific analysis of an object that indicates its complex nature, concentrating instead on the cases where the linguistic analysis does this. Thus we will say that *quicksilver* is bimorphemic even though it denotes an atomic (chemically unanalyzable) object, and *brass* is monomorphemic, even though it denotes an alloy of copper and zinc.



Our first task is to show how the interpretation relation, generated by atomic $\langle \text{form, meaning} \rangle$ pairs, can be modeled by machines. This is not entirely trivial, since an ordered pair of machines is not necessarily a machine. In fact, the main case where it would be easy to interpret an ordered pair as a direct product is one where the bases of the two components are identical, an assumption that is not necessarily met by forms and meanings. Therefore, we follow a different route, and pack both members of the pair into the machine base X by the following technique. We start by assuming a basic set of forms (called **phonemes** in linguistics) and a basic set of meaning units, called *primitives*. Both sets are rather small – we need to countenance no more than a few dozen phonemes and a few thousand semantic primitives (the exact number will be discussed in Section 6.5). The invariance of the human vocal tract under changes in person, language, culture, and ethnicity results in a phonemic inventory that is entirely universal in the sense that all phonemic inventories can be obtained as subsets of a single, **relatively small** set. Currently, our level of understanding of the human sense-making process is incomparably weaker than our understanding of the sound-making process, and we are nowhere near the definition of ‘sememes’. Therefore, our primitives are more analogous to a set of arbitrarily chosen basis vectors in a linear space than to a natural system of cardinal coordinates. In order to sidestep the issue of whether these basic elements are truly primitives, we will simply talk of *defining* elements and collect these in a set \mathbf{D} (a specific list is provided in Section 4.8).



Definition 4.11 The *definition graph* of a dictionary has nodes corresponding to all the words (and some bound morphemes; see Section 5.2), including those words that only appear on the right-hand sides of definitions (even if they are not present as headwords). An edge runs from w_i to w_j if w_j appears in the definition of w_i or, if w_i has no definition, we add a self-loop. (In Definition 5.6 on page 166, we will refine this by distin-

guishing whether a definiens refers to the subject, the object, or the entire definiendum, but for now this is sufficient.) We say that a subset of nodes D *directly defines* a subset G if all edges starting in G end in D , and we say D is *defining* in G if there are subsets of nodes D_1, D_2, \dots, D_k such that $D = D_1, G = D_k$, and D_i is directly defining in D_{i+1} . We say that a subset of graph nodes has the [uroboros property](#) if no arrows lead out of it. Any subset of nodes \mathbf{D} that is defining for the entire dictionary, and has the uroboros property, is a set of *defining elements*.



To make this clear, consider the Hungarian verbal stem *toj* and the derived *tojó* ‘hen’, *tojás* ‘egg’, and *tojni* ‘to lay an egg’. It is evident that eggs are what hens lay, hens are what lay eggs, and the laying of eggs is what hens do. In Hungarian, the interdependence of the definitions is made clear by the fact that all three forms are derived from the same stem by productive processes: *-ó* is a noun-forming deverbal suffix denoting the agent, *-ás* denotes the action or the result, and *-ni* is the infinitival suffix. But the same arbitrariness in the choice of primitives can be just as evident in less transparent examples, where the common stem is lacking: for example in the English *hen* and *egg* it is quite unclear which one is logically prior. Consider *prison* ‘place where inmates are kept by guards’, *guard* ‘person who keeps inmates in prison’, and *inmate* ‘person who is kept in prison by guards’. One could easily imagine a language where prison guards are called *keepers*, inmates *keepees*, and the prison itself a *keep*. The mere fact that in English the semantic relationship is not signaled by the structure of the words does not mean that it is not there – on the contrary, we consider it an accident of history, beyond the reach of explanatory theory, that the current (somewhat archaic) nominal sense of *keep*, ‘fortress’ is ‘fortified place to keep the enemy out’ rather than ‘to keep prisoners in’.

Altogether, lexemes will be defined by other lexemes, each obtaining its definition in terms of its position in the network. Of course, if all words and larger expressions obtain their definitions by the position they occupy in the system of other words and larger expressions, we are facing the problem of circularity. In fact, the first English dictionary, (Cawdrey, 1604) already defines *heathen* as *gentile* and *gentile* as *heathen*. The problem was noted by Leibniz (quoted in Wierzbicka (1985)):

Suppose I make you a gift of a large sum of money saying you can collect it from Titius; Titius sends you to Caius; and Caius, to Maeuius; if you continue to be sent like this from one person to another you will never receive anything.

One way out of this problem is to come up with a small list of primitives, and define everything else in terms of these. There have been many efforts in this direction (the early history of the subject is discussed in depth in Eco (1995)), but the modern efforts begin with Ogden’s Basic English (Ogden, 1944). The modern tradition of [knowledge representation](#) begins with the list of primitives introduced by Schank (1972), and a more linguistically inspired list has been developed by Wierzbicka and the NSM school (Goddard, 2002). Here we develop a more systematic approach that exploits preexisting lexicographic work, in particular dictionary definitions that are already restricted to a smaller wordlist such as the Longman Defining Vocabulary (LDV) or Ogden’s *Ba-*





[sic English](#) (BE). These already have the proven capability to define all other words in the *Longman Dictionary of Contemporary English* (LDOCE) or the [Simple English Wikipedia](#) at least for human readers, though not necessarily in sufficient detail and precision for reasoning by a machine.

Looking at these lists, the LDV is less than 2,700 words (actually, morphemes, since it includes prefixes and suffixes as well as free-standing words), and the original Ogden list is just 850 words, while the list in the Appendix (Section 4.8) has about 1,230 entries. This was obtained by building a definition graph (see Definition 5.6 on page 166) of the Collins COBUILD dictionary, and searching for a list of defining elements: we discuss the process in a more formal setting in Section 6.4.



Exercise^o 4.21 Pick a random word from the [Ogden list](#), and define it in terms of the LDV without consulting the [LDOCE](#). Conversely, pick a random word from the LDV and define it in BE, without consulting the [Simple English Wikipedia](#). The word *random* does not appear in either of these lists; try to define it in terms of one or the other.

The building blocks will be pointers to phonemes on the one hand and to semantic primitives on the other. It is rather tempting to think of these as [urelements](#), but we resist this temptation, noting that both phonemes and semantic primitives can be decomposed in various ways. The pointers will be grouped into sets we will call *partitions*, and the partitions together will form the directly linked part of the base set X of the machine. For reasons that will become clear shortly, we assume one of the partitions is distinguished. The distinguished element will be called the *head*. As we discuss in Section 5.1, machines with only one partition are few: we have NULL, 0, 1, and 0+1. Most of the machines we deal with will have two partitions, one for the form (the component phonemes and further phonological structure) and one for the meaning. The notation keys off the form: for convenience, we will use orthographic rather than phonological forms, and write these in typewriter font. For example, the *dog* is four-legged, animal, hairy, barks, bites, faithful, inferior; the *fox* is four-legged, animal, hairy, red, clever.

In addition to the two-partition machines, which we will actually call *unary* lexemes because they are unary when conceived as functions, we will also have a handful of irreducible three-partition machines, which we will call *binary* lexemes because they denote binary relations. These will be written in SMALL CAPS and infix style. In Chapter 6 we will discuss further how lexemes, as defined here, are capable of doing all the work that lexicographers expect from their lexemes; here we concentrate on their modes of combination.

We emphasize that our lexemes are intended as highly modularized knowledge containers suited for describing our knowledge of words, as opposed to our encyclopedic knowledge of the world, which involves a great deal of non-linguistic knowledge such as motor skills or perceptual inputs, for which we lack words entirely. Unary lexemes will correspond to most nouns, adjectives, verbs, and content words in general (including most transitive and higher-arity verbs as well), while binary lexemes will correspond to adpositions, case markers, and other function words, for example x AT y ‘ x is at location

y , x HAS y ‘ x possesses y ’, x CAUSE y , etc. In Section 4.6 we will see how variables can be eliminated from the system entirely; for now, we retain them for expository convenience.

Unlike unaries, which have a single list (partition) defining their meaning, binaries have two defining lists of properties, one pertaining to their first (superordinate, head) argument and another to their second (subordinate, dependent) argument of the lexeme. We illustrate this with the predicate HAS, which could be the model for verbs such as *owns*, *has*, *possesses*, *rules*, etc. The differences between John HAS Rover and Rover HAS John are best seen in the implications (defaults) associated with the superordinate (possessor) and subordinate (possessed) slots: the former is assumed to be independent of the latter, the latter is assumed to be dependent on the former, the former controls the latter (and not the other way around), the former can end the possession relationship unilaterally, the latter can not, etc. The list of definitional properties is thus partitioned in two: those that belong to the superordinate argument are collected together in the *head* partition, and those belonging to the subordinate argument are listed in the *dependent* partition.

The *selectional restrictions* discussed in Section 4.2 provide many pertinent examples: for example, the verb *elapse* selects for a temporal subject, a fact we encode by listing `temporal` in its subject partition. When we hear *A sekki elapsed* we know that `sekki IS_A temporal` even if we don’t know how long exactly. Similarly, when we hear *Wiles proved Fermat*, we must follow an inferential chain starting with the fact that *prove* selects for an object that is a `statement`, and conclude that here *Fermat* stands not for the person, [Pierre de Fermat](#), but for the eponymous [Fermat’s Last Theorem](#). This inferential mechanism, and other kinds of inferences sometimes collected together under the heading of [pragmatics](#), will be discussed further in Section 5.6.

The lexical entries in question may also include pointers to sensory data, such as biological, visual, or other extralinguistic knowledge about dogs and foxes. We assume some set **E** of external pointers (which may even be two-way in the sense that external sensory data may trigger access to lexical content) to handle these, but here **E** will not be used for any purpose other than delineating linguistic from non-linguistic concerns. How about the defining elements that we have collected together in **D**? These are no different; their definitions can refer to other lexemes that correspond to their essential properties. So definitions can invoke other definitions, but the circularity causes no foundational problems, in that each lexeme is defined, up to isomorphism, by the control FSA, the cardinality of the base X , and the relations on X that the FSA alphabet is mapped to. In particular, we cannot easily distinguish the `fox` and the `dog` lexemes without inspecting what is stored in the partitions.

Notation 4.3 If *expression* is a linguistic expression so written, the lexeme (machine) corresponding to it will be written *expression*. The form side of a \langle form, meaning \rangle tuple will be written form if we need to emphasize its phonological nature, and the meaning of unaries is simply given as an unordered (comma-separated) list of machines.



We will generally omit the set-theoretic braces surrounding these lists, and make little distinction between a machine and a pointer to the machine.

Following Quillian (1967), semantic networks are generally defined in terms of some distinguished links: IS_A to encode facts such as that dogs are animals, and ATTR to encode facts such as that they are hairy. Here neither the genus nor the attribution relation is encoded explicitly. Rather, everything that appears in the distinguished (head) partition is attributed (or predicated) directly. There are two ways to think about IS_A. In one conception, close to the classic AI tradition, IS_A is just a dedicated link type modeling the Aristotelian notion of *genus*. This is the approach taken for example, in WordNet, where IS_A links are called *hypernyms*. This is clearly advantageous for the dictionary writer, who just needs to put in the link and need not separately specify everything about an item that follows from this.

The second approach, followed here, sacrifices some of this modularity and ease of dictionary-writing for logical transparency. Here there is no dedicated IS_A link; the concept is *defined* by the containment of the essential properties. Elementary pieces of link-tracing logic, such as $A \text{ IS_A } B \wedge B \text{ IS_A } C \Rightarrow A \text{ IS_A } C$ or $A \text{ IS_A } B \wedge B \text{ HAS } C \Rightarrow A \text{ HAS } C$, follow without any stipulation if we adopt this definition, but the system becomes more redundant: instead of listing only essential properties of dogs, we need to list all the essential properties of the supercategories such as animals as well. Altogether, the use of IS_A links leads to better modularized knowledge bases, and for this reason we retain them as a presentation device, but without any special status: for us *dog IS_A animal* is just as valid as *dog IS_A hairy* and *dog IS_A barks*. From the KR perspective, the main point here is that there is no mixing of strict and default inheritance; in fact there no strict portion of the system (except possibly in the encyclopedic part, which need not concern us here).

If we know that animals are alive, then we know that donkeys are alive. If we know that being alive implies life functions such as growth, metabolism, and replication, this implication will again be inherited by animals and thus by mules as well. The encyclopedic knowledge that mules don't replicate has to be learned separately. Once acquired, this knowledge will override the default inheritance, but we are equally interested in the *naive* world-view where such knowledge has not yet been acquired. Only the naive lexical knowledge will be encoded by primitives directly: everything else must be given indirectly, by means of a pointer or set of pointers to encyclopedic knowledge. The most essential information that the lexicon has about *tennis* is that it is a *game*, all the world knowledge that we have about it, the court, the racket, the ball, the pert little skirts, and so forth, are stored in a non-lexical knowledge base. This is also clear from the evidence from word-formation: clearly *table tennis* is a kind of *tennis*, yet it requires no court, has a different racket, and ball, and so forth. The clear distinction between essential (lexical) and accidental (encyclopedic) knowledge has broad implications for the contemporary practice of knowledge representation, exemplified by systems like CyC (Lenat and Guha, 1990) or Mindpixel in that the current homogeneous knowledge bases need to be refactored, splitting out a small, lexical base that is entirely independent of

Comp



domain. Finding a set of defining words, for example as listed in Section 4.8, is just the first step of this process; we need to also consider how the semantics of these words is computed, a matter we return to in Section 5.8.

4.6 Inner syntax

The outer syntax of lexemes concerns itself with the description of combinatorial phenomena: for example, both *dog* and *fox* form the basis of more complex words such as *dogged* or *foxy*. There is a verb *outfox* ‘be more clever than’, while there is no **outdog* ‘be more faithful than’ even though we have *dogged* ‘faithful, persistent’. Linguistic morphology generally takes these problems on board, and uses dedicated machinery, such as diacritic features, to handle such cases. Here we will abstract away from many details of outer syntax (also known as *phenogrammar*), using the machinery only to handle the productive processes. The combinatorial properties of words will be handled by carefully crafting the control FSA of lexemes, a matter we defer to Chapter 5.

Inner syntax, also known as *tectogrammar*, concerns itself with the logical combination of elements. As an example, consider *the shooting of the hunter was terrible*. It is not clear whether it is the ability of the hunter to shoot that we find terrible or the fact that they got shot. In both cases, *shooting* is what linguists call a *nomen actionis*, an action noun, but in one case the hunter is a subject (agent) performing the action and in the other they are the object (target) of the shooting. For the moment, assume simply that SHOOT is a binary primitive with tectogrammar x SHOOT y , where x is the subject and y is the object. To simplify matters further, take *be_terrible* as a unary predicate: what we need to find is the distinction between *hunter SHOOT be_terrible* on the one hand and *SHOOT hunter be_terrible* on the other. What we need are the machine-language operations to place *hunter* in the head partition of SHOOT in one case and in the dependent partition in the other.

The mechanism to accomplish this may look a bit contrived at first blush, since it will involve not just one but two silent elements, NOM and ACC. There are many languages where these elements are overtly marked by the morphology as case affixes; we will see examples in Section 5.3. The fact that in English they are marked only by position (the nominative preceding and the accusative following the verb) is an accident we leave to phenogrammar. Instead of analyzing *hunter SHOOT* and *SHOOT hunter*, we will be looking at *hunter NOM shoot* and *hunter ACC shoot*, respectively. The salient point is that *shoot* is no longer assumed to be a two-place predicate that requires both an agent and a patient argument; the real two-place predicates are NOM and ACC. By following this route, *shoot* becomes an ordinary (unary) predicate, and as we shall see in Chapter 5, this predicate is distinguished from the nominal *shot* only by its outer syntax.

What does NOM mean? The key semantic contribution is the sense of agency, that whoever is the subject is the one contributing the agentive force behind the shooting. It is, really, only the agent we can hold morally responsible, as emphasized by the NRA



slogan *guns don't kill people, people kill people*. The point, quite independent of one's political stance on gun control, is that grammatically the NRA has it right; guns are *instruments* of shooting, not *agents*. We can say *the hunter shot the deer with a long-range gun (for no good purpose)*, but it is awkward and strange (the technical term is *infelicitous*, henceforth marked by #) to say *#the long-range gun shot the deer (for no good purpose)* as it is only agentive behavior that can be evaluated for purpose. In fact, we will reserve another binary primitive, *INS*, for the purpose of linking instrumental clauses to the main predicate. The meaning *shoot* *NOM* *hunter* implies that the hunter *caused* the shooting, the hunter *controlled* the shooting, the hunter *directed* the shooting, and to the extent agentive behavior implies rationality and forethought, that the hunter *planned* the shooting.

We will therefore analyze *NOM* as the unordered collection of these (unary and binary) primitives just as we analyzed *dog* as an unordered collection of unary primitives. The point is not so much the exact list of the meaning elements that appear in the definition as the fact that in all inferences the first argument of *NOM* will always be in the superordinate role, it will always be the causer, the controller, the director, and possibly the planner, and it will never be that which is caused, controlled, directed, or planned. A full discussion of the semantics of *NOM* is unnecessary at this point: we will simply say that unaries such as *causer*, *agent* are associated with the superordinate (head) partition, but *NOM* may also have this partition shared with the first partition of *HAS* *plan*, *HAS* *forethought*, etc. Actually, there is much more to be said about the semantics of the nominative, but here we are more interested in its syntax, its tectogrammar in particular.

The goal is to make sure that the NP marked by the nominative case, *NP.NOM* for short, gets to enjoy the properties that we have just discussed, such as being the causer/planner of the action. The way to achieve this is by taking *NOM* to be a specific two-place machine operation that takes two unary lexemes, in our example *hunter* and *shoot*, as inputs and produces a third machine. We already have an operation with the required signature, conjunction, but applying that would lead to *hunter, shoot*, roughly 'there was a hunter and there was shooting', an expression that leaves unexpressed the critical part, that the hunters are the agent. Applying *NOM* puts *hunter* in one partition x , and *shoot* in the other partition y , superordination of *hunter* to *shoot* is the key issue in tracing commonsensical implications like *the hunter shot the deer* \Rightarrow *the hunter is responsible for the death of the deer*.

Binary relations like the mathematical $>$ or the grammatical *NOM* have two slots, and the unaries are the fillers. We employ the actions of the machine to encode which slot gets filled by which element. If the two partitions are denoted h and d (head and dependent), we consider the relations $I = \{\langle h, h \rangle, \langle d, d \rangle\}$, $H = \{\langle h, h \rangle\}$, and $D = \{\langle d, d \rangle\}$ and \emptyset . In terms of relation composition we have $IH = HI = H$, $ID = DI = D$, and $HD = DH = \emptyset$. Recall Definition 4.4, which requires a control FSA and a mapping M from the alphabet of the machine to binary relations over the base. Any symbol of the alphabet that is neither the verb nor a nominatively marked element *NP.NOM* will

be mapped by M to I , meaning that it does not affect the slots. NP.NOM corresponds to D , and the verb to H . When the control FSA consumes the verb, the initial relation I is composed with H , keeping the head partition open, but closing off the dependent partition, and when it sees NP.NOM, the relation is composed with D , which closes down the head partition. The order of operations doesn't matter: the relations do the bookkeeping that tells us, at any given moment, which slot is already filled and which one is still empty.

Exercise^o 4.22 In mathematics, relations like \leq are normally written in infix order, with the larger element written to the right and the smaller to the left of the relation symbol. Formulate in an FSA the fact that the only syntactically correct ordering is a numeral followed by \leq followed by a numeral; orderings like *num num* \leq are disallowed. Is an expression like $8 \leq 2$ well-formed? Why or why not?

What we have described above is a very rudimentary slot-filling calculus, one that lacks many of the combinatory possibilities of λ -calculus or combinator calculus. Again, simplicity is seen as a virtue, especially when it comes to learnability.

Exercise^o 4.23 Is the same slot-filling intuition expressible in terms of hyperedge rewriting (Definition 4.5)? How many attachment nodes will unaries and binaries have?

Returning to natural language, with the accusative ACC the situation is the exact opposite; the operation again creates a two-partition machine, with `shoot` in the head partition and `NP . ACC` in the dependent partition. Here the information associated with the dependent partition is that it is the undergoer, so when we see, for example, *shooting him*, we know *he* is not the cause but the target of the shot. In this special case we can take advantage of the fact that English preserves a historical remnant of case in 3rd person singular male pronouns.

Instruments are coordinated with, rather than subordinated to, the predicate, so the INS operation produces from `shoot` and `gun` the compound predicate `shoot, gun`, which is a unary machine. This is in accordance with the fact that instruments can be promoted both to subjects (as in *the robber killed the victim*, *the gun killed the victim*) and to predicates (as in *the robber gunned down the victim*). An interesting case is provided by locatives, such as AT, which can never be promoted to subjects or verbs. There are some indications, at least in certain languages, that they should be subordinated to the predicate by *locative inversion* (Salzmann, 2004).

Exercise^o 4.24 Analyze the sentence *the hunter killed the deer with a long-range gun*.

Here we take a highly abstract view that distinguishes three components of the system that implements inner syntax: the actual *mechanism* used for implementing deep cases, the precise *inventory* of deep cases, and the linguistic *patterns* that link deep cases to surface cases. Let us discuss each in turn.

The mechanism individuates only two kinds of links that typically correspond to cases, '1' and '2'. These are assumed universal, together with '0' (attribution, IS_A), which, however, does not correspond to surface case. We handle these kind of links with dedicated partitions as described above. One could in principle admit further link types



‘3’ etc. using the Eilenberg machine apparatus with more partitions, but as we discussed in Section 4.3, the number of non-isomorphic machines grows as 2^{n^2} in the number of partitions, and it is essential for learnability to control the search space. Everything else, including the instrumental case (what Pāṇini calls *karāṇa*), is seen as a primitive binary element with its own ‘1’ and ‘2’. Thus, we distinguish Sanskrit *kuṭhāraḥ chinatti* ‘the axe.NOM cuts’ from *kuṭhāreṇa chinatti* ‘(he) cuts axe.INS’ just as in English, where the instrument is signaled by the preposition *with* rather than by a case suffix. This is shown diagrammatically in Fig. 4.7.

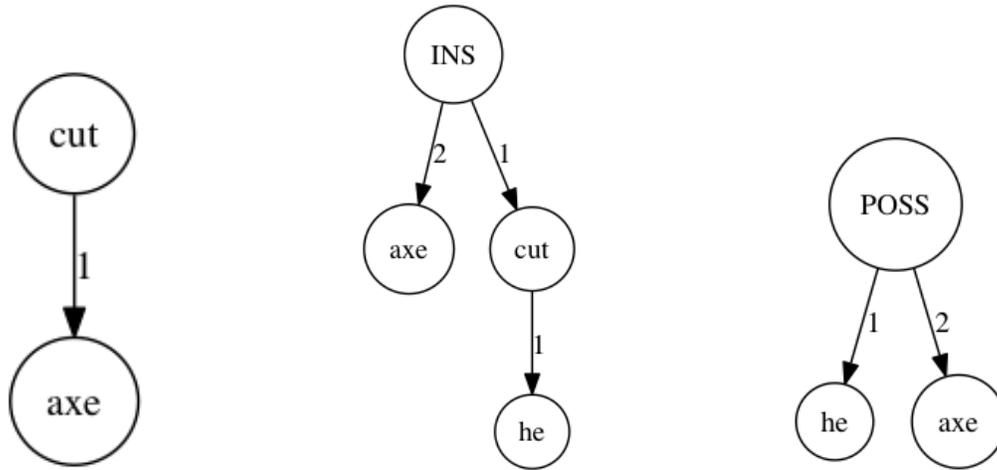


Fig. 4.7. *axe* as agent, instrument. and possession

The inventory of deep cases begins with AGT and PAT, which are directly encoded by the mechanism as ‘1’ and ‘2’ and on the surface as nominative and accusative in nominative–accusative languages. (In *ergative* languages the surface encoding is different, but we will not discuss these here in any detail.) In the system of Pāṇini, what we call AGT is called *kartṛ*, and what we call PAT is called the object, *karman*. But we use ‘1’ and ‘2’ in a more general sense than Pāṇini, as a technical means of linking arguments to all kinds of binary relations.

Fig. 4.7 shows that INS ‘*x* is an instrument for (doing) *y*’ is treated like any binary relation, with a ‘1’ link to the first argument and a ‘2’ to the other. In such situations, it doesn’t quite make sense to say that *y*, cutting, is the “subject” of the relation, and, from a grammatical standpoint, to call *x*, the axe, an “object” would also be dubious. Yet ‘1’ and ‘2’ are sufficient for the bookkeeping, in that it is always ‘2’ that bears the surface instrumental case, and when it does not, as in ‘the axe cuts’, we don’t consider it an instrument, but rather an agent.

Further deep cases include the three locatives FROM, TO, and AT. Of these, the first one may be familiar from Latin grammar as the *ablative* and from Sanskrit as *apādāna*, source. AT is the *locative* (really, *essive*) case, Pāṇini’s *adhikaraṇa*, while TO (goal)

roughly corresponds to his *saṃpradāna* or the Latin *dative*, at least in the locative sense of dative (the other senses will be considered shortly). There are two, somewhat more technical relations we need to consider: the deep case REL, typically linking elements within a single word, and the primitive relation POSS. In some analyses of Latin, the possession relation is considered a case, *genitive*, but here we follow the larger tradition that keeps this relation separate from cases proper, as it obtains between two nominals, the possessor ‘1’ and the possession ‘2’, rather than between a verb and a nominal (see Section 6.3 for how we distinguish verbal and nominal parts of speech). Other relations between nominals such as PART_OF or ELEMENT_OF are also conceivable, but they have little impact on grammar; their importance is in drawing inferences.

With these, our inventory of abstract cases is complete, and we may turn to the issue of the linguistic patterns that link deep cases to surface cases or other surface patterns (such as word order in English). Needless to say, this is one of the focal issues in grammatical theory, and here we can only scratch the surface. We will pay special attention to the *dative* case, first, because the temptation to handle this directly by the mechanism by use of yet another partition ‘3’ is hard to resist, and second, because it illustrates rather nicely the cross-linguistic differences one must take into account.

The dative has several uses. Clearly the most frequent is when it marks the recipient of a gift or promise, as in Hungarian *Marinak virágot adott Péter*, Mary.DAT flower.ACC give.PAST Peter.NOM, ‘Peter gave flowers to Mary’. For Sanskrit, this is sutra 2.3.13 of Pāṇini. For English the situation is more complex, as there is a phenomenon of *dative shift* exemplified by *Peter gave flowers to Mary* and *Peter gave Mary flowers*. Under the ‘hard to resist’ analysis both of these sentences, and even the nominal form *Peter’s giving of flowers to Mary*, mean the same thing, aside from the past tense.

Ling

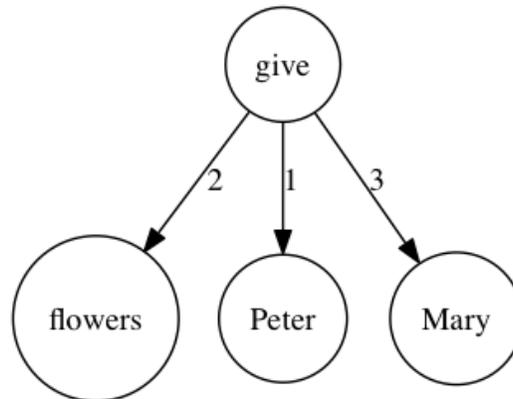


Fig. 4.8. Peter’s giving of flowers to Mary

Under the 41ang style of analysis, this is problematic because there are no verbs that have three arguments: *give* is defined as =AGT CAUSE [=DAT HAS =PAT]. The mechanism has to link up the agent ‘1’ of CAUSE with the NP in the nominative, the

agent ‘1’ of HAVE with the NP in the dative, and the patient ‘2’ of HAVE with the NP in the accusative, as in Fig. 4.9.

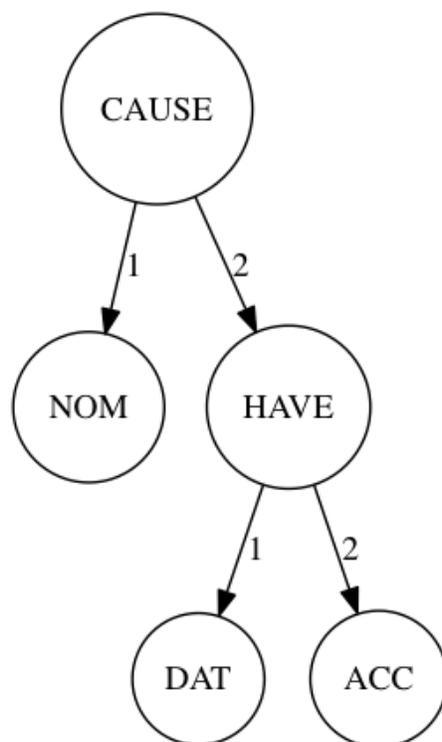


Fig. 4.9. *give*



What this means is that agent the causes the recipient to have the object, something that [generative semantics](#) would have expressed by stating that *give* means ‘cause to have’. Clearly, the dative shows aspects of both having (possession) and getting to (direction), and in language after language we find a dative possessive construction such as Hungarian *Marinak a virága*, Mary.DAT the flower.POSS, ‘the flower(s) Mary has’ or Latin *homini cum deo similitudo est*, man.DAT with God likeness is, ‘man has a likeness to God’. We also find purely directional datives, as in Hungarian *falnak megy*, wall.DAT walk.3SG, ‘he walks into a wall’.

Since the prototypical act of giving involves the object both moving from the donor to the recipient and coming into her possession, it is not at all surprising that the dative case ending has both a directional and a possessive sense. More mysterious, from the standpoint of unified semantics, are the patterns where the dative is an experiencer, as in *Marinak tetszik az ötlet*, Mary.DAT appeal.3SG idea, ‘The idea appeals to Mary; Mary finds the idea appealing’ or in *Péternek ez fáj*, Peter.DAT this.NOM pain.3SG, ‘This pains Peter’. Further, there are even more remote patterns such as *Mit csinálsz*

nekem?, what.ACC do.2SG I.DAT, ‘what the heck are you doing?’, where there is nothing received by the dative NP, it just concerns itself with the issue. These patterns again recur in language after language: the ‘concern’ is known as *dativus ethicus* in Latin. Finally, there are cases that seem impossible to subsume under any generalization; consider *Péternek el kell mennie*, Peter.DAT away must go.INF3SG, ‘Peter must leave’.

The number of such cases is so large, and so varied across languages, that to subsume all of them under a single deep dative (often called the *indirect object*) seems practically impossible. Stories can of course be told, of how the experiencers are recipients with the feelings/sensations just coming to them, or how the person concerned is really a beneficiary, but the predictive power of such stories is practically zero, for if it were otherwise, the same dative patterns would always occur independent of language. There are quite a few similarities, to be sure, but whether the explanation is to be sought in universal cognitive patterns or in etymological relatedness or cultural borrowing remains to be seen.

Be that as it may, we need a mechanism for expressing such patterns on a per-language basis, and `4lang` keys them to the main verb. Consider *appear*, which in Hungarian governs a double dative: *Marinak Péter betegnek tűnik*, Mary.DAT Peter.NOM sick.DAT appear.3SG, ‘John appears sick to Mary’. We analyze *appear* as *give impression of* or, more precisely, as *agent cause recipient to have impression of condition*.

Exercise → **4.25** Analyze the verbs *defend* (from/against), *equal*, *feed*, *prefer*, *protest*, *shoot* (*at*), and the adjective *full*.

4.7 Further reading

The definitions of semiautomata, automata, languages, and transducers given in Section 4.1 reflect the simplest (classical) case. In the modern theory, a great deal of attention is paid to the case when the language is defined over formal products of an arbitrary monoid, not just the free monoid, and when a simple yes/no decision on membership is replaced by valuation over an arbitrary semiring; see, for example, Kuich and Salomaa (1985). We offer a simple version in Section 5.8. Clocks are discussed in Kornai (2015).

In linguistics, the study of the syntactic congruence goes back to the structuralist school, in particular Chapter 16 of Bloomfield (1926) and Chapter 16 of Harris (1951). The mathematical theory begins with Rabin and Scott (1959). Pin (1997) offers an excellent survey of more modern developments. For a broader introduction to formal syntax, we recommend Chapters 1 and 2 of Kracht (2003) and Chapter 5 of Kornai (2008). These two run to 175 and 75 pages, respectively, and cover largely disjoint material, with Kracht concentrating more on the mathematical and Kornai more on the linguistic issues.

Our work continues the trend toward more formalized lexicon-building that started around the Longman Dictionary (Boguraev and Briscoe, 1989) and the Collins COBUILD dictionary (Fillmore and Atkins, 1994). The idea of separating tectogrammar from



phenogrammar goes back to Curry (1961) and is employed by a variety of modern systems (see Pollard (2006) and the references cited therein), of which we single out [Lexical Functional Grammar](#), whose notion of ‘a-structure’ (argument structure) is closest to the view of tectogrammar presented here, and has been argued for in the syntax literature on far richer data than we could even mention here (Goldberg, 1995).



For a less formal, but still highly relevant discussion of the nominative, accusative, and other cases, see Jakobson (1936), translated as Jakobson (1984). We contrasted our system of deep cases to Pāṇini’s system in such detail because *kāraḥas* are well understood and well worked out, and fit seamlessly into a comprehensive system of phenogrammar. For later developments, see Ostler (1979) and Smith (1996); for a broader survey, see Butt (2006). For an even better example of pronominal case preservation in a Germanic language, see Pullum (2015). For a more detailed discussion of how we can get by without ‘3’, see Kornai (2012).

4.8 Appendix: defining words

The following list, selected from the definition graph of the Longman Defining Vocabulary (which uses British spelling), has the uroboros property (see Section 6.4 for further discussion).

able about accept accident acid across act action activity actual add addition advertise affect Africa aft against age ago agree agreement aim air aircraft airforce alcohol all allow alone along although always America amount amuse an ancient and angle angry animal another answer any anyone appear appearance approve April area argument arm arms army around arrange arrangement arrive arrow art as Asia ask at atom attach attack attention attitude attract attractive August Australia authority available away baby back bad bag ball band bank bar base baseball bath bathroom be bean beat because become bed bee beer before begin behave behaviour behind belief believe belong below bend between bicycle big billiards bird bite bitter black blade blame block blood blow board boat body boil bone book bore both bottom bound bow bowl box branch brass brave bread break breathe breed bridge bright bring Britain brown brush bubble Buddhist build building burn bury bus business but button buy by cake call calm Cambridge can Canada car carbon card care careful carry case cat catch cattle cause cell center ceremony certain chance change character charge chemical cheque chess chicken child choose Christian church cigarette circle city class clay clean clear clever climb close cloth clothes cloud club coal coat coin cold collect college colour comb combine come comfort common company competition complete computer concern condition confidence connect consider consist contain container continue continuous control cook cool copper copy corn correct cost could country courage course court cover cream cricket crime criticize crop crush cup curve customer cut damage dance danger dark day dead deal December decide decision decorate decoration deep degree deliberate design destroy detail determine develop development die different difficult dig direct direction dirt dish disk distance divide do document dog dollar door down Dr draw dress drink drive driver drop drug

dry duty each eager earn earth easy eat edge effect effort egg Egypt electrical electricity electronic element elephant else embarrass emotion end energy engine England English enough enter equal equipment escape especial Europe even event exact example exchange excite exercise exist expect experience explain explode express expression extreme eye face fact fail fair fall family farm fast fasten fat fault feature February feel feeling female field fight fill film final find fine finger finish fire firm first fish fit five fix flat flesh floor flow flower fly fold follow food foot football for force form formal forward four frame France free friend frighten from front fruit full funeral fungus funny fur furniture further future gain game garden gas general gentle Germany get give glass go gold golf good goodbye goods government gradual grain grass gray great Greece green group grow guilty guitar gun hair hand handle hang happen happy hard harm have head healthy hear heat heavy hello help here hide high him his hit hold hole hollow holy home honest hope horn horse hospital hot house how hurt ice idea if ill imagine important impressive in include increase India industry influence information injure insect inside instead instrument intend interest into invite involve Ireland iron it jacket Japan Jesus jewellery Jewish jinks job join joke judge jump June just keep Kenya key kick kill kind king knight know knowledge Korea ladder land language large late laugh law layer lead leader learn leather leave left leg legal length lens less let letter level lid lie life light lightning like lime limit line linen lion liquid list listen literature live living London long look lose lot loud love low lower luck machine magazine main make male man many March mark marry mass material mathematics may meal mean meaning measure meat medical meet member mental mention message metal middle might milk mind Mississippi mix mixture modern monastery Monday money month moral more most mother mountain mouth move movement Mrs much Muhammad muscle music musical Muslim must nail name narrow natural near necessary neck need needle negative nervous -ness new newspaper next night no nobility noise nor north nose not note nothing notice noun now number nun oat object occasion o'clock October octopus of off offend offensive office officer official often oh oil old on one only onto open opinion opponent oppose or order ordinary organ organization organize other out outside over own package pain paint pair Pakistan pale paper parent Paris park parliament part participle particular party pass passage past pastry pattern pay penny people perform perhaps period permanent person petrol petroleum photograph phrase physical pick picture piece pig pipe place plain plan plane planet plant plastic plate play player pleasant please pleasure plural poem point Poland pole police polite political politics pool popular port Portugal position possess possible post pound powder power powerful practice prepare present preserve press price principle print prison private problem process produce product programme progress promise proper protect protection protest proton prove provide public pull punish purpose push put quality quantity queen question quick quiet race radio rain raise range rank rather raw reach react read ready real realize reason receive record red regular relate relationship relax religion religious remember remove repair report represent request respect responsible rest result return ride right ring rise risk river road rock romantic Rome roof room rope rough round row rub rubber rude rule run

Russia sad safe sail salad salt same sand Saturday say scale school science Scotland screen sea search season seat second secret see seed seem sell send sense sensible sentence separate series serious serve service set seven several sew sex shall shape share sharp sheet shelf shell ship shirt shock shoe shoot shop short show sick side sign signal silk silly similar simple sincere sing single sink sit situation six size skill skin sky sleep slide slight slippery slope slow small smell smoke smooth snow so social society soft soil soldier solid Somalia some someone something sometimes somewhere song sorry sound sour space speak special speed spell spend spirit spoil spoon sport spread square stand standard star start state statement station stay steady steal stem step stick sticky stiff stitch stock stomach stone stop store storey straight strange strength stretch string strong structure study stupid style subject substance succeed success such Sudan sudden suitable summer sun support sure surface surprise surround sweet swim system table tail take talk tall Tanzania taste tax teach team tear telephone television tell temperature tennis tense tent test than thank that the theatre their theirs them then these they thick thin thing think this though thought thread through throw Thursday tidy tie tight time tire title to together too tool tooth top total touch towards town track tradition train travel treat treatment tree trick trip trouble true try tube Tuesday turn twist two type typical Uganda UK under understand unit university unless until up upper upset Ural urine use useful U-shaped usual valuable value vegetable vehicle very video vine violent voice volcano vote waist Wales walk wall want war wash Washington waste watch water wave wax way weak weapon wear weather weave weight welcome well western wet whale wheel when where whether which while white who whole why wide wife will win wind window wine winter wire wish with within without woman wood wooden woods wool word work works world worry worth wound wrap write wrong year yellow yet you young your

Phenogrammar

Contents

5.1 Hierarchical structure	124
5.2 Morphology	126
5.3 Syntax	133
5.4 Dependencies	143
5.5 Representing knowledge and meaning	150
5.6 Thoughts in the head	153
5.7 Pragmatics	158
5.8 Valuation	165
5.9 Further reading	169

In modern linguistics, the idea that there is more to grammar than meets the eye is associated with Noam Chomsky, who has made the distinction between *surface* structure and *underlying* (also known as *deep*) structure a centerpiece of his theory of [transformational grammar](#). The general idea can be traced back at least to Heraclitus, who wrote (DK 54) “Latent structure is master of visible structure”, and it is to this broader explanatory method that we reach back to when we distinguish *phenogrammar* (from Greek *phainenin*, ‘show’) from *tectogrammar* (from Latin *tectus*, covered). In the system presented here, the division of labor between pheno- and tectogrammar will correspond to the distinction between the control and the base of machines.

In [5.1](#) we build up a rather standard picture of linguistic analysis in which morphemes are assembled into words, words are assembled into phrases, and phrases are used as the basic functional units in forming full sentences. While we explain in some detail *how* the system works, we cannot, within the bounds of this volume, reasonably explain *why* such a seemingly complex architecture is used, and must direct the interested reader to the broad range of introductory linguistics textbooks from Gleason ([1955](#)) to Fromkin, Rodman, and Hyams ([2003](#)) – suffice it to say that a great deal of the justification comes not from any single language or family of languages, but rather from the simultaneous study of all languages.

In [5.2](#) we summarize the basic ideas we will need from morphology to build a morphology–syntax interface based on lexical categories marked for inflectional distinctions. It would be rash to say that finding the lexical categories associated with words





(a task known in computational linguistics as [part of speech \(POS\) tagging](#)), or analyzing/generating inflection, are solved problems, but for many languages of great interest algorithms that perform these tasks are now [freely available](#), and the technologies to build new ones are not beyond the reach of the individual researcher. The same is not (yet) quite true of the analysis and synthesis of derived and compound forms, but as we shall see, we will only need the inflectional part of morphology to get a working semantic system.

In [5.3](#) we turn to syntax, primarily at the phrase level, but also introducing clauses and related units. We continue developing the formal syntax apparatus introduced in [4.2](#), with special emphasis on phenomena that are frequent in natural languages, in particular fixed-order obligatory complements, agreement, and constituent grouping.



In [5.4](#) we link the material from [5.1–5.3](#) to the computationally most relevant syntactic framework, [Universal Dependencies \(UD\)](#). This section is aimed primarily at the computational linguist already familiar with UD; others are advised to look first at Nivre et al. (2016) and the UD website linked above. Here we will also provide the first, informal introduction to the `4lang` theory of semantic representation, which will be more formally defined and discussed in greater depth in [Chapter 6](#).

In [5.5](#) we begin to introduce our theory of semantic representation, which serves both as a means of capturing verbal knowledge in general and as a means of describing the meaning of particular utterances, and in [5.6](#) we begin to make good on a promise we made in [3.4](#) about how people model what thoughts others and themselves may have in their heads. In [5.7](#) we discuss several phenomena, ranging from cooccurrence restrictions to ‘incomplete’ utterances, that are hard to explain without invoking some form of extralinguistic knowledge. Finally, in [5.8](#) we discuss how we can focus attention on smaller, dynamically built parts, meaning representations, within a larger static structure, the entire network of lexical entries.

5.1 Hierarchical structure

The everyday concept of speech and language embodies the observational notion of the *utterance* as a complete unit of talk, bounded by the speaker’s silence, and the (perhaps more normative) assumption that utterances are composed of sentences, sentences are composed of words, and words are composed of speech sounds. We say that the model is somewhat normative because we observe utterances composed of incomplete, unfinished, or otherwise broken sentences quite often, yet it makes sense to say that the idea of parsing utterances into complete sentences is at least honored in the breach.



In [Section 4.5](#) we have already introduced the minimal meaningful sub-word unit, the morpheme, and here we will introduce a super-word unit called the [phrase](#).

In general, phrases are the largest multiword stretches of speech (or writing) that function as single words. A typical, and for our purposes quite essential example is the *noun phrase* (abbreviated as NP) which serves as a single noun (abbreviated as N). For example, in the context `__ could not repeat last year’s success` the NP *The club that*

is more than a club can be replaced by the noun *Barça*. Since a similar replacement is possible in every context, the longer string and the single word are congruent. Definition 4.8 (page 99) has already given us the notion of lexical categories as congruence classes of words, and we leave open the issue of what other categories, besides nouns, we may encounter in the study of a language (one specific proposal will be discussed in detail in Section 5.4).

The reader may continue to think of the categories encountered in grade school such as *verb*, *preposition*, *adjective*, *adverb*, *pronoun*, and so forth, but should not necessarily assume that the exact same set of categories are available in every language. In fact, counterexamples abound, for example what English does by prepositions (so called because they precede the noun) as in *under the lamp*, Hungarian will do with postpositions as in *a lámpa alatt*. Once a language L is fixed, its category system C_L can be computed, and we may collect together all strings $a_1 \dots a_k$ congruent with some $X \in C_L$ and call these XPs. For example, in English we have the phrasal category of PPs, typically composed of a preposition P and a full NP, as in *John looked behind* (a ‘bare’ preposition) and *John looked behind the ancient monochrome TV set* (a ‘full’ PP).

Exercise° 5.1 In the above example, the NP and the N are not just congruent, but synonymous: one can replace the other not just without loss of grammaticality but also without change of meaning, what Leibniz called *salva veritate*. Provide examples of some phrase XP and word X where the replacement preserves grammaticality but does not preserve meaning. Provide examples of words or phrases that are largely synonymous but nevertheless fail to be congruent.



In understanding the category and the phrase system of some language L , a key observation concerns the repeatability of substitutions. When we say that a P such as *behind* can be replaced by a full PP such as *behind Bill* or *behind the TV set* it does not follow that the substitution can be repeated. On the contrary, the result of such a repetition is generally ungrammatical, as in **behind the TV set Bill*. The first formal calculus of substitution, that of Harris (1951), used the equals sign and a superscript mechanism to capture this fact, increasing the superscript to denote that the construction had reached a new (higher) level as in $P^1 = P^0N^1$. An important technical innovation, due to Chomsky (1956), was to replace equality by containment and write $P^1 \rightarrow P^0N^1$, meaning that strings with category P^1 can be formed by strings P^0N^1 but need not be so formed, because an alternative expansion or *rewrite rule* $P^1 \rightarrow P^0$ is also available.



Definition 5.1 A context-free grammar (CFG) is given by a set of terminals Σ , a set of nonterminals V , a start symbol $S \in V$, and a set of rewrite rules of the form $N \rightarrow R$, where N is a nonterminal and R is a regular expression composed of terminals and nonterminals by means of the regexp operations concatenation, union, and Kleene star $*$.

Exercise° 5.2 The above definition is often known as that of an ‘extended’ CFG, in contrast to ‘ordinary’ CFGs where the right-hand side of a rule must be a single string, rather than an arbitrary regular expression, over $\Sigma \cup V$. Prove that if a language L can be defined by an extended CFG, it can also be defined by an ordinary CFG.

Notation 5.1 In an influential paper, Chomsky (1970) used $\bar{X}, \bar{\bar{X}}, \dots$ instead of a superscript as in X^1, X^2, \dots which gave rise to the name *X-bar theory*; see also Jackendoff (1977). However, while the name stuck, the notation did not, and most linguists today use primes, as in X', X'', \dots instead of bars, reserving XP for the maximal X-like construct. Superscripts are best avoided because they conflict with the power notation familiar from formal language theory, where X^2 is the same as XX , X^3 is XXX etc. Another point where notation is somewhat unsettled concerns the treatment of optional elements.

Definition 5.2 For a string $a_1a_2 \dots a_n$ of some language L we say that the substring a_k is *optional* if the string $a_1 \dots a_{k-1}a_{k+1} \dots a_n$ is also in L .

Notation 5.2 Optional parts of strings are denoted by parentheses () in linguistics and brackets [] in computer science. Here we follow the former convention, except when omitting or emending parts of direct quotations, where the brackets are kept to mark the change. In particular, in regular expressions, where () is commonly used for grouping, we will use [] instead.

For example, in *John looked behind the old TV set* the word *old* is optional since *John looked behind the TV set* is also grammatical, even though it does not mean the exact same thing. This is conventionally denoted by parenthesizing the optional element, i.e. *John looked behind the (old) TV set*, a notation that is used in the right-hand side of rewrite rules as well: for example, $PP \rightarrow P(NP)$ abbreviates the disjunction of two rules $PP \rightarrow P$ and $PP \rightarrow P NP$.

Exercise^o 5.3 Using the extended CFG formalism, provide a set of rewrite rules generating the English numerals *one, two, \dots, nine hundred and ninety nine million nine hundred and ninety nine thousand nine hundred and ninety nine*.

Words, defined phonologically as minimal units between potential pauses, occupy an interesting position between morphemes and phrases, in part because some languages make them very small while others make them very large. A good example of the former type (known as *isolating* or *analytic* languages) is Vietnamese, where words are typically composed of a single morpheme, and phrases are therefore typically composed of many words. At the other, *synthetic* or *polysynthetic* extreme we find languages such as Finnish or *Nahuatl* which have many morphemes per word and consequently only a few (in the limiting case, only one) words per phrase or even per sentence. One version of structuralist analysis (Harris, 1946) conceives of sentences as being directly composed of morphemes, but this is something of an aberration, in that both classical grammar (starting with Pāṇini in around 500 BCE) and contemporary generative grammar, starting with Aronoff (1974), and Anderson (1982), admit an intermediary level of words composed of morphemes and serving as sentential building blocks. In fact, many structuralist grammarians, from Bloomfield (1926) to Hockett (1954), also use the word as an irreducible analytic category, and modern research such as that summarized in Aronoff (2007) leaves little doubt that words or, more precisely, *lexemes*, are indispensable for stating rules of grammar.



5.2 Morphology

Literally, *morphology* just means ‘the study of shape’ (from Greek *μορφή*, form, shape). In linguistics, where words play a central role, morphology means the study of word shapes. The pivotal position of words between morphemes and phrases justifies dividing the entire study of linguistic form into two parts: *morphotactics* studies how words are built from morphemes, and *syntactics*, more commonly called *syntax*, studies how phrases and larger structures are built from words. We begin with morphotactics, where the morphemes are divided in two broad classes: *free* morphemes such as *sevenm* which can stand ‘freely’ as words on their own, and *bound* morphemes such as *-th*, which can appear in words only in the company of other (free or bound) morphemes.

It should be emphasized at the outset that in a word like *seventh* there is no sense in which the free component contributes more (or less) to the meaning of the word than the bound component. The difference is purely a matter of tactics, as can be seen from the fact that different languages may express the same meaning by free or bound morphemes. For example, the idea of definiteness is expressed in English by the article *the*, at least orthographically a free form, while in Romanian (for masculine nouns) it is expressed by the suffix *-ul*. That said, the traditional distinction between **content words** and **function words** is one where words with the same content generally have unmarked free forms across languages, while function words are often best translated by bound morphemes in other languages. Also, diachronically, content morphemes tend to stay free, while function morphemes often migrate between bound (suffixal), semi-free (clitic), and free forms. For this reason, neither pure morphotactics nor pure syntax is appropriate for fully defining notions, such as ‘lexical category’ (part of speech), that lie at the interface between the two, a problem we shall return to in Section 6.2.

The matter is further complicated by the existence of *roots*, which are bound but contentful morphemes arising in the process of analyzing words into their smallest constituent parts. Consider Sanskrit *çak* Whitney (1885):169–170 glossed by Whitney as ‘be able’. In fact, *çak* in no way connotes any aspect of being; it is a peculiarity of English that ability is expressed by a copulative form *is able to*. In general, roots fail to participate in the category system of words; their tactics is determined purely at the morphological level.

Exercise^o 5.4 There are four possibilities: a function morpheme can be bound or free, and a content morpheme can also be bound or free. Try to collect examples of all four from a language that you are familiar with.

One final distinction we need to make is between *inflectional* and *derivational* affixes. To quote Anderson (2003):

[] inflectional categories are those that provide information about grammatical structure (such as the fact that a noun in the accusative is likely to be a direct object), or which are referred to by a grammatical rule operating across words (such as the agreement of verbs with their subjects). The validity of other correlates with inflectional status, then, follows not from the nature of the categories



themselves, but rather from the existence of grammatical rules in particular languages that refer to them, and to the freedom with which items of particular word classes can appear in positions where they can serve as the targets of such rules.

Derivational affixes, on the other hand, are involved in the creation of new words from roots or stems. In terms of tactics, derived words tend to be distributionally equivalent to nonderived ones both morphologically (within word tactics) and syntactically (across word tactics), while inflected words generally occupy a unique position that is shared only by similarly inflected forms. The totality of inflected forms that can be created from a stem is known as the *paradigm* of the stem, and the structural positions within a paradigm are known as *paradigmatic slots*. As discovered independently by Pāṇini (fl. 500 BCE) for Sanskrit, the grammarians of the Alexandria school (third and second centuries BCE) for Greek, and subsequently by the Latin, Arabic, and Hebrew grammarians, the system of slots is so strong that it is preserved in meaning and function even when the forms that fill the slots are irregular (as in English *ring*, *rang*, *rung* instead of *ring*, **ringed*, **ringed*).

Here and in what follows we simplify matters by assuming that morphemes are put together concatenatively, while in fact non-concatenative effects are prominent in many languages, of which the various Semitic languages are the best known examples. Consider Arabic *kataba* ‘he wrote’, *kutiba* ‘it was written’, *katabnā* ‘we wrote’, *naktubu* ‘we write’, *yuktibu* ‘he dictates’, *maktab* ‘office’, etc. The subject, known variously as *templatic*, *root-and-pattern*, or *non-concatenative* morphology, will not be discussed in detail here, but we note that such effects can also be found in languages such as English whose morphology is dominantly concatenative; cf. *sing*, *sang*, *sung*, *song*. We also simplify matters by ignoring *clitics* which are prosodically bound to adjacent words, but not necessarily to the ones they modify. With these simplifications, we expect morphotactics to be describable by a few simple tactical rules:



$$\text{Stem} \rightarrow \text{Root DerivAffix}^* \quad (5.1)$$

$$\text{Stem} \rightarrow \text{Stem DerivAffix}^* \quad (5.2)$$

$$\text{Stem} \rightarrow \text{Stem Stem} \quad (5.3)$$

$$\text{Word} \rightarrow \text{Stem InflAffix}^* \quad (5.4)$$

Needless to say, to define the morphotactics of any given language we need considerably more. First, we need to list the entries that belong in the preterminal categories. Such lists can never be complete for *open* classes such as Word, Stem, or Root, since new words enter the language all the time and, much less perceptibly but just as steadily, old ones often fall into disuse. On the whole, words from other languages tend to be borrowed as full forms, and get reanalyzed as stems only later. In languages where roots play a critical role, such as Hebrew, the process of assimilating loans to the native system of tactics can go as far as endowing a loanword which originally had no non-concatenative

structure such as *to telephone* with a templatic root pattern *tlpn*. On the other hand, the *closed* classes of inflectional and derivational affixes can be exhaustively listed at any given point in the development of a language, not just because their rate of change is considerably slower, but also because there are many fewer closed-class elements, on the order of $10^2 - 10^3$, than open-class, of which there are $10^4 - 10^6$.

Second, we need to provide more detailed tactic information than ‘affix’, a term that glosses over linear order (prefix v. suffix) or non-linear affixation pattern. Inflectional affixes are sensitive to stem class (for example, noun or verb) to such an extent that it is largely possible to decide membership questions of lexical categories just by inspecting the inflected forms. Derivational affixes are even more sensitive, often selecting only stems of a particular class. Software packages that compute the morphological analysis of input words (a complex task that requires undoing all the phonological/orthographic changes that are triggered by putting the morphemes together) generally rely on *continuation lexicons* that list for each root, stem, or affix class which classes may appear next.

Exercise† 5.5 Consider the stem classes N (noun), V (verb), A (adjective), Adv (adverb), and P (preposition), and the suffixes *s* (cars, waits), *'s* (king's), *ed* (waited), *ance* (deliverance), *ing* (eating), *ment* (treatment), *th* (seventh), *ive* (restive), *ous* (bulbous), *y* (beefy), *ion* (hellion), *er* (eater, smarter), *work* (stonework), *ize* (vulcanize), *ization* (vulcanization), *ward* (forward, upward), *wards* (towards), *able* (hearable), *ible* (sensible), *ic* (manic), *ical* (identical), *ly* (manly), *ate* (probate), *ist* (centrist), *ess* (governess), *al* (sensational), *dom* (boredom), *ence* (credence), *hood* (priesthood), *ity* (celebrity), *ness* (greatness), *or* (successor), *ship* (friendship), *ish* (smallish), *like* (kinglike), *less* (painless), *ful* (delightful), *ation* (damnation), and *est* (greatest). Which suffixes apply to which stem categories? Which suffixes can be continued with further suffixes? What roots, if any, enter the analysis?

Exercise† 5.6 Consider the same stem classes as above, and the prefixes *un* (unpredictable), *en* (entrust), *fore* (foretell), *mis* (mistreat), *well* (wellbeing), *mid* (midsized), *dis* (discover), *im* (immaterial), *in* (inexact), *ir* (irrelevant), *non* (nonentity), *vice* (viceroi), and *re* (restart). Which prefixes apply to which stem categories? Which prefixes can be further prefixed? What roots, if any, enter the analysis?

Especially for (5.1)–(5.3), a subtle distinction needs to be made between *newly formed* and *lexicalized* entries. A *totalizer* could be anything that creates totals, but the term typically refers to the kind of adding machine used at horse races. Newly formed entries rely entirely on the compositional meaning (an *X-izer* is something that makes *X*), but once the term enters the lexicon it can accrue all kinds of specific information that is not predictable based on its component parts. In contrast, inflectional affixes always contribute to the meaning of the form in a fully predictable way, and rarely get lexicalized. (The main class of exceptions is furnished by words such as *scissors* and *pants*, which are listed in the lexicon as plurals, and irregular forms such as *went*, which override the regularly inflected **goed*.)

While the linguistic study of morphology must deal with both kinds of entries, for the study of semantics we can drastically curtail the range of facts that we consider relevant. As we have seen in Section 4.5, a grammar that derives *prison*, *guard*, and *inmate* from a single root would actually be simpler than one that needs to account for the actually attested forms, and it is just a historical accident that English does not call these ⁰*keepee*, ⁰*keeper*, ⁰*keep*. Here and in what follows we will make a distinction between ungrammatical (*) and unattested (⁰) forms, reserving the ⁰ for *accidental gaps* such as ⁰*keepee* that would make perfect sense in light of well-attested forms such as *licensee* ‘the one being licensed’, *awardee* ‘the one being awarded’, or *employee* ‘the one being employed’.

Exercise^o 5.7 Consider forms such as *moviegoer*, *surefooted*, and *fastacting*, which are composed of two stems and a suffix but lack the intermediate forms *?moviego*, *?surefoot*, *?goer*, *?footed*, *?fastact*. Is this lack systematic (*) or accidental (⁰)? Can you fit such forms into the scheme (5.1)–(5.4)?

To see that we really can speak of accidents, rather than just a lack of proper understanding of historical language development, we consider an example close at hand. In biology we speak of *phenotype* and *genotype* rather than *phenotype* and ⁰*tectotype*, and in linguistics we speak of *tectogrammar* rather than ⁰*genogrammar*. Why biologists chose *geno-* in preference to *tecto-* is hard to say, but in mathematical linguistics the opposite choice can be traced back to a decision by a single individual, [Haskell B. Curry](#), taken half a century ago. It is clearly beyond the reach of explanatory theory to replicate Curry’s decision process, for if there was a theory of this we would have to assume he did not have free will in this matter, an assumption contradicting our everyday experience (see also Gen. 2:20) that we can name things. Therefore, the best we can do is construct a theory that accounts for the combinatorial possibilities of morphemes only up to (un)grammaticality, as opposed to actual (un)attestation. In other words, our best theory will be one that explains the nonexistent ⁰*genogrammar* and ⁰*tectotype* just as well as the widely attested *tectogrammar* and *genotype*.

To give a slightly different example of the kind of overgeneration we must tolerate, consider the standard English names for groups of animals such as *a flock of birds*, *a school of fish*, and *a pride of lions*. The semantics of these expressions is rather plain: we have *group, bird* and *group, fish* and *group, lion* – there is really no extra content added by the fact that in the fish case we speak of *schools* and in the bird case we speak of *flocks* and never the other way round. Since the lexicon already has the capability to associate arbitrary form (phonological content) with any meaning, it is not a problem to list these and similar items. If we wanted the parser to be able to deal with the fact that *herd* is appropriate for antelope, bison, or caribou but not for ants or dogs, we would need to distinguish *herd*₁ ‘antelope group’ from *herd*₂ ‘bison group’ and so forth. This would be rather uncomfortable, as it is quite clear that both *herd* and *flock* mean *group, animal* just as *glass* means the same thing in *a glass of wine* and *a glass of water*. Here we pursue the less uncomfortable alternative, keeping the meaning of both *herd* and *flock* simple, at the price of being unable to account for the idiosyncratic



distinctions that English happens to make between the two. That this is the right choice is plainly seen from comparison with other languages.

Leaving this kind of tactic data unaccounted for has two important consequences. First, this means that we can decompose the structure of the lexicon only by means of subdirect, rather than direct operations. Second, we need to abandon some of the cherished goals of knowledge representation: rather than describing all that can be known about the meaning of a particular word, we are forced to restrict attention to the essential properties that define it. Once we commit to the view that *glass* has the same meaning in *a glass of water* and *a glass of wine* we have no resources to account for the fact that a *wineglass* will generally have a stem while a *water glass* will not, a *martini glass* will have a conical shape and a *champagne glass* will have a hollow stem, and so on. This is exactly where the line must be drawn: in the theory developed here we must distinguish between the *lexicon*, the repository of linguistic knowledge about words, and the *encyclopedia*, the repository of world knowledge. Random facts about glasses dedicated to particular liquids or group names dedicated to particular animals are neither tectogrammatical nor phenogrammatical; they do not pertain to grammar at all.

Since both compounding (5.3) and derivation (5.1)–(5.2) can change the meaning of the result in unpredictable ways, we need to treat the results of such processes as lexical entries (lexemes) on their own. Because of their predictability, inflected forms need not be listed in the lexicon, and traditional lexicographic practice departs from this rule only in cases where the inflected form filling the paradigmatic slot is not the one we would normally expect (for example, *children* instead of **childs* or *ate* instead of **eated*). At the same time, the category system C_L is highly sensitive to inflectional distinctions, with differently inflected words rarely appearing in the same distribution. For example, singular nouns can naturally appear with certain quantifiers (*every boy*, *a boy*) where plural nouns cannot (**every boys*, **a boys*), while the converse holds for other quantifiers (*some/most boys*, *some/*most boy*).

To make our notation conform to that used in linguistics, we will use angle brackets $\langle \rangle$ to separate inflectional information from the main category and write, for example, $N\langle SG \rangle$ to denote singular and $N\langle PL \rangle$ to denote plural nouns. Within these angle brackets, notation is far from uniform: some authors would have +PL and –PL to denote plural and singular, others would write NUM:PL and NUM:SG in [attribute–value notation](#), and in XML we could write `<num="pl">` and `<num="sg">`. Besides number, there are several inflectional categories one needs to consider, such as [person](#) (1st, 2nd, ...); [gender](#) (feminine, definite, animate, ...); [topic](#) (familiar, known, ...); [tense](#) (past, present, ...); [aspect](#) (perfect, habitual, ...); [case](#) (nominative, accusative, dative, ...); [voice](#) (active, benefactive, ...); [degree](#) (comparative, superlative, ...); [mood](#) (interrogative, negative, ...) and others. Languages differ greatly as to which of these categories they express by inflectional means: for example, Spanish verb inflection distinguishes past, present, and future tenses while English distinguishes only past from non-past, expressing futurity by free morphemes (auxiliaries *will*, *shall*) rather than by affixes. Languages also differ greatly as to which values a given attribute can take: for example,



English number distinguishes only singular from plural, while classical Arabic interpolates dual between these.



One characteristic property of paradigmatic dimensions is that one of the values is generally left **unmarked**, i.e. denoted by zero: for example, in English only the plural is marked by *-s*; the singular has to be inferred from the absence of this marker. In computer science the unmarked value would be the default; for example, the unix command `ls` lists the contents of the current working directory, unless supplied with an argument as in `ls /tmp`. Another characteristic property is that not all combinations are always attested: for example, French marks gender in 3rd person singular pronouns in the nominative, (*il, elle, on*), but the distinction is partially lost in the accusative (*le, la*) and completely lost in the dative (*lui*) – as a result, morphological analysis often returns ambiguous results. A great deal of such ambiguity will be resolved by context, as in *Charlie put the book down* (past tense of *put*) v. *Charlie, put the book down!* (non-past *put*).

Exercise[→] 5.8 Try to account for *be*-imperatives as in *Charlie, be nice to your sister!*, *Be here by 9PM!*, etc. Do you have to modify any assumptions about the imperative of other verbs?



The task of recovering the lexical category that precedes the \diamond is known as part of speech tagging, or POS tagging. **Computational systems** perform this task with about 2.5% error, similar to the error rate of morphological analyzers for languages with complex inflection. The Penn Treebank part of speech tags, on which the performance of English systems is standardly measured, actually conflates lexical categories and their inflections, but conversion to the kind of system we are using here is trivial.

Exercise[°] 5.9 Separate the lexical categories from the inflectional part in the traditional Penn Treebank tags in Fig. 5.1.

To summarize the main features of morphological analysis/generation systems that we will assume in the rest of this book, for each language we assume some (possibly zero) inflection along the dimensions of person, number, gender, tense, mood, aspect, case, degree, topic, etc. We emphasize again that neither these dimensions (formalized as attributes in description-logic-based systems, see Chiarcos and Erjavec (2011)) nor the range of values they can take are quite universal across languages. We also assume a system of lexical categories (parts of speech) such as noun, verb, adjective, adverb, etc. (see Section 5.4 for a more detailed and flexibly defined inventory), but we cannot assume that such a system applies rigidly to all languages. Even the weaker correspondence that concepts expressed by one category (say, verb) in one language will be expressed by the same category in another language (provided the category exists in both languages) is only a statistical tendency.



Once these limitations are acknowledged, the morphologies of different languages are surprisingly uniform. In particular, experience with a wide range of languages (basically, all languages that have standardized **orthography** or **transcription**) has shown that the relation that obtains between a form and its inflectional analysis can be captured by FSTs. This is vacuously true for isolating languages such as Chinese and Vietnamese, where morphological analysis is an essentially empty operation: the real difficulty in

1	CC	Coordinating conjunction	19	PP\$	Possessive pronoun
2	CD	Cardinal number	20	RB	Adverb
3	DT	Determiner	21	RBR	Adverb, comparative
4	EX	Existential <i>there</i>	22	RBS	Adverb, superlative
5	FW	Foreign word	23	RP	Particle
6	IN	Preposition or subordinating conjunction	24	SYM	Symbol
7	JJ	Adjective	25	TO	<i>to</i>
8	JJR	Adjective, comparative	26	UH	Interjection
9	JJS	Adjective, superlative	27	VB	Verb, base form
10	LS	List item marker	28	VBD	Verb, past tense
11	MD	Modal	29	VBG	Verb, gerund or present participle
12	NN	Noun, singular or mass	30	VBN	Verb, past participle
13	NNS	Noun, plural	31	VBP	Verb, non-3rd person singular present
14	NP	Proper noun, singular	32	VBZ	Verb, 3rd person singular present
15	NPS	Proper noun, plural	33	WDT	Wh-determiner
16	PDT	Predeterminer	34	WP	Wh-pronoun
17	POS	Possessive ending	35	WP\$	Possessive wh-pronoun
18	PP	Personal pronoun	36	WRB	Wh-adverb

Fig. 5.1. The traditional Penn Treebank tags

Chinese is in finding the word boundaries, a task whose automation is far from settled. While in languages with more developed morphologies the inflection is often sufficient to establish the part of speech, in isolating languages the first task is observing the distribution of the words relative to the function words (grammatical particles, known in the Chinese tradition as *helping words*).

The fact that the relation between word forms and their inflectional analysis is regular (can be expressed by an FST) means not only that we can use standard [interlinear glosses](#) as needed in the discussion of non-English material, but also that it is reasonable to assume that such glosses can be automatically made available to the input of the syntactic component of the phenogrammar. Conversely, for a generative system that starts with some meaning representation and generates utterances, it is sufficient to trace this process as far as the lexical entries and their inflectional categories: in this book, we need not concern ourselves with how *boy*<PL> becomes *boys* while *child*<PL> becomes *children* and, further, how such strings are turned into sound by [text to speech systems](#).



5.3 Syntax

To see what syntax involves we start with an example from Caesar, who begins his [Commentary on the Gallic war](#) with a description of Gallia and the Gauls:

Gallos	ab	Aquitanis	Garumna	flumen	[]	dividit
N<PL.M.ACC>	PREP	N<PL>	N<SG.M.NOM>	N<SG.N.NOM>	V<3.SG.PRES.IND.ACT>	
Gauls	from	Aquitans	Garonne	river		divide

‘The river Garonne separates the Gauls from the Aquitans’



Since in Latin the focal points of a sentence are at the beginning and at the end, a translator wishing to preserve Caesar's emphasis may opt for the passive, writing 'The Gauls are separated from the Aquitans by the river Garonne'. This preserves an important aspect of the original, namely that it is about the Gauls, and is quite faithful in meaning, in that one simply cannot imagine a situation where the active *x separates y from z* is true while the passive *y is separated from z by x* is false or, conversely, the passive is true but the active is false. This is an extremely strong form of paraphrase, called *truth-conditional equivalence*, and it is a rare case indeed when translations can meet this standard.

Exercise° 5.10 What are the truth conditions of *the river Garonne*? Does it, or does it not, mean the same as *the Garonne river*? In what sense are *Garonne* and *Garumna* identical, especially if we agree with Heraclitus (DK 41) that we cannot step into the same river twice? Is *the summer Garonne* the same as *the winter Garonne*? Is *the languid Amazon* the same as *the cruel Amazon*?

Given the symmetrical nature of *separate*, *x separates y from z* is truth-conditionally equivalent to *x separates z from y*, enabling an analysis in which *y* and *z* are coordinated objects of the separation. Note that the same exchange of arguments would fail for asymmetrical predicates like *protect y from z*, *subtract y from z*, or *deduce y from z*, while the active/passive interchange would work for these verbs just as well as for *separate*.

By making the word order of the translation more faithful to the original, the grammatical structure is changed quite a bit, because *dividit* is obviously active (cf. *dividus*) and the copula, a hallmark of the passive, is also missing from the Latin original. In the Latin original the river Garonne is the subject (agent) of the separation, and the Gauls are the object being separated. For English-speaking students of Latin it is strange at first sight that Latin can have the object first in the sentence, and even stranger that this is accidental; the object could appear anywhere ('flexible' or 'free' word order – see Bailey (2010) for a modern summary).

So how do we know what separates what? For Latin at least, the answer is given by the morphology; it is the noun (or noun phrase) with the nominative ending that does the separation and it is the N or NP with the accusative that undergoes it. The principle is so strong that it applies even in cases where the morphology leaves us in doubt. Consider *omnia*, a form that can stand for both the nominative and the accusative. When we encounter a sentence such as

omnia	vincit	amor
A⟨PL.N.VOC/NOM/ACC⟩	V⟨3.SG.PRES.IND.ACT⟩	N⟨SG.MASC.NOM⟩
all	conquer	love

we must solve the puzzle by first noticing that *Amor* must be the subject (for if it were the object, it would stand in the accusative form *Amorem*), next noticing that conquest demands both a conqueror and a conquered, and *Amor* is the one doing the conquering here, and thus concluding that *omnia* must be the object (this is not demanded by, but at least is compatible with, its morphology), leading to the conclusion that the proper translation is 'love conquers all' rather than 'all conquers love'. That this is the right



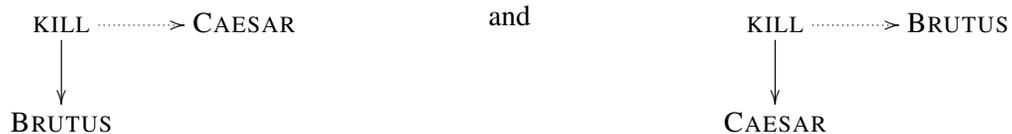
analysis is strongly confirmed by the next clause: *et nos cedamus Amori* ‘so let us (too) surrender to love’. The fact that *omnia* is ambiguous between the nominative and the accusative (and also the vocative) is no more strange than the fact that we must use context to disambiguate between the two senses of *light* in *This window doesn’t let in a lot of light* and *A light load will let the dogs pull the sled faster*. In fact, Frege (1884) makes this his *Principle of Contextuality*:



Never ask for the meaning of a word in isolation, but only in the context of a sentence.

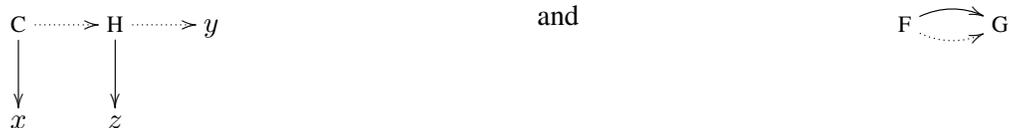
As Janssen (2001) argues, this principle is just as important for semantics as the Principle of Compositionality. Here we concentrate on the syntax, which, based on what we have seen so far, has two goals: first, to delimit the combinations of words (really, inflected word-forms) the language permits, and second, to provide some structural information to compositional/contextual interpretation.

There are two major schools concerning what form semantic representation should take, roughly corresponding to the typological difference between English and Latin. In what we will call the ‘fixed order’ approach, the difference between *Brutus killed Caesar* and *Caesar killed Brutus* is captured by placing the arguments in different order: *kill* is a two-variable function $kill(x,y)$ and in one case we have $kill(B,C)$ while in the other we have $kill(C,B)$, always putting the killer in the first and the victim in the second slot. In the ‘variable order’ approach, the distinction is signified by the nature of the links: we will use straight lines (or just label the arrow by the number 1) for the nominative (subject) position and dotted lines (label 2) for the accusative (object) position. This way we can distinguish



In this chapter, we will use such diagrams more or less intuitively, and we defer the task of interpreting these formally in terms of machines to Section 6.5. In fact, we will use the fixed and the variable order notations interchangeably, as it is often quite trivial to convert between the two.

Exercise^o 5.11 Write $f(g(x, y), z)$ and $f(x, g(y, z))$ as graphs in the variable order notation. Find the fixed order formula for the graphs



This is not to say that the two systems are equivalent: for example, the fixed order system is trivial to extend to functions with three, four, five, or even more arguments,

while in the variable order system this requires the introduction of more arrow colors, as many as the maximum arity used in the fixed order system. (As we have seen in Chapter 2, FOL is quite profligate in this regard, in that there is nothing in the system limiting arity.) Conversely, in the graphical system it is easy to create diagrams such as $F \rightleftarrows G$ that resist formulation in terms of a fixed order system.

Exercise 5.12 Let T be the operation of composing the argument twice with itself, so that $T(\sqrt{})$ means $\sqrt{(\sqrt{(\sqrt{})})}$ i.e. the 8th root. What does $T(T)$ mean?

Perhaps the simplest sentences encountered in everyday language are the imperatives like *stop!*, but even these are composed of several morphemes. In addition to the main verbal stem which describes the nature of the activity demanded, there is an imperative morpheme, which in English can be overtly signaled by intonation, and in Latin by truncation. The imperative indicates the fact that a demand is being made, and there could be others indicating at least the number (SG or PL) and also the person (1st 2nd or 3rd) of the one on whom the demand is being made (English lacks 3rd person imperatives, but many languages have them). Two key elements of the meaning, who makes the demand and to whom, are supplied by the context: it is the speaker, in other sentences generally linguistically marked by a 1.SG pronoun, who makes the demand; and it is the hearer, in other sentences generally linguistically marked by a 2.SG pronoun, on whom the demand is made. While the syntactic string is only $V\langle IMP \rangle$, the meaning is $\text{demand}(\text{PRO}\langle 1.\text{SG} \rangle, \text{stop}(\text{PRO}\langle 2.\text{SG} \rangle))$, where ‘stop’ is used in the sense of ceasing one’s own activity (cf. *stop the car!* and *stop drumming with your fingers!*).

One might speculate that it is the communicative urgency of the situation that makes single-word utterances as in *stop!* or *danger!* preferred to longer expressions like *watch out, you are in danger!* but in fact the more elaborate forms are used quite often. Altogether, the simplicity of such sentences composed of a single inflected verb is rather deceptive. Many languages with more complex inflectional and derivational morphology can pack a remarkable amount of information into a single-verb utterance, as in Hungarian

megkeresnélek

$V\langle \text{PERF.COND.1SG.O2SG} \rangle$

search

‘I would like to set up a meeting with you’ (literally, ‘I would like to conclude the search for you’)

Examples like *faster!*, *now!*, or *greetings!* demonstrate that single-word utterances are not restricted to verbs, and it is not hard to set up the context so that many other utterances, such as *random* or *gas*, appear completely grammatical there. Whether the context includes the intonation pattern, distinguishing *now!* from *now?*, or whether intonation should also be decomposed into morphemes will not be discussed here, but we note in passing that modern grammatical theory opts for the latter choice; see Hirst and Di Cristo (1998). This is not to say that any string of words will be grammatical in the appropriate context. In fact, we still have striking contrasts both in acceptability and in probability (frequency of occurrence) between strings that we feel incomplete: consider

the second, which can be a completely reasonable utterance, for example in response to the question *Which floor is next?*, and **second the*, which cannot.

In these situations, when we are capable of making clear grammaticality distinctions, the formal syntax mechanism (see Definitions 4.6 and 4.7) becomes applicable. Let us first consider $a, b \in \Sigma$ such that a and b , when adjacent, can appear only in this order. The language of ungrammatical strings is $U = \Sigma^*ba\Sigma^*$; the language of grammatical strings is $G = \Sigma^* \setminus \Sigma^*ba\Sigma^*$. Both can be described by the same three-state FSA with a start, an alert, and an end state, except that the accepting/rejecting status of the states is inverted (Fig. 5.2).



Fig. 5.2. Automata for G (left) and its complement U (right)

Notation 5.3 The constraint forbidding the order ba is the same irrespective of what other letters c, d, \dots appear in Σ . This is recognized by reserving the symbol $@$ (read ‘other’) to denote any member of Σ not specifically listed on any arc of the automaton (for easier reading in regexps, we will use o rather than $@$ for this purpose).

The minimal deterministic automaton \mathcal{A} gives information about the *right congruence* associated with the language L (and its complement, see Exercise 4.9 on page 98). Two strings α and β are right-congruent modulo L iff, for every γ we have $\alpha\gamma \in L$ iff we have $\beta\gamma \in L$. This condition is obviously satisfied if the strings α and β will take \mathcal{A} to the same state from the starting state (from every start state, if we permit multiple start states). Conversely, assume that there is a start state s from which α and β lead to different states r and t . Since the automaton is minimal, there must exist some string γ that leads from r to an accepting and from t to a non-accepting state, for if there existed no such γ then r and t could be collapsed into a single state, contradicting the assumption that \mathcal{A} is minimal. Using this γ , we readily see that α and β are not right congruent. This proves the following theorem.

Myhill–Nerode Theorem The states of the minimal deterministic automaton \mathcal{A} accepting a language L are in one to one correspondence with the right congruence classes of L . In particular, if the right congruence associated with some language L has finitely many classes (this is called the *finite index* property), the language must be regular, since a deterministic automaton \mathcal{A} accepting it can be constructed by taking the right congruence classes as states, and the right-multiplication table as the transition table.

This theorem, together with the somewhat older (Kleene, 1956) [Kleene’s Theorem](#), establishes the central commuting square of the diagram in Fig. 4.6, providing a threefold characterization for regular languages in terms of FSA, regexps, and finite (right)congruences. In Section 4.3 we have already mentioned that there is a slight gap



between right and full congruence, and here we can illustrate this point with the language G . Assume a two-letter alphabet $\Sigma = \{a, b\}$ (so that we don't need a symbol $@$ for other letters) and consider the right congruence classes represented by the empty string, b , and ba , denoted by $[\lambda]$, $[b]$, and $[ba]$, respectively. Here a is right-congruent with λ , since for every string γ we have $a\gamma \in G$ iff $\gamma \in G$. Yet $\lambda \not\equiv a$, since in left context b we have $b\lambda = b \in G$ but $ba \notin G$. This of course can never happen if for each each string α in L the reversed string $r(\alpha)$ is also in L , i.e. L is closed under reversal. For the languages that do not enjoy this closure property, in particular natural languages, where grammaticality of a reversed string is rarely guaranteed, we need to consider not just L but also its reversal $r(L)$. The automaton $r(\mathcal{A})$ obtained by simply reversing the arrows will not necessarily be deterministic, and will not necessarily have a single starting state, but there are standard algorithms (the earliest one due to Brzozowski (1962); see also Hopcroft (1971)) to determinize and minimize $r(\mathcal{A})$. In our case, the result is depicted in Fig. 5.3.

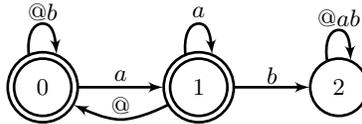


Fig. 5.3. Minimal deterministic automaton for $r(G)$

At first sight, something is amiss, as this automaton also has only three states, yet we know that the full congruence will have at least four separate classes $[\lambda]$, $[a]$, $[b]$, $[ab]$. To resolve the issue, we need to consider \mathcal{A} and $r(\mathcal{A})$ jointly. As usual, we define the state set of $\mathcal{A} \times \mathcal{B}$ as containing pairs of states from the two state sets, and we define the transitions componentwise. However, the usual notion of acceptance will be refined in the following fashion: we say that the product machine \mathcal{A} -accepts if the machine is in a state whose first component was accepting in \mathcal{A} , it \mathcal{B} -accepts if is in a state whose second component was accepting in \mathcal{B} , it \mathcal{AB} -accepts if both of the above are true, and it $\mathcal{A} + \mathcal{B}$ -accepts if at least one of them is true.

Exercise^o 5.13 Let \mathcal{A} and \mathcal{B} be the minimal deterministic finite automata accepting the languages A and B over the same alphabet Σ . Prove that the direct product machine defined above will \mathcal{A} -accept the language A , \mathcal{B} -accept the language B , \mathcal{AB} -accept the language $A \cap B$ and $\mathcal{A} + \mathcal{B}$ -accept the language $A \cup B$.

Exercise^r 5.14 Find a language N for which the minimal deterministic automaton \mathcal{N} has fewer states than the minimal deterministic automaton for its reverse $r(N)$ has. How much bigger can the automaton for the reverse language get? How much smaller can it get?

Note that the state set of the direct product automaton is not simply the direct product of the state sets of the components, since the direct product is not necessarily minimized.

Take, for instance, the state $(1, 1)$ in the product of \mathcal{G} and $r(\mathcal{G})$ depicted in Fig. 5.4 – this is not accessible, because only an a can lead to state $(x, 1)$, and only a b can lead to state $(1, y)$, and no string can end in both at the same time. Obviously, we are only interested in states that are both accessible from some initial state and co-accessible (have an outbound path to some accepting state) – the rest of the states can be *trimmed*.

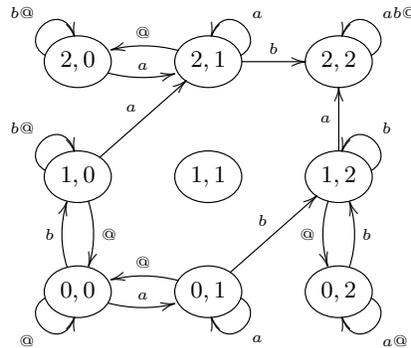


Fig. 5.4. Direct product of \mathcal{G} and $r(\mathcal{G})$

Another key fact the language G illustrates is that combining the left and the right congruence doesn't necessarily yield the full syntactic congruence. This is particularly clear for the case of λ and $@$, which are both right congruent (the grammaticality of a string γ is unchanged by prepending $@$) and left congruent (the grammaticality of a string δ is unchanged by appending $@$), yet they are not congruent, since the string $b@a \in G$ while $b\lambda a = ba \notin G$.

Therefore, the direct product construction provides a relation that can be more coarse (in notation, $>$) than the syntactic congruence, and thus provides only a lower bound on the number of elements in the syntactic monoid. An upper bound can be obtained by extending the logic of the proof of the Myhill–Nerode Theorem: two strings α and β are guaranteed to be congruent if there is no state s that would separate them in the sense of $s\alpha \neq s\beta$. If there is a total of n states, which we number from 0 to $n - 1$, the total distribution of a word α can be described by the state numbers that we arrive at by following the α path, starting from 0, \dots , $n - 1$. As there can be no more than n^n such distributions, we have the following trivial estimate: given a language L , if the minimal deterministic automaton \mathcal{L} accepting it has n states, the index of the syntactic congruence is at least n and at most n^n .

Exercise^o 5.15 Minimize the automaton of Fig. 5.4. How many states does it have?

In many cases, direct inspection of the algebraic structure of the syntactic monoid M_L associated with the language L is quite feasible. In the case of G , we know not only that ba is ungrammatical but also that any string containing ba is ungrammatical, so both left- and right-multiplication of any equivalence class $[t]$ by $[ba]$ will result in

$[ba]$. We also know (not just for this language but in general) that both left- and right-multiplication of any equivalence class $[t]$ by $[\lambda]$ will result in $[t]$, so we may as well look at a reduced multiplication table that omits these two elements. We have already seen that $[a]$, $[b]$, $[ab]$, and $[o]$ are pairwise different classes, so we can set up the multiplication table for M as shown in Fig. 5.5

	a	b	ab	o
a	a	ab	ab	a
b	ba	b	ba	o
ab	ba	ab	ba	o
o	o	b	b	o

Fig. 5.5. Multiplication in M_G (λ and $[ba]$ omitted)

Let us now consider the converse case, when the presence of some element b doesn't forbid, but rather positively demands the presence of some other element a right after it. In English, unlike in French, adjectives typically precede the noun they modify, as in

the fat juror slept (through the trial)
 DET A N V<PAST> (PP)

We can clearly paraphrase this as *the juror slept (through the trial)* and *the juror was fat*. As this last sentence shows, *fat* can appear in predicative position, without a noun following. There is a class of non-predicative adjectives, including words like *supposed* and *former*, where this is not possible; compare *the former juror* with **the juror was former*. Let us denote this class by R . Clearly, in the subject we require DET R N, while DET R alone is ungrammatical: **the supposed slept*. DET itself requires some noun or adjective–noun combination following it, so we already have two examples of the phenomenon of some element a positively demanding some other element b .

Exercise° 5.16 Find further natural language (not necessarily English) examples where some a must be followed by b , but b doesn't have to be preceded by a .

The regex $\neg[[\Sigma^*a[\Sigma\backslash b]\Sigma^*][[\Sigma^*a]]]$ defines a toy language H exhibiting this phenomenon – the corresponding automaton is depicted in Fig. 5.6.

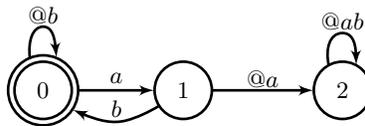


Fig. 5.6. Minimal deterministic automaton for H

Exercise° 5.17 Compute the multiplication table of M_H , omitting the sink class $[aa]$ and the initial class $[\lambda]$.

So far, we have seen that the language of semigroups can describe some syntactic phenomena related to strict cooccurrence and strict non-cooccurrence, but there are many other syntactic phenomena we haven't touched upon. Some of these, referred to as *agreement*, are worth separate discussion, because they occur in language after language. Following the distinction made in Section 5.2 between the POS label proper such as N or V and the inflectional part given between $\langle \rangle$ such as $\pm PL$, we say that the two items *agree* if no string containing $N\langle\alpha\rangle\Sigma^*V\langle\beta\rangle$ is grammatical for $\alpha \neq \beta$.

Agreement is a challenge to theories operating with fixed window sizes, in that the string in the intervening Σ^* can be arbitrarily long: consider *The people/person who called and wanted to rent your house when you go away next year are/is from California*, where the N and the V agreeing in number are separated by 14 words, yet the phenomenon is quite robust (Miller and Chomsky, 1963). Morphologically naive tagsets such as that of the Penn Treebank (Fig. 5.1) can actually make discovery of agreement very difficult: for example, in English the subject can be NN or NNS (singular versus plural), but also NP or NPS (again singular versus plural), while the tag PP (pronoun) covers both singular and plural, and for the predicate the distinction surfaces only in 3rd person as VBP and VBZ (singular) versus VB (base form, and also the plural). Given the potentially intervening material, one has to wonder about the sophistication of the pattern matcher that detects this kind of rule in the data, but once we know it, expressing it in the style of Exercise 5.16, with one kind of string that *must be* followed by another, is not hard.

Exercise^o 5.18 Express the subject–predicate agreement rule of English by a regular expression using the tags $N\langle-PL\rangle$, $N\langle+PL\rangle$, $V\langle-PL\rangle$, $V\langle+PL\rangle$ for singular and plural subjects and predicates. Express the rule with a regexp using Penn tags.

Exercise[†] 5.19 Long Short-Term Memory (LSTM) is a well-known neural architecture, built to deal with long-range dependencies in data. Build a regexp for a language over an n -member alphabet where a_1 must agree with a_2 and a_3 with a_4 , but a_5, \dots, a_n occur freely. Build a sample by random generation from a [Bernoulli scheme](#) over the n outcomes, from which you filter the strings showing the wrong agreement patterns. Train an LSTM on this sample and compare its [perplexity](#) with that of the Bernoulli scheme you started with.



We now turn to one further idea, grouping, which is of great practical and theoretical significance. In observing longer sentences, it becomes quite clear that these are built on a considerably shorter pattern. Take from Section 3.1 the example *Mr. Hug was removed by members of the Police Emergency Squad and taken to Long Island College Hospital*. The basic pattern seems to be ‘someone was removed and taken somewhere’ or ‘someone was removed by someone and taken somewhere’, whether we feel that the agents performing the removal are an essential part of the situation expressed by the sentence or not. This is reflected by the optionality of the *by*-phrase in *x was removed (by y) and taken to h*.

But if we feel that a part that is present in the sentence is optional, we must consider the possibility that other parts, which are not present, could also (optionally) be there. A case in point is the absent from-phrase: usually removal is *from* somewhere, as in *The policeman removed the traffic cone from the intersection*. In this case, Mr. Hug was removed from the elevator shaft, a piece of information known to those who have read the entire story up to the point of the sentence, and thus requiring no overt telling. Another point relevant to finding the simpler structures behind the complex sentence is the presence of the coordination particle *and*: clearly, the whole is composed of two smaller parts *x was removed (from z) (by y)* and, subsequently, *x was taken (to h) (by w)*. There is no guarantee that the people *y* (members of the Police Emergency Squad) who did the removal were the same people *w* who took *x* to the hospital; in fact, the cops may have called an ambulance without this fact being deemed newsworthy by the paper.



Without going into details of how the joining of the two structures *x was removed (from z) (by y)* and *x was taken (to h) (by w)* is accomplished (see Harries-DeLisle (1978) and [the Wikipedia article on Gapping](#)), there is a central fact, already encoded in the naming of the variables, that the person *x* who was removed is the same as the person *x'* who was taken, while the person *y* doing the removal is not necessarily the same as the person *w* doing the taking. It need not be explicitly marked in the sentence that the location *z* where *x* starts is different from the location *h* where *x* ends up, but if these two locations were the same, we would have expected an overt *back* to occur as in *The accused was removed from the courtroom for unruly behavior and taken back later*. That a sentence needs to be evaluated not just at face value but also in relation to what is *not* being said is a major point that we will come back to in Section 5.7; here we are chiefly concerned with *representing* facts like $x = x'$, $z \neq h$, and $\diamond(y = w)$.

What are *x, y, z, w* in the above? They can be proper names like *Mr. Hug*, pronouns like *someone*, or longer descriptions like *members of the Police Emergency Squad* or *Long Island College Hospital* – members of a syntactic class called the NP. The key observation, without which it would be virtually impossible to do syntax, is to note that these longer descriptions must be *grouped* together in that the individual words contributing to them must be understood in relation to the NP rather than the sentence alone. A *long hospital* may be just one rectangular building, but *Long Island* is a specific place (which need not be, and as a matter of fact is not, an island), *Long Island College* is a specific institution somehow related to this place (typically, institutions are related to a place by being headquartered there, but this is just a rule of thumb – Dresdner Bank is in Berlin and Banco Santander is in Madrid), and *Long Island College Hospital* is a hospital associated with this institution, and there need not be anything long about it. By the time we understand this (the human understanding process is highly automated, extremely fast, and not particularly open to introspection), we no longer need to worry whether *long* is used in a spatial sense (*a long building*) or in the temporal sense (*a long hospital stay*) as it is an arbitrary part of a place name, and it is just an accident of history that this place is not called, for example, *Shinnecock County*.

We can thus divide syntax into two parts: the combination of words that make up NPs, and the combination of other words and NPs that make up sentences. Software is now commonly available (not for every language, but certainly for many typologically diverse languages from Basque and Hindi to English and Hungarian) to automatically group words into NPs, and there are several packages such as [YamCha](#) that can be trained on new languages. While the identification of NPs is by no means a fully solved problem (the best systems still fail to find or falsely group about one NP in 20), in the rest of this book we will assume that the NP-level parsing and generation problems are under control, especially as the issues of NP syntax are largely orthogonal to the issues of semantics.



Exercise^o 5.20 Find the NPs in the following sentence. *He was badly shaken, but after being treated for scrapes of his left arm and for a spinal injury was released and went home.*

In the sentence that we started out with, *Gallos ab Aquitanis Garumna flumen dividit*, we may assume that verbal conjugation and the declension of case- and preposition-marked NPs are already handed to us in the form [Gaul]_{N<PL.ACC>} [Aquitan]_{N<PL.ABL>} [Garumna flumen]_{NP<SG.NOM>} [divido]_{V<3SG.PRES.IND.ACT>}. Using such a ‘shallow’ syntactic representation that has the NP chunks and the morphological analysis but little else (beyond word order) is more a matter of convenience than a theoretically justified level of representation. What ‘deeper’ syntactic representation, such as a [parse tree](#), a [Reed–Kellogg diagram](#), a [dependency graph](#), or perhaps some hybrid of these, is justifiable as a separate representational level is a highly debated question, and we see no need in this book to make the determination. Yet the computationally oriented reader needs to pick *some* syntactic formalism, and in the next section we describe one formalism we consider a very reasonable choice.



5.4 Dependencies

One characteristic that distinguishes the professional from the amateur semanticist is the concern for universality. Amateurish knowledge representation approaches generally do not amount to a great deal more than paraphrasing in a standardized vocabulary, perhaps coupled with a simple (generally first order) logic formalism. But as soon as we have more than one language, it is evident that their concepts do not align perfectly, a matter we shall discuss in more detail in Section 6.2. For example, in English *wood* is the root of both *woods* ‘forest’ and *wooden* ‘made out of wood’ but the language has a separate word for *tree*; German has three separate words *Wald*, *Holz*, *Baum* for these; while Hungarian *fa* pulls together the individual tree (*Baum*) and the tree material (*Holz*), but uses a separate word *erdő* for woods (*Wald*).

In the Middle Ages, the main approach to the problem of languages differing was to declare one language as *the* basic ancestral language, and treat all other languages as deviations from this ideal. Debate focused on which (Biblical) language, Latin, Greek,

Aramaic, or Hebrew, should be the true language of ideas, and to this day we find enthusiasts arguing for one of these as the obvious solution. By the 18th century the primary vehicle for studying the deviations was the Indo-European family, where words can be traced back to a remarkable time depth.

As linguists came into contact with an increasing number and variety of languages in the 19th century, it became evident that the range of variability was so large that finding a common ancestor was hopeless, and attention shifted to *universal grammar*, a set of principles shared by all languages. The idea that the grammars of all languages have the same substance but differ in accidents goes back to the Schoolmen, Roger Bacon and the *modists* in particular, and will be discussed from a more general standpoint in Section 9.3. Here we will discuss a specific instantiation of the general idea, *Universal Dependencies (UD)*, which commits us to a well-defined theory of POS tags, morphology, and syntax.

First, the parts of speech. UD makes a distinction between *open classes*, *closed classes*, and *other*. The open classes have a large membership, and new entries are added to the vocabulary in these classes all the time (hence the name). UD recognizes exactly six open classes: ADJ (adjective); ADV (adverb); INTJ (interjection); NOUN (noun); PROPN (proper noun); and VERB.

Exercise[†] 5.21 Count the number of words falling into a given POS class in a machine-readable dictionary. Take a frequency count from the output of a POS tagger and thereby estimate the relative frequencies of words in each of the open classes. How do the two counts (called *type* and *token* frequency counts) differ?

Next we have the eight *closed classes*: PART (particle); PRON (pronoun); SCONJ (subordinating conjunction); ADP (preposition/postposition); AUX (auxiliary); CONJ (coordinating conjunction); DET (determiner); and NUM (numeral). It takes significant changes in grammar and style to affect the inventory of words in closed classes: for English, it is fair to say that dozens or perhaps hundreds of new words are added every day, while it takes decades, sometimes centuries, to change the membership of the closed classes (hence the name). Finally, there are the *other classes* necessitated by the exigencies of text processing: PUNCT (punctuation); SYM (symbol); and X (unspecified POS), the fallback value.

The major distinction made in UD corresponds very well to the traditional grouping into *content* and *function* words we discussed in Section 5.2. Repeating Exercise 5.21 with the closed classes included would show that the function words, while few in number, take up an extraordinary amount of the probability mass; close to half of the tokens are function words. Putting interjections into the function rather than the content category, perhaps the only disputable decision made in UD, would increase this proportion only slightly, since interjections are not very frequent to begin with. Punctuation, on the other hand, is close to 10% of the tokens, often more.

What, exactly, makes this list of 17 major categories universal? As we shall see in Section 6.3, it is not at all the case that each and every language manifests each and every one of these categories: the claim is simply that (i) most languages will have most of



Comp

them and (ii) cross-linguistically, the typical translations will preserve category. There is also an implicit claim of exhaustiveness, that we will not find languages that require further major categories in their grammatical description, but this is even harder to assess, in that we don't (yet) have a detailed grammatical analysis of each language with the required cross-linguistic mappings. Be that as it may, it is hard to find a grammatical description that does not rely on these categories, and having a well-specified inventory at hand is extremely useful, especially in multilingual work.

As we have seen in Definition 4.8 (page 99) and also in Section 4.2, from a formal standpoint, if a language L is conceptualized as a stringset over its vocabulary V , the system of lexical categories emerges as the intersection of $V \times V$ with the Myhill–Nerode congruence \equiv_L . In other words, we say two words v and w belong in the same category iff they can be substituted for each other in every context without change of grammaticality. This is an extremely stringent requirement, and certainly we can find words like French *cousin* and *cousine* which in English mean the same thing, ‘cousin’, yet one requires *un* and the other *une*, as is evident from the ungrammaticality of **un cousine*, **une cousin*. We thus need to divide the categories into *subcategories* based on *inflectional features*, such as Gender, Animacy, Number, Case, Definiteness, and Degree (most pertinent to nouns and nominals), and Mood, Tense, Aspect, Voice, Person, VerbForm, and Negativity (most pertinent to verbal forms). These are precisely the distinctions we put in \diamond in Section 5.2 above.

Note that such subcategories are not always realized by inflection; they can also be *inherent* as in *cheveux* ‘hair’ (masculine) or *chaise* ‘chair’ (feminine), which don't have alternants in the other gender that would mean the same thing except for the gender difference, as is the case, for example, with Russian *kartofel* ‘potato’ (masculine) and *kartoshka* ‘potato’ (feminine). UD also allows for inherent distinctions (subcategories) in features such as [PronType](#), [NumType](#), [Poss\(essive\)](#), and [Reflex\(ive\)](#). As with the major categories, there is no expectation that every combination of every subcategory expressible by these features will be relevant for any single language, just that such subcategories offer a good way of sorting the words (and, on occasion, bound morphemes) so that translational equivalents are likely to get into the same subcategory. But unlike with major categories, where cross-linguistic category mismatches are rare, at the level of subcategories such mismatches are quite common, as with French *table* (feminine) and German *Tisch* (masculine).

We will return to the discussion of parts of speech in Section 6.3, but one thing should already be clear: the distinctions expressed by POS are syntactic in nature. One would be very hard put to express the difference in meaning between *un* and *une*, or *kartofel* and *kartoshka* – the difference exists simply because gender is a category operative in the grammar of French and Russian. It would not occur to us to treat English *a* and *an* as different concepts just because one is used before a consonant and the other before a vowel, and it would be just as wrong to treat the concepts *un* and *une* differently just because one is used before masculine and the other before feminine nouns.





The main reason we discuss UD at this level of detail is that it offers data: Nivre et al. (2016) list 37 treebanks covering 33 languages and containing over 7.5m words of grammatically analyzed data. The 4lang system discussed at various points in this book currently relies on the *Stanford Parser* (Chen and Manning, 2014) for its input, but as this parser already provides UD-formatted output we are gradually moving away from the older format. Those interested in semantics who don't want to take on the burden of building their own parser will find it expedient to use UD parsers and treebanks to test and refine various aspects of their system.

The classical introduction to dependency grammar is Tesnière (1959), and UD actually stays remarkably close to the spirit of this work. The key idea is that syntax is formulated in terms of a *dependency* relation among words that can take many forms. For example, an adjective modifying a noun is depicted by an arrow running from the noun to the adjective and labeled *amod* (adjectival modifier). The entire syntactic structure of a sentence is depicted as a tree, with leaf nodes corresponding to the words, which are tagged for POS and inflection as described above; an abstract root node; and edges bearing various labels running from each word to its dependent(s), if any.

Here we survey the label inventory of UD, currently admitting 40 different labels, and relate it to the more sparse 4lang label inventory, which admits only three (these are numbered 0, 1, and 2). This economy is made possible by two central design decisions that distinguish UD and 4lang. First, UD is intended as a syntactic representation, aiming at an exhaustive description of the syntactic structure of a sentence, while 4lang is a semantic representation. This means that in 4lang we can abstract away from much of the phenogrammar that distinguishes, for example, *The office of the Chair* from *The Chair's office* (here and in what follows we take our examples from Nivre et al. (2016) to the extent feasible), representing both as *office, Chair HAS*. Second, UD uses rooted trees to depict the structure, while we use hypergraphs (see Definition 4.5 on page 90).

Both of these factors contribute to making the semantic hypergraph more compact than the syntactic tree. In UD, a significant effort is made to expose the dependencies between the word stem and its grammatical marker: the link type *case* is used to label the arc running from a nominal head to a preposition, and the same label is used for possessives. Quite often, case marking of this sort (whether it is spelled out in a free morpheme or as a suffix) is contentful: for example, the superessive case in Hungarian denotes the spatial relation *ON* of one thing being on top of the other. In these situations, 4lang will treat this as predication, for example, *a könyv az asztalon* 'the book on the table' will be treated as *book ON table* with a subject (type 1) link from *book* to *ON* and an object (type 2) link from *ON* to *table* irrespective of whether an overt copula (UD link type *cop*) is present or not. But to the extent that the case is purely conventional (lexically driven), as in *John met with Mary* or *John met Mary*, we concentrate on the tectogrammar (Section 4.6) and depict both as in Fig. 5.7.



UD takes responsibility for encoding *all* relations. For example, it uses *mwe* or *name* to link together parts of *multi-word expressions* such as *as well as* or *Hillary Rodham*

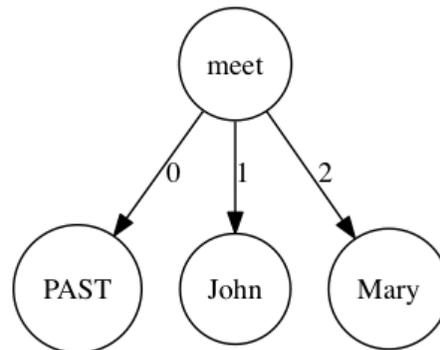


Fig. 5.7. 4lang representation of *John met Mary*

Clinton, which we will be content to list as a single lexeme (see Chapter 6); it links those words that were said in error (reparanda) to the correction with the link type `reparandum`; it links parts of words separated by a typographical error with the link type `goeswith`; it links the prepositional part of phrasal verbs to their heads or parts of compounds with the link type `compound` (*call up, three thousand*); and it deals with changes to another language (link type `foreign`). 4lang doesn't have the resources to deal with disfluencies. The semantic representation assigned is the one that would go with the corrected sentence, as if the errors were never made. In semantics generally there is no reason to link in punctuation (UD link type `punct`) unless it has semantic force, as exclamation points or question marks often do. These are linked as conceptual elements `imp` (imperative) or `?`, typically attached to the subject of the imperative or the part being questioned.

Because it encodes all relations, UD has a fallback dependency type `dep`, and a technical link type `root` to connect the root to the main verb. These have no functional equivalents in 4lang, which is closer to traditional dependency grammar in that the root is taken to be the main verb and no separate root node is maintained. Since coordination is simply treated as superposition of the hypergraphs corresponding to the conjuncts, no dedicated `cc` (coordinating conjunction), `parataxis`, or `discourse` link types are required in 4lang.

The difference between the syntactic and the semantic representational style is quite striking when the syntax is complex, as in non-constituent coordination. Consider *John won bronze, Mary silver, and Sandy gold*. The UD analysis, reproduced in Fig. 5.8 from the excellent [UD website](#), relies on chains using the link type `remnant` to link *John* to *Mary* and *Mary* to *Sandy*, and *bronze* to *silver* and *silver* to *gold*. The semantic representation is depicted in Fig. 5.9.



The same simplification is seen in topicalization and other similar word order changes marked by the `dislocated` link type of UD: the semantic representation simply ignores the change and treats *Bagels I like*, with the object *bagels* fronted, the same way it would treat *I like bagels*. Expletive (pleonastic) elements like the existential *there*

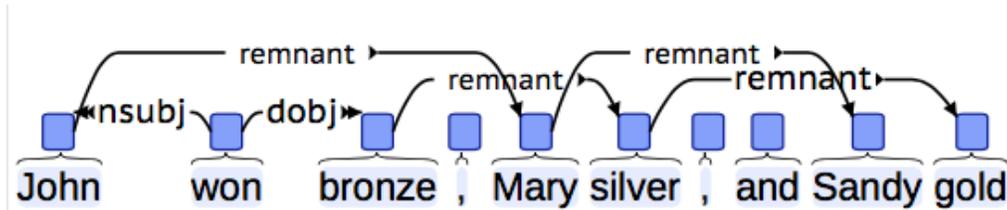


Fig. 5.8. UD analysis of *John won bronze, Mary silver, and Sandy gold*

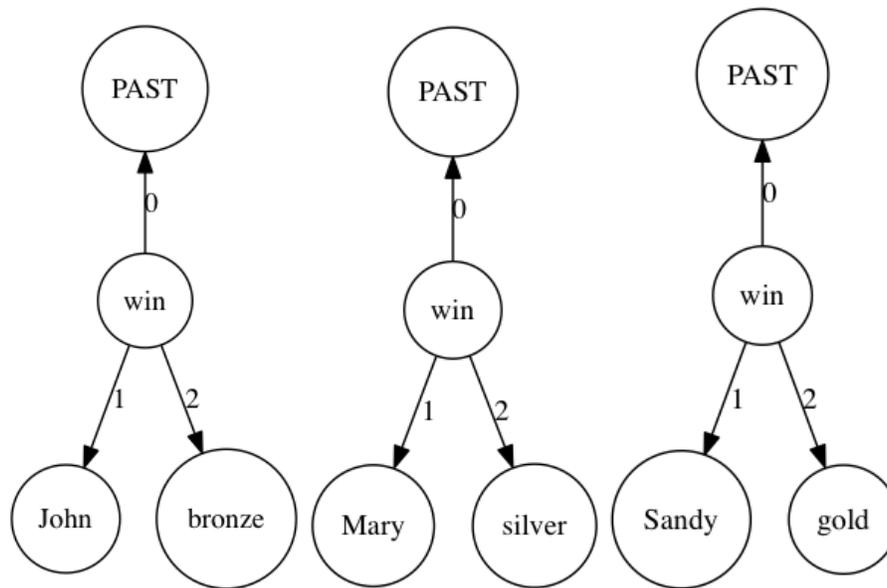


Fig. 5.9. 41ang representation of *John won bronze, Mary silver, and Sandy gold*

of *There is a ghost in the room* have their own link type `expl` in UD, but the semantic representation has no use for these, treating the meaning simply as `ghost IN room`. We defer the `list`, `appos`, and `vocative` link types to Section 5.7, where we will discuss attribute–value matrices (AVMs), and defer `neg` to Section 7.3, where negation is discussed in detail.

We note only in passing some of the UD link types that we see as too closely tied to the syntax of English, chief among them `aux` (auxiliary) and `auxpass` (passive auxiliary). In a less English-centric system these would be analyzed as operators (typically modal operators; see Section 7.3) that modify the main verb (our link type 0), especially as they are often expressed by morphological means rather than by separate function words. The same goes for `determiner`, which links function words like *the* and *which* to their head nouns, and the `mark` link type used in English to denote the introduction of subordinate clauses by function words such as *that* or *after*.

With the special cases out of the way, we now come to what we consider the core UD link types. The simplest, and perhaps best understood, dependency is that between a head noun and an adjective. Consider *An angry boy smashes his toys*. Even though semantically the key element is supplied by *angry* (boys normally don't smash their toys), it cannot stand on its own (**An angry smashes his toys*); it depends on a head noun, which may be contentful as in the case of *boy* or empty as in the case of the pronoun *one*, as in *Boys normally take good care of their toys, but an angry one will often smash them*. UD denotes this syntactic link by `amod`, and we use the link type 0 (standing both for attribution and for `IS_A`) to encode the fact that the property meant by the adjective is attributed to the head noun.

There are several other UD link types that are distinguishable at the syntactic level, but signify the same kind of attributive semantic link: `acl` is used between a clausal modifier and a head noun as in *the issues as he sees them*, relative clauses as in *the man you love*, content clauses as in *the fact that nobody cares*, and so on. We also use the link type 0 for cases where the modification affects a verb (so that the modifier is an adverb rather an adjective), where UD uses `advcl` or `advmod`; and we treat numeric modifiers (UD `nummod`) and nominal modifiers (UD `nmod`) the same way.

The other two well-understood dependencies are between a verb and its subject, denoted by the link type `nsubj` in UD and the link type 1 in `4lang`; and between a verb and its direct object, denoted by `dobj` and 2, respectively. These basic link types go back to the Greek grammatical tradition, and are part of almost all systems of syntax and semantics (see Section 4.1 for a bird's-eye overview). The `4lang` system takes them to be semantic, even though the numbers '1' and '2' were appropriated from a syntactic theory, Relational Grammar (Perlmutter, 1983), which uses them in a more complex manner. This means that both the active *The river Garonne separates the Gauls from the Aquitans* and the passive *The Gauls are separated from the Aquitans by the river Garonne* will have the same semantics depicted in Fig. 5.10, and the more syntax-driven link types such as UD `nsubjpass` (passive nominal subject) and `csubjpass` (clausal passive subject) are treated as 2s. Also, since the semantics is kept free of syntactic typing, UD `csubj` (clausal subject) is not at all distinguished from UD `nsubj`, both corresponding to 1 in `4lang`.

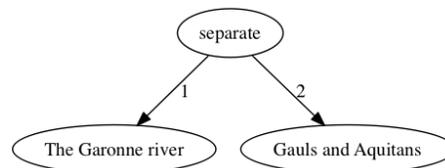


Fig. 5.10. `4lang` representation of active and passive sentence

For the sake of completeness, we discuss the remaining three UD types: `ccomp` (clausal complement), `xcomp` (open clausal complement), and `iobj` (indirect object).

Generally, what UD considers a clausal complement will simply be an object (type 2) for us, because at the semantic level we see no difference between *The boss said to start digging*, *The boss said, ‘start digging’*, and *The boss said, ‘(hey), you, start digging’* (with or without the quotes). The situation is far more complex with indirect objects, and Relational Grammar actually employs a primitive link type ‘3’ to encode these. Yet there is very little cross-linguistic coherence in indirect objects and, as we shall see in Chapter 6, 41ang gets by without these.

In summary, we should emphasize that the task of assigning to a sentence a syntactic analysis such as a UD dependency tree is much harder than assigning a semantic analysis *that is based on the syntactic structure*. We cannot get to the top of Mount Everest without climbing the [Hillary Step](#), but the real difficulty is not in climbing these 12 meters; the hard part is to get there first. When we say that the semantics doesn’t require links for punctuation this is not to say that these links are not useful, for example for disambiguating between *Eats, shoots, and leaves* and *Eats shoots and leaves*. Just as putting up scaffolding is often extremely useful, indeed essential, for putting up a building, creating an intermediate syntactic structure is extremely useful, and many would say essential, for finding out the meaning of a sentence.



5.5 Representing knowledge and meaning

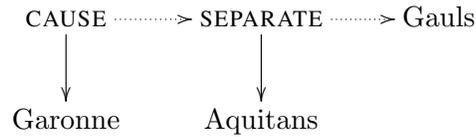
The issue of how to represent the meaning of linguistic expressions is perhaps even more unsettled than the issue of representing their syntactic structure, but this is a book about semantics, and we will discuss meaning representation in far greater detail, gradually developing a highly specific formal theory. There is well-developed line of thought, originating with Chomsky (1973), and usually referred to as the *autonomy of syntax* thesis, claiming that *the syntactic rules and principles of a language are formulated without reference to meaning, discourse, or language use* (see [Fred Newmayer’s class notes](#) for this particular formulation and much relevant discussion, and see Anderson (2005a) for a strongly opposed view). In this book we assume syntax to be autonomous not so much because we feel we are capable of settling the debate in this direction, but simply because it makes good engineering sense to maximally isolate our theory of semantics from the details of syntactic representation.



Ling



[Knowledge Representation \(KR\)](#) is a separate field of study. Historically, KR started out as a subfield of [AI](#), but contemporary KR has to a large extent moved away from some of the original cognitive concerns of AI. Meaning representation is not generally viewed as a separate field of study in its own right, but rather as a chapter of semantics. Since meaning representations are the key data structures used in semantics, they tend to take on the character of the semantic theory that employs them: theories of a logical nature tend to use formulas, while theories of a more cognitive kind usually promote diagrammatic (network) representations of meaning. Anticipating developments in Chapter 6 we will say, with some simplification concerning tense, mood, and aspect, that the semantic representation of our example sentence will be



where CAUSE and SEPARATE are two-argument predicates, the latter corresponding to the state ‘being separated’ or ‘are separate’ rather than the process ‘doing the separation’.

Perhaps the most striking thing about this representation is the inclusion of underlying graph nodes such as CAUSE that have no direct reflection on the surface. The sentence does not overtly say ‘the Garonne is causing the Gauls and the Aquitans to be separated’, yet we make the claim that the same representation is adequate for both the original sentence and its longer paraphrase. How this stance can be justified is left to the next chapter, but we note here that the idea goes back at least to the [generative semantics](#) of the late 1960s and early 1970s. At that time, a sentence like *Floyd broke the glass* was commonly analyzed as being composed of several more elementary underlying structures, roughly as *I declare to you that it past that it happen that Floyd do cause it to come about that it BE the glass broken*, not just with the hidden element of causation made explicit but also with hidden elements of declaration/assertion, tense, result state, etc.



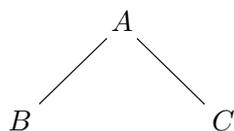
The underlying syntactic/semantic representation was taken to be a tree, and this would become, by a series of tree manipulation steps, converted to the observable (surface) form. We will actually make good use of several descriptive devices pioneered in the generative semantics tradition, such as defining *kill* as ‘cause to die’, but not necessarily others, such as defining *break* as ‘cause to be broken’. We will also discuss some of the phenomena that motivate the assumption of superordinate speech acts like *I declare to you* which remain hidden in the surface form, but will generally treat these as default implications rather than as actually part of the meaning. Since our basic structures will be machines, rather than trees, the constructions will be machine operations, rather than tree transformations. This makes the parse tree an epiphenomenon, nothing more than parentheses that serve to describe the order of rule application.

For the most part we will opt for the variable order notation, and discuss how a fixed order notational variant (using only unary and binary predicates) can be created. The graph-based meaning representations primarily serve a pedagogical purpose, linking the system of semantics to standard systems of KR in the same way a fixed order notation would link semantics more closely to logical calculi. But neither the fixed nor the variable order notation is more than notation – the objects we take as underlying are the machines introduced in Section 4.3. Meaning representations are systems of machines, and the entire syntax–semantics mapping will be built using machines.

This should be contrasted with the formal systems that were introduced in the 1960s and 1970s and are assumed without argumentation in much of the literature to this day. These systems took [trees](#) to be the basic representational objects both in syntax and in semantics. Both generative semantics and its main theoretical opponent at the time, [transformational grammar](#) assumed that the parse tree is the result of some tectogram-matical process that starts with a deep (underlying) constituent tree or, in the case of more



complex sentences, several such underlying trees. This view would inevitably make tectogrammar itself some kind of tree manipulation system, and it was commonly assumed in linguistics that not just the sentences but also their basic building blocks, the words, came equipped with some kind of tree structure. Subsequent work has shown the need for a more nuanced conception, one that divorces the *temporal* notion of constituency and the *architectural* notion. A simple tree diagram such as



summarizes two different notions: first, that a unit of type A is obtained from putting together units of type B and C , and second, that the result is obtained by concatenating B and C in this order. In general, these two notions need not come hand in hand, as we know, for example, from studying the syntax of mathematical notation by the same methods. When we put together a function \cos and a constant π we obtain another constant $\cos(\pi)$, and when we put together the function with a variable as in $\cos(x)$ we obtain another (dependent) variable. There are only two substantive units in this example: the cosine function A , and the B that it applies to, the constant π or the variable x . This is broadly analogous to the case of simple predication as in *John sleeps*, *John is sleeping*, or *John is fat* except that the order is turned around (we would want to say that *sleep* is the function and *John* is the constant it is applied to).

While the intention of somehow putting together two elements may be clear and simple, it requires a significant amount of technical development to actually realize it. First, the expression is composed of not two, but four parts: the function, the argument, and the two parentheses. The same is true for the linguistic expressions, which also require some kind of glue, such as the subject–predicate agreement between *John* and *sleeps*, or the copula *is*. Second, there is a sense that the left and the right parentheses somehow belong together: either an expression omits both (which may be feasible in the simplest cases, but will lead to ambiguity in cases like $\cos \pi + x$) or it includes both, but we cannot have one without the other. Third, we may wish to keep together those cases where the argument is a constant and where the argument is a variable, rather than devising separate tree structures for each.



One important development was replacing the symbols A, B, C by more complex data structures. For example, using [attribute grammars](#) and a property ‘mutable’ that is 1 for variables and 0 for constants, a single rule that can combine a function with a variable or a constant can be formulated by making sure that the mutability attribute is carried from C to A . It will be critical for understanding how meaning representations work that components of a rule are themselves complex data structures that can have other interactions besides being juxtaposed by the rule.

Another important development was the understanding that strings (linearly ordered sequences of units) are but a limiting case obtained when all the things contained in one

data structure such as *B* precede all the things that are contained in *C*. Less perfectly synchronized configurations, with some parts of *B* ending later than where some parts of *C* begin, are found at all levels of language, including phonology and morphology. As a result, strings themselves have been replaced in phonology by more complex structures, [autosegmental representations](#), which carry partial synchronization information across independent articulators such as the vocal cords and the lips. There is very clear evidence from etymologically and geographically unconnected language families, from [Yokutsan](#) to [Semitic](#), that the basic functional elements (usually appearing as prefixes or suffixes) and the basic content elements (stems and roots) can appear interdigitated, as in the Arabic examples discussed in [Section 5.2](#).

The same effect is observable in syntax, where constituents can be broken up by intervening material. Consider *John couldn't get over the breakup with Mary and decided to call her up one more time rather than letting her go forever*, where we find three multiword lexical entries, *get over* 'feel happy again', *call up* 'telephone', and *let go* 'disengage', exemplifying three different kinds of syntactic behavior. In the case of *call up* the object can follow the main verb or the particle, as in *John should call Mary up* or *John should call up Mary*; in the case of *let go* the object must appear immediately after the verb, as in *John should let Mary go*, **John should let go Mary*, and in the case of *get over* the object can only appear after the particle, *John should get over Mary*, **John should get Mary over* (unless in a very different sense, as in *We should invite/get Mary over for the party*). This last case is easy to describe in the syntax by saying that *get over* is a single indivisible lexical unit, but this of course entails some highly unusual morphology where the standard verbal suffixes appear in the middle of this indivisible unit as in *getting over*, **get over-ing*, *gets over*, **get over-s*, *(has) gotten over*, **(has) get over-ed*.

There are several mechanisms to deal with the discontinuous cases, of which we single out [Tree-Adjoining Grammar \(TAG\)](#), [Combinatory Categorical Grammar \(CCG\)](#), and [Lexical-Functional Grammar](#), because all three formalize important aspects of the problem. TAG makes it easy to *substitute* patterns in one another, CCG has good facilities for handling *non-constituents*, and LFG is designed from the ground up for *co-description*, i.e. maintaining multiple kinds of information links across constituents. All of these ideas play a role in the machine implementation, but for [KISS reasons](#) neither of these grammatical theories will be adopted wholesale. Rather, we will stay within the finite state realm (see [Definition 4.3](#) on page [88](#)) and build the context-sensitive functionality by methods similar to those used in computational phonology and morphology.

5.6 Thoughts in the head

One area where much of the technical apparatus discussed so far comes into play is the description of thoughts in other people's heads. Clearly, such a thing is possible only because we have clear indications of what is happening there: when someone says *I am angry* there is every reason to suppose they are angry. Since this is more or less trivial,



we discuss here a more subtle case, where the conclusion is obtained by implication. Consider one person, Joe, describing what happened to another person, Jack, as follows: *It is tragic that he died of hunger*. Another person, Jill, may object to this, noting that Jack was in fact engaged in a vile plan for mass murder by poisoning the water supply when he got caught in a small room in the water purification plant and died: *No, it was a blessing to all*.

Our model of the situation involves three different representations: (1) hunger CAUSE Jack Die-PAST out in the world, (2) Tragic[1] in Joe's head, and (3) Blessing[1] in Jill's. What does it mean for (1) to be 'out in the world'? This may not be the actual truth of the matter, for Jack could have died of his own poison, which would not change (3) at all – maybe Jill would now speak of 'poetic justice' but her overall assessment that the outcome was good would be unchanged. Instead of saying 'out in the world' we will take a more nuanced stand, and say that Jack's death is now part of a jointly maintained store of knowledge, standardly called the *common ground*. (1) conveys two different things: the subordinate clause conveys that Jack died, and the main clause conveys Joe's assessment of this event.

Remarkably, by saying *no* in (3) Jill can only challenge the main clause, but the subordinate clause is now part of the common ground. Compare this to (1') *It is likely that he died of hunger*. Here (1) is not established as part of the common ground, and it is just as easy to challenge the subordinate clause by *No, I heard he escaped* as it is to challenge the main clause by *No, it's rather unlikely, given the amount of poison the coroner found in his body*. When we say things, we would like to convey ideas to the hearer, and this is often better accomplished by letting the hearer finish the thought. For example, when we say *The former real estate mogul has stopped cheating on his wife* we do not overtly say that he used to, but the implication is evident, as you cannot stop something you have not done. We have also slipped in another, perhaps equally insinuating statement, that he is no longer a real estate mogul; perhaps he went bankrupt? To top this off, we now also have what is called an *existential presupposition*, that the person has a wife, and has also had one or more lovers.



Thinking about presuppositions goes back to Frege's "On Sense and Reference", Frege (1892), and Strawson (1950), whose major concern was the case where the existential presupposition fails, as in *The present king of France is bald*. (Actually, the key examples for studying existential presupposition come originally from theistic writing, to which we will turn in Section 9.1, and usually involve figures from Graeco-Roman mythology, about whom the author could secure strong agreement that indeed they did not exist.) There are three major directions for approaching the issue: we may simply say that such sentences are false (Russell, 1905), we may say they don't mean anything, and we may say they mean something but it is neither true nor false. The first view gets into difficulties because existential presuppositions, and presuppositions in general, cannot be undone by negating the sentence that gives rise to them: when we say *The present king of France is not bald*, this will again have to be taken to be false. Russell, witty as ever, quips that maybe he wears a wig. The second view is problematic because there

are many entities whose very existence we are uncertain about: for example, one can prove many theorems about ‘the first nontrivial root of ζ not on the critical line’ without knowing whether such roots exist. Since we don’t want the status of such sentences to flip-flop according to our current best theory of what exists and what does not, we are forced to follow the third approach, that they mean something but they are neither false nor true. But if so, what do they mean? As far as the individual words are concerned, we will follow [Locke](#) and assume that what they mean are ideas, thoughts in the head. Many philosophers, starting with Plato, assume that ideas have an independent realm of existence, the “third realm”, distinct both from the external world and from the internal world of consciousness, but we need not take a stand on this matter here.



Let us briefly compare this with the standard formal model of lexical entries introduced by Katz and Fodor (1963), discussed in Section 3.9. This model employs a tree structure where the root node is associated to the (phonological) form, and the top branches each give a single, disambiguated sense of the lexeme, so that for *chrome* the first branch would be for ‘hard and shiny metal’, the second for ‘eye-catching but ultimately useless ornamentation, especially for cars and software’, and so forth. Within a single sense Katz and Fodor used a set of binary features to describe the systematic aspects of the meaning, for example that *chrome*₁ is +PHYSOBJ (physical object) while *chrome*₂ is –PHYSOBJ, and largely unsystematic *distinguishers* to define those aspects of the meaning that they felt were important even though they were not amenable to expression in terms of binary features.

Two main lines of criticism were leveled against the Katz–Fodor model: cognitive scientists attacked the [Prague School](#)-style [binary features](#), and the formal semantics community faulted the theory for its inability to confront what they saw as the central issue of meaning, how concepts relate to things in the real world. Lewis (1970) decried the model as ‘markerese’ for interpreting words in an uninterpreted language of markers, rather than in model-theoretic terms. Still, variants of this much-despised ‘markerese’ survived as the principal tool of lexical semantics in generative grammar from Jackendoff (1972) to Pustejovsky (1995) and beyond. We believe the model owes its resilience to a clear, and therefore clearly debatable (Bolinger, 1965), restatement of the Aristotelian notion of *eidopoios diaphora* as the ‘distinguisher’, and to the great heuristic appeal of systematizing structural decomposition in terms of a featural theory of ‘semantic markers’. Evidently Katz and Fodor did not approach the issue lightly, and a surprising amount of what earlier generations of philosophers and grammarians had to say about word meaning can be restated in their formalism clearly, and without much technical difficulty, at least for nominal bases (nouns and adjectives).



Both in the Katz–Fodor theory of lexical semantics and in the theory presented here, the meaning of nominals is conceived of as a conjunctive bundle of properties. Whether we need to enforce the structure of binary oppositions is unclear – this is not an essential feature of the present theory and was not, we would argue, an essential feature of the Katz–Fodor proposal either. Given the lack of commitment we have to any defining vocabulary (see Sections 4.5 and 6.4), the requirement that only a restricted set of



primitives should appear as defining properties no longer makes sense. In the theory developed here circular definitions are quite permissible, and cause no difficulty in the formal theory given by the CFG above. This amounts to an unabashedly [Meinongian](#) theory of nominals along the lines of [Parsons \(1974\)](#), and we now demonstrate that such a theory is positively required for the hard cases of hyperintensionals where we need to compare not just nonexistent objects, for which the intensional treatment of opacity works fine, but also *necessarily* nonexistent objects whose extension is empty at every index.

Consider *Pappus tried to square the circle/trisect the angle/swallow a melon*. In one case, we see Pappus intently studying the works of Hippocrates, in the other we see him studying Apollonius, and in the third case we see him in the vegetable patch desperately looking for an undersized melon in preparation for the task – clearly the truth conditions are quite different. We may very well imagine a possible world where throats are wider or melons are smaller, but we know it for a fact that squaring the circle and trisecting the angle by ruler and compass alone are logically impossible tasks. Yet searching for a proof, be it positive or negative, is quite feasible, and the two searches lead us in different directions early on: squaring the circle begins with the [Hippocratic lunes](#), and culminates in [Lindemann's](#) 1882 proof, while trisecting the angle begins with the *Conics* of [Apollonius](#) and does not terminate until [Wantzel's](#) 1832 proof.



This example brings into sharp relief the distinction between analyticity in the sense relevant for linguistics, that is, the analyticity of meaning postulates, which are true by convention ([Grice](#), whose main contribution will be discussed shortly, called presuppositions ‘conventional implicatures’, a somewhat unfortunate terminological convention that we will not follow in this book), and the standard philosophical notions of analyticity, which relate analyticity to necessity in the logical sense. When we say that bachelors are unmarried men, we indeed say that to claim otherwise would be a violation of a conventionally agreed rule of language.

When we say, to take an example from [Putnam \(1976\)](#), that knowing something amounts to having evidence about it, this much is again an analytic truth. However, what constitutes the right kind of evidence for the impossibility of trisecting the angle is no longer part of the linguistic realm. Indeed, the layman can very well imagine that a clever series of moves with ruler and compass might amount to a valid trisection procedure, and cannot fathom how mathematicians can be so damned certain that nobody will ever come up with the right set of moves. The fact of the matter is that the relevant evidence, coming from Galois theory, is not part of the mental encyclopedia or the mental lexicon of non-mathematicians, for whom the lack of a known trisection procedure therefore appears as a contingent fact about the world.

We must agree with Putnam, who says that for a philosopher it is not a happy thing to ask whether a purported law of nature is analytic or synthetic, but our perspective is different: for the linguist, this is a perfectly reasonable thing to ask; it is just that our expectations of such laws ending up analytic are quite low. To continue further with Putnam, the linguistic evidence for some notion of time, embedded in tense markers,

temporal adverbials, aspectual morphemes, and so on, is irrefutable. But we can hardly expect to learn about the true structure of time, whether it is continuous or discrete at the Planck scale, or how it is intertwined with space, matter, and energy, from reverse-engineering the lexica of natural languages: clearly the appropriate tool for shedding light on such issues is physical experimentation and theory-making.

In contemporary semantics, the bundles of properties model is the basis of both the Semantic Web (there called Web Ontology Language, or OWL) and the influential WordNet approach to the lexicon (Miller, 1995). Following Quillian (1968), semantic networks are generally defined in terms of some distinguished links: IS_A to encode facts such as `dogs are animals`, and ATTR to encode facts such as that they are `hairy`. As we shall see, the genus and the attribution relation need not be encoded separately (we use link type 0 for both; see Section 4.5). Everything that appears as part of a lexeme is attributed (or predicated) directly, and IS_A means simply containment of the defining properties.

To preserve this nice and clean Aristotelian picture we need two technical tools. First, the notion of *essential* or *definitional* properties, since viewing every property of an object as analytic would destroy the descriptive power of the system. As Plato already notes in *Theaetetus*, to know a wagon does not require full knowledge of its hundred planks. Second, we need a notion of *default*, as opposed to strict, satisfaction. To treat the cases of commonsensical inferences that we take as the primary explicandum of lexical semantics, such as *friendly dogs don't bite*, the system described here uses primarily conjunction – disjunction, negation, and all forms of quantification are considered secondary phenomena (Kornai, 2010b). We rely not so much on implication, classical or relevant, as on abduction, which we take as the interpolation of silent elements. Consider a classic example from Parsons (1970), *enormous flea*, which means simply *flea*, *size*, *enormous*. It is a key task of the theory to elucidate how *size* gets interpolated, and why not use *flea*, *appetite*, *enormous* instead.

One thing to note is that *enormous* means ‘very big in size or in amount’. Not only do all competent speakers of English have this idea in their head, they *know* about each other that they do, and know that the other speakers also know that they do, and know that they know that they know, and so forth. In other words, the idea is part of the common ground. In all forms of communication, people very cleverly exploit this common ground (we will see some examples in the next section), and rely on the hearer’s ability to perform inferences based on mutual knowledge. This idea was first spelled out by Grice, who offered an analysis of the norms that determine what counts as a reasonable, appropriate conversational contribution. The first of these norms is the Maxim of Quality, *Try to make your contribution one that is true*. Hearers will, under most circumstances (but not, say, during a police interrogation), assume that the speaker is trying to adhere to this norm, and will interpret what they hear charitably. Next we have the Maxim of Quantity, *Make your contribution as informative as is required*, the Maxim of Relation, *Be relevant*, and finally the Maxim of Manner, *Be perspicuous*.

One case where these maxims often come into play is [scalar implicature](#), when some-



one says *Even Bill likes Mary*. What *even* contributes beyond the plain fact of Bill's liking Mary is some presupposition that Bill is a curmudgeon, and Mary is somehow an especially likeable person.

Exercise^o 5.22 (Green, 1973) Compare *Jane is a sloppy housekeeper and she doesn't take baths either* with *?*Jane is a neat housekeeper and she doesn't take baths either*. Explain the difference.

Exercise^r 5.23 (Ecclesiastes 3.1) *There is a time for everything*. Translate this statement into first order logic. Is this translation charitable? Try to capture the meaning that you think Koheleth had in mind. Is FOL sufficient for this? What other tools are needed for a charitable interpretation?

5.7 Pragmatics

So far, we have mentioned five major classes of phenomena that have a significant effect on the grammaticality and intelligibility of linguistic expressions: phenogrammar (visible constituent structure and word order, including the intonation pattern), tectogrammar (function–argument structure), the choice of words and word-forms (lexicon and morphology), context, and external knowledge. The last two (sometimes the last three) are often subsumed under the heading of *pragmatic* factors, and in many systems of grammatical description *pragmatics* is considered a separate field of study. The division of labor between these classes is not at all a trivial matter, especially as functions such as distinguishing questions from statements, or subjects from objects, are often carried out by different classes in different languages: questions can be marked by intonation, word order, grammatical particles, or some combination of the three; objects by adjacency (for example in English by immediately following the main verb) or by morphological marking (for example in Latin by accusative suffixation); and so on.

Comp When we decided that morphology and phrase-internal syntax were separate fields of study, we based our decision on the wide availability of computational systems that actually perform morphological analysis/generation and phrase-level chunking. Since there is no similarly worked-out computational system for pragmatics, we must subsume this area under semantics. This does not amount to a theoretical claim that pragmatics cannot possibly constitute a separate field of study on a par with morphology, syntax, and semantics, but we believe that those who wish to claim the opposite can only do so by exhibiting a working system.

Here we will discuss three well-known ranges of 'pragmatic' phenomena that any theory of semantics needs to account for: cooccurrence restrictions, implicature, and the dependence of linguistic meaning on external (non-linguistic) factors. We have already seen that syntactic problems can easily render utterances ill-formed: adding a plural marker when singular is called for, or exchanging the order of two words, can be quite sufficient to turn sentences anomalous. Ill-formed sentences are actually quite rare, not just in absolute terms (in absolute terms most sentences are extremely rare, with probabilities on the order of 10^{-25} and below) but also relative to well-formed sentences –

one can generally read through the Sunday paper (10^5 sentences) without finding a single ungrammatical example. Remarkably, we can easily construct a large class of sentences that are both anomalous and rare. Take ordinary sentences like *John derived the theorem* and *John slapped the boy* and exchange the objects to obtain *?John derived the boy* and *?John slapped the theorem*. Or take *John hit the rock* and *The rock hit John* and compare *John wanted to hit the rock* with *?The rock wanted to hit John*.

That the status of such sentences is anomalous is evident both from the reaction of listeners and from their low frequency. When we ask people what makes these sentences strange, they will explain that theorems cannot be slapped, one cannot derive boys, and rocks don't want things. Note that in doing this they use the same sentences in negative/conditional contexts that were anomalous in a positive context, but now without a trace of anomaly: sentences asserting the obvious such as *Nobody knows how to slap a theorem* are perhaps rare, but fully understandable, underscoring the point that it is not some syntactic violation that makes these sentences anomalous. In particular, the characteristic mental effort it takes for the listener to come up with some science-fictional context when trying to cope with derived boys and wilful rocks is completely absent when the context sets up the expectation of impossibility.

This phenomenon, which we already discussed from another perspective in Section 4.2, was first noted by Harris (1957), who simply called these *cooccurrence restrictions*, and left the issue of directionality open, as it is somewhat unclear whether it is the theorems that resist slapping or the act of slapping that cannot have theorems as its object. Chomsky (1965) opted for a directional treatment, and talked about *selectional restrictions*, the choice of term reflecting the thesis that it is the verb that can select its argument and not the other way around. The details of the causal mechanism that render such sentences anomalous are not evident, especially as similar cooccurrence restrictions are clearly at play even in cases where there is no verb to govern the selection, as in Chomsky's classic example of *green ideas*. The same holds for verbs like *have*, which seem metaphoric in sentences like *the walls have ears here*, even though there is no selectional restriction forbidding walls to be possessors (cf. *these walls have a peculiar color*) or ears to be possessions (cf. *foxes have pointy ears*).

Since anomalous sentences are rare, one may consider taking the easy way out and ignoring the problem entirely. But the true importance of cooccurrence restrictions is evident from the interpretation process of ordinary sentences as well, where we almost effortlessly filter out the anomalous readings in sentences like *The astronomer married a star*, giving preference to the straight reading 'movie star' over the metaphorical reading seen, for example, in *John is married to his work*. Since many words have more than one meaning, Frege's Principle of Contextuality is at work almost all the time during text understanding, and cooccurrence restrictions are among the most powerful means of disambiguating words and sentences. Since we know that people don't literally marry their work, people marry people, we have two choices: either we interpret the sentence metaphorically, substituting 'spends all his time with' for 'is married to', or we interpret the predicate literally at the expense of choosing the less literal meaning of *star* over the

one we would otherwise take as basic. But how do we conclude that people don't literally marry stars? What we need is a theory of semantics that can sustain *commonsensical inferences* of the kind we discussed in Section 3.1.

Exercise → 5.24 Derive *people don't marry celestial bodies* from *people marry people*. Do acids marry bases? Justify your answer.

As a more revealing example consider why John, taking a stroll in the park, upon encountering a large bulldog, will be somewhat reassured by the owner uttering the words *She is very friendly*. Clearly, there is an implication that the dog will not bite. To the extent the dog will do other friendly things such as jumping on and slobbering all over him, John is not at all reassured, but at least his fears concerning getting bitten are put to rest. How is this possible? Somehow, the owner has managed to provide information about the likelihood of the dog biting without even bringing up the subject of biting, so there is a *conventional implication* (Karttunen and Peters, 1979; Potts, 2005) that friendly dogs don't bite.

How is this implication made available to the hearer at the moment of parsing the sentence? One possibility would be that it is stored, like many pieces of conventional wisdom, as encyclopedic knowledge. This assumption brings two well-known problems. First, that the number of such everyday propositions is in the millions, and every attempt such as Cyc (Lenat and Guha, 1990) aimed at listing and cataloging them has been hugely incomplete. Second, and more important, that the speaker cannot know what is really stored in the head of the hearer, so uttering every such sentence would be a gamble, for what if the hearer knows that calcium carbonate is used for cleaning white gloves but fails to know that friendly dogs don't bite?

The traditional response is that *friendly dogs don't bite* is not just true but analytic, while *calcium carbonate is used for cleaning white gloves* is merely synthetic. In the eyes of many, Quine (1951) has, for all intents and purposes, demolished the analytic/synthetic distinction, but we hold with Putnam (1976) that the distinction is a good one, and not just for the reason adduced in Grice and Strawson (1956) that people make the call rather uniformly over novel examples. One may also agree with Putnam that

‘Bachelor’ may be synonymous with ‘unmarried man’ but that cuts no philosophic ice. ‘Chair’ may be synonymous with ‘moveable seat for one with back’ but that bakes no philosophic bread and washes no philosophic windows. It is the belief that there are synonymies and analyticities of a deeper nature – synonymies and analyticities that cannot be discovered by the lexicographer or the linguist but only by the philosopher – that is incorrect.

A key thesis of this work is that one philosopher's trash may just turn out to be another linguist's treasure – even if Putnam is right and the distinction between analytic and synthetic has no use in philosophy, it definitely has a use in linguistics and cognitive science. We are proposing a theory of meaning postulates that makes clear that the locus of our knowledge about friendly dogs not biting is not the encyclopedia, but rather the lexicon. Once this is established, the rest is easy: it is pragmatically unwise for the speaker to

make far-reaching assumptions about the mental encyclopedia of the hearer, but speakers may very well assume that hearers know what common words mean. In particular, it is part of knowing the word *bite* that this typically results in forced removal, by means of teeth, of some part of the object getting bitten: $x \text{ BITE } y \Rightarrow x \text{ REMOVE PART-OF } y$. We also know that beings value their bodily integrity and that removal of a part will diminish this integrity (this is postulated as part of the meaning of *remove*). From this, we can conclude they would not like being bitten even if this was painless. Further, we know that friendly behavior entails not harming things that are valued by those beings toward whom the behavior is directed, again as part of the lexical content of *friendly*.

Our goal is to derive *Friendly dogs don't bite* from premisses that are stored in the lexicon as part of the very definition of words. (Once precomputed, the implication may end up stored in the lexicon, just as perfectly predictable paradigmatic forms are often stored, see Pinker and Prince (1994), but this need not concern us here.) This suggests a larger program of refactoring Cyc and similar collections of facts into a small analytic core and a large, non-linguistically organized, synthetic encyclopedia. To a large extent this is just following common lexicographic practice, which puts a premium on brevity. Instead of providing a picture of the *yak* or an elaborate description of its shape and habits, the lexicon will simply say ‘large ox-like animal’ and make reference to {*Bos grunniens*}, thereby pointing the dictionary user explicitly to some encyclopedia where better information can be found. We collect such pointers together in a set **E**, and we will use curly braces to set them apart typographically from references to lexical content (see also Section 6.2). As a practical matter, whenever we feel the need to extend the definition to include such external knowledge, we will use [hyperlinks to Wikipedia](#).



Let us now consider the following sentence, collected as part of a larger effort to verify the parser of a larger computational system (Nemeskey et al., 2013) that procures railroad tickets based on natural language input:

Kaposvár	kérek	egy	ilyen nyugdíjas
N⟨SBL⟩	V⟨1SG.PR.IN⟩	Num Dem	N⟨NOM⟩
to Kaposvár	please	one	like pensioner

Frankly, the sentence is very hard to understand without knowing that the person who uttered it was standing in front of a railroad ticket counter. The demonstrative *like* is functioning as a filler ‘you know, like’ and the *pensioner*, not being in the accusative case, is unlikely to be the object of the request. All the same, the ticket clerk to whom the request was addressed showed not the slightest hesitation in fulfilling it, knowing full well that the object of the request was a *ticket*. The customer knew that the clerk would know (after all, the main reason for people to go to ticket counters is to buy tickets) and didn’t even bother to say so. However, he did say *pensioner* to ensure that he got a ticket that was discounted for seniors.

Here we will trace, at first informally, and later more rigorously, what a semantic system should do when placed in the role of the ticket clerk. We make no claim that what follows is in any way a realistic model of what goes on in the head of the human

ticket clerk (and, even if we did, it is not at all obvious how such claims could be tested) but we do take the methodological stance that a ticket clerk can differ from, say, a heart surgeon only in two respects that are relevant to semantics: first, they have access to different kinds of encyclopedic knowledge, and second, they are aware of their role in the world, so that the ticket clerk will hopefully not attempt to perform open heart surgery and the surgeon will not try to fulfill ticket requests. Our primary design goal is to create a system that is composed of maximally situation-independent, reusable parts corresponding largely to the words of a language. We certainly don't want the word *one* or *pensioner* to mean something different for the clerk and the surgeon; in fact, we hold that excessive reliance on specially crafted technical vocabulary is the major reason why the Semantic Web cannot get off the ground.

Comp

Many linguists would claim the sentence *To Kaposvár (please) one pensioner* is ungrammatical, obviously lacking a predicate, perhaps a subject, and definitely an object. However, an algorithm that produces a judgment of grammaticality for each string of words, while often presented as a central goal of syntax, is neither necessary for a semantic system nor sufficient. Semantics, very much including pragmatics, must be able to assign a meaning representation to this sentence, and to the extent that this does not rely on complex extralinguistic knowledge, should also be able to compute an action sequence that fulfills the request. This is not to deny that extralinguistic knowledge is part of our competence in the world, the cardiac surgeon relies on complex motor sequences to perform her job, and the ticket clerk knows how to fill in a form on a webpage that has slots for date of travel, starting city, destination, fare class, and so forth, before pressing a button that will make a computer print the right ticket.

Here we distinguish two classes of objects: one is the computer program that computes the price and prints the ticket, and the other is the clerk's internal model of this. Objects in the former class will simply be called *objects* or *things* and are largely outside our purview, for two main reasons. First, because we don't generally understand their workings (man-made objects like computer programs are of course understood quite well, but natural objects like the weather are not) and second, because they are non-linguistic in nature. Our goal is to understand language, and we can speak of everything, but it does not follow that to understand language we need to understand everything. The reason for this is that our model, for example the clerk's internal model of the ticket-printing and accounting program he uses, or the surgeon's internal model of the lancet, need not be faithful to reality. Knowing the truth is not a prerequisite for understanding. When Anna Karenina throws herself under the train, we may very well understand what this means and why she is doing it, but of course none of it is true. Following the philosophical tradition running from Aristotle to Locke and beyond, we can call internal models *ideas* or *concepts*, with the important proviso that both of these thinkers had perceptual adjectives such as *red* and *loud* that directly correspond to sensations, and physical objects such as *lancet* as their primary examples of simple and compound concepts. How this can be translated into a formal theory will be discussed in Section 6.2, together with the much more challenging question of attributing meaning to words like *absolute* or *when*.

Since interfacing with computer programs is of particular importance for computational theories of semantics, we discuss the issue here in some detail. The *concept* of a computer program we use is that of a function or automaton which produces, upon receiving specified inputs, some specified output or outputs. The prototypical case, well suited for describing, for example, the program used by the ticket clerk, is table (or database) lookup: what the human-internal model needs to contain is a specification of the possible inputs, and the fact that some of the time (hopefully most of the time) some output(s) will be produced. Accordingly, we model encyclopedic knowledge about computer programs as simple attribute–value matrices (AVMs) that have *keys* (typically printnames, as discussed in Section 6.1, but in any case some string describing the attribute), with slots for *matcher*, *default*, *value*, and *required*.

For example, the clerk’s internal model of the ticketing interface will have attributes like SOURCE, DESTINATION, CLASS, DATE, and DISCOUNT. Some of these may have reasonable defaults; for example the typical ticket is second class, so it may make sense to save time and supply a different value only when explicitly requested. Filling in some of the fields may be truly optional, but others are *required*. The reader not concerned with implementation detail may simply consider our AVMs as analogous to the kind of [forms](#) one can fill in on the web. In such cases the human, for example the ticket clerk, who does the form-filling is using their internal *matcher* to know that the prepositional object of *from* will fill the SOURCE slot and the prepositional object of *to* will fill the DESTINATION slot – for a computational system, we will use explicit pointers to code that does this matching. Readers implementing AVMs may wish to look at (possibly nested) [attribute-value lists](#) or [JSON](#) objects.

It is worth emphasizing that constructing such a *naive* model of programs is completely orthogonal to the goals of [programming language semantics](#). We are not at all interested in whether actual programs will terminate and produce a result, whether they behave according to their specification, or whether two programs produce equivalent results, just as we are not at all interested in the actual sequence of incisions a heart surgeon needs to perform. Our goal is to model the understanding that non-experts, and in fact every competent speaker of the language, will share: that programs take inputs, produce effects, and run on computers, mobile phones, and other devices with chips. The AVM view supports the abstraction that we can think of such programs as function calls, with separate keys (attributes) for each argument, but this is already beyond the ken of the non-expert, and the decision we make here is one that serves our implementation needs, as opposed to characterizing expert knowledge of the subject.

The same distinction between lexical and encyclopedic, analytic and synthetic knowledge must be kept in mind in every domain. What people know about *tickets* is that they are pieces of paper that give the holder permission to do things – in particular, *train tickets* give permission to board trains and travel some *distance* from *source* to *destination*. The word ‘destination’ can appear both as a pointer to a concept *destination* and as a key DESTINATION in an AVM.



By virtue of sitting behind the ticket counter, the clerk is aware of being part of a commercial transaction, a common concept shared between *buying* and *selling* whereby the seller obtains a *product* from the buyer, in exchange for a *price*. The expectation is that the person appearing on the other side of the counter is a buyer, and indeed people who just wish to obtain information need to make a special effort (which may well be rebuffed) to get themselves out of the buyer role.

While it is impossible to build an automated ticket seller without invoking some domain-specific (encyclopedic) knowledge, in the example we started with almost everything can be done by generic mechanisms. In particular, we will rely on default values such as setting the travel source to the location of the ticket office, and the travel date to today. In Hungarian, the travel destination is expressed by the sublative case, so the minimum utterance *Kaposvárra kérek* would already be interpretable as a request for a ticket to *Kaposvár*, by means of a tectogrammatical process called *linking* that matches the destination slot to the appropriately case-marked noun. (The second word, *please*, is required for politeness, and a human clerk would likely respond with a corrective *talán kérek* ‘perhaps you mean please’ were the buyer to omit it.) The matcher driving the process of linking the value *Kaposvár* to the key `DESTINATION` in the AVM already operates on morphologically analyzed strings – it is not the concept of `destination` that we use, but rather the sublative matcher that is associated to the key `DESTINATION` in the AVM.

Another mechanism that we will rely on is *spreading activation*, whereby *pensioner* is mapped onto a particular fare class, that of senior discount. There is no fare class called ‘pensioner’, but the lexical entry for this word contains the information that pensioners are elderly people no longer working. In this situation, the attributes of the ticket are already active, and one of these, *senior*, is close enough for the activation from *pensioner* to spread there, completing the parse. To understand how this works we need to recall some basic definitions from (hyper)graph theory.

Definition 5.3 A (directed) *graph* (X, R) is a set of *nodes* X equipped with a binary relation $R \subset X \times X$. If $\langle x, y \rangle \in R$, we say the graph has an *edge* from x to y . We say two nodes x, y are *connected* in n steps if $\langle x, y \rangle \in R^n$. In a hypergraph, hyperedges are generally still called *edges* but can contain more than two points, and there is no sense of any of these being the beginning or the end of the edge. A hypergraph is *k-uniform* if all edges contain exactly k points.

In Definition 4.5 we have already presented a more sophisticated notion of hypergraphs, where edges (and the graph as a whole) were equipped with a specific sequence of attachment nodes called $\text{att}(e)$ and $\text{ext}(e)$, respectively, as well as edge labels. Definition 5.3 is more skeletal: such hypergraphs can be obtained from the richer variety discussed in Chapter 4 by dropping all labels and all references to attachment or external nodes. This will of course make the key operation of hyperedge replacement undefined, but other operations, most notably the *spreading activation* that we now turn to, are already definable on these skeletal hypergraphs (and are trivial to extend to the more data-rich variety). Let us consider some subset S (often a single node) of X to be the

activation seed. For ease of notation, instead of S we may also consider I_S , the identity relation (where all edges are self-loops) restricted to S . The set of nodes directly reachable from S , $\{y | \exists x \in S \ xRy\}$ is denoted SR , the set of nodes directly reachable from SR is SR^2 , and so on.

Definition 5.4 A node y is *activated* by a seed I_S in n steps iff $x \in SR^n \wedge \forall x \in R^{-1}y \ x \in SR^{n-1}$, i.e. iff it is reachable from S in n steps and all its incoming edges are reachable in $n - 1$ steps.

Exercise° 5.25 Take $X = \mathbb{Z}$ and $\langle x, y \rangle \in R$ iff $y = x + 1$. Let S be $\{0\}$. If activation by S starts at time t , what nodes are active at $t, t + 1, t + 2, \dots$?

Exercise° 5.26 Take $X = \mathbb{Z}$ and $\langle x, y \rangle \in R$ iff $y = x + 1$ or $y = x + 2$. Let S be $\{0\}$. If activation by S starts at time t , what nodes are active at $t, t + 1, t + 2, \dots$? What if $S = \{0, 1\}$?

Exercise° 5.27 Take $X = \mathbb{Z} \times \mathbb{Z}$ and $\langle (x, y), (x', y') \rangle \in R$ iff $x' = x + 1$ or $y' = y + 1$. Let S be $\{(0, 0)\}$. Assume activation by S starts at time t ; what nodes are active at $t, t + 1, t + 2, \dots$?

In subsequent chapters we will refine these notions considerably, but for now it is best to think of X as the lexicon, with one node per word, and of R as *association* in the original psychological sense starting with [Wundt](#). Contemporary lexical networks such as [WordNet](#) use different types of edges (though these types do not nicely match up with the nominative/subject and accusative/object links that we use), but for now it is best to ignore this and to consider all edges to be in the same R . For example, in WordNet *pensioner* is linked to *old-age pensioner*, which (assuming things are linked to their components) is linked to *old*, which in turn is linked to *senior*, so the path from *pensioner* to *senior* has length 4.

External (situationally given) information is modeled by activating the lexical entries pertaining to the situation. In our example, these include the *buy*, *sell*, *train*, and *ticket* nodes; see Nemeskey et al. (2013). The activation of participant nodes is a particularly rich area, sometimes considered a part of pragmatics, sometimes of semantics, and sometimes even of syntax. Participants are generally signaled in the text by pronouns, and it is evident that *My father was killed by Brutus* means different things based on the identity of the speaker. Modern natural language processing (NLP) systems, such as the [Stanford Parser](#), have an entire module dedicated to this issue, variously known as *pronoun resolution*, *anaphor resolution*, or *coreference resolution*.



5.8 Valuation

In Section 3.5 we demonstrated the need for *valuations*, which we defined as mappings v from a state space S to some linear order $<$. In the simplest case, if the linear order has only two values, say $\top > \perp$, there is actually no need for the valuation mechanism, in that the inverse images of \top and \perp under v will just be two disjoint subsets S_\top and S_\perp of S , and everything we wish to compute with v is quite easy to compute using S_\top and S_\perp .

When the linear order is more complex, working with such level sets becomes awkward, and it makes more sense to work directly on the direct product of S with members of the ordering. All linear orders of interest can be embedded in the closed interval $I = [-1, 1]$ with its natural ordering, and when there are n valuations to be considered together, we will work with the direct product $S \times I^n$. We defer continuous valuations to Chapter 8, and first will use this mechanism in the special case where the linear order has only three elements, -1 ‘nonexistent’, 0 ‘fixed’, and 1 ‘active’, to model spreading activation. For this we consider in more detail the set that is the base X of an Eilenberg machine of the kind we use here to model semantics.

Definition 5.5 Given a lexeme l_i , we denote its 0th, 1st, and 2nd partition by l_i , l'_i , and ${}^{\prime}l_i$, respectively.

In terms of hypergraphs, a lexeme is a hyperedge, composed of a maximum of three (but typically only two) nodes corresponding the *partitions* we discussed in Section 4.5. We make the technical distinction between a lexeme and its head partition only when necessary, and use the same notation l_i for both.

Notation 5.4 To understand the system of left and right primes, consider a binary lexeme such as HAS that obtains, say, between John and his dog Rover. To make more readable the formulas that we will define precisely in Section 6.5, we follow the SVO (Subject–Verb–Object) word order of English, so that we can write JOHN HAS ROVER to make clear the identities of the owner (possessor) and the owned object (possession). With this in mind, the $'$ is used as a simple *plug-up* operator: HAS' means the object is plugged up, only the subject is considered, and conversely ${}^{\prime}$ HAS means the subject is plugged up, only the object is considered. In terms of graphs, HAS' is the node you reach from HAS by following the 1 (subject link, 1st partition), and ${}^{\prime}$ HAS the node you reach by following the 2 (object link, 2nd partition).

Definition 5.6 Given a collection of lexemes L , we collect the partitions of all lexemes together in a set P_L , and endow P_L with graph structure as follows. If l_i , say fox, has a pointer in its 1st partition to l_j , say clever, we run an edge from fox' to clever. Similarly, if the lexeme has a pointer in its 2nd partition, say DRINK, which subcategorizes for a liquid object, the edge will run from ${}^{\prime}$ DRINK to (the head of) liquid. We call the result the *detailed* definition graph of L to contrast it with the definition graphs introduced in Definition 4.11 on page 108.

Comp For an ordinary working vocabulary of 10^4 – 10^5 words, P_L will have perhaps a quarter million nodes, and perhaps two or three edges running from each node, often fewer. There could be encyclopedic associations of all sorts, for example, many people will know that that the taxonomic name of fox is *vulpes vulpes*; but these are not made part of the graph. Dangling pointers to lexical material, on the other hand, are not permitted; the graph must be definitionally closed. Counting both nodes and edges, we are still below a million data points, and adding in the three-level valuation ‘nonexistent/fixed/active’ would still make the entire data structure fit comfortably into three megabytes.

Definition 5.7 Given a collection of lexemes L , we define the *base* X as the detailed definition graph of L subject to a three-level valuation. We consider each lexeme, as an Eilenberg machine, to operate over this graph.

Let us now see how *spreading activation*, a process we already sketched in Section 5.7 for a railroad ticket application, can be more formally described using the apparatus developed here. Initially, a node or edge will have the value 0 if present in the lexicon, and -1 if absent. If we wish to assign some meaning representation to a sentence, such as *A sekki elapsed*, as discussed in Section 4.2, we begin with *activating* the content words that appear in the sentence, that is, by raising their valuation from 0 to $+1$. If a content word is missing from L , and thus its subgraph is missing from X , not all is lost: we can add in an isolated node with value $+1$. Note that this is hard to do in the standard Eilenberg machine formalism, where X is fixed once and for all: what we have to do is to define the original X as containing some number of initially empty nodes, and use up one of these each time a new word is encountered, the same mechanism that we propose for *acquiring* new words.

Another technical device we will need for parsing is that of the *construction*. A fuller discussion is deferred to Section 6.2; here we restrict ourselves to a very simple example, intransitive sentences in English, given by the pattern $\text{NP} \widehat{\text{V}}$, where NP is a noun phrase such as a proper noun like *John* or a Det N combination like *a dog*. We use an arch $\widehat{\text{ }}$ in the pattern to signify linear adjacency. Lexical items can be used to fit into pattern slots, and (partially or entirely filled) patterns can be fit to each other. All of this is standardly formulated in terms of CFGs (see Definition 5.1 on p. 125), but here our interest is in implementing what roughly amounts to CFG parsing in a network of Eilenberg machines by a valuation mechanism that activates only a very small portion of the lexicon at any given time.

The three key operations required for this are placing one entry (hyperedge) in a partition of another, unifying material on the same partition, and activating a new hyperedge. We discuss each in turn, but note that they need not play out in this particular order in any parse. By placing one entry x in a partition y , we simply mean running a new edge, or activating a preexisting one, from y to x . This is definite in regard to y , which is an elementary node (not a hypernode), but indefinite in regard to x , which may be a single elementary node, but may also be the head of any kind of larger structure. By unification of x with y , we mean running an IS_A link (or activating a preexisting one) from the head of x to the head of y . Unification is not a symmetrical operation; for symmetry (which would imply Leibnizian identity, all properties shared between x and y), we would also need to identify y with x . Finally, activating a new hyperedge means raising its valuation from -1 (nonexistent) or 0 (existent but inactive) to 1. It is this activation operation that we model by a relation in an Eilenberg machine.

Definition 5.8 Given a machine base X which contains an edge $x \rightarrow y$, we say that the relation $A_{xy} \subset X \times X$ is the *activation* of this edge if the value $v(xy)$ of the edge $x \rightarrow y$ on the right-hand side of the relation is $+1$, with all other valuations unchanged.

At any given state of the X machine, we can ask what nodes and edges of X are activated (have valuation +1). We consider a *node* activated if all incoming edges are activated, and once a node is activated, we can spread this to all its outgoing edges. Such spreading is automatic and immediate from the heads to the rest of the nodes, but will require a spreading step (assumed to take a time tick) for links across lexemes.

Here we assume, for the sake of illustration, that the word *sekki* is unknown to the system, but the word *elapse* is known, as would be the case for most speakers of English. The pattern-matching system tells us three things: that *elapse* is in the past tense, that *sekki* immediately precedes it, and that *a* immediately precedes *sekki*. Since the nodes for *a* and *sekki* are active, and one immediately precedes the other, we get the preexisting $\text{Det} \widehat{\text{N}}$ pattern activated. With this, the information stored in its head partition that this is an NP is immediately activated.

Next, since both the NP and *elapse* are active, and the NP immediately precedes the verb, the $\text{NP} \widehat{\text{V}}$ pattern is activated. Since in English the nominal preceding the verb is its subject, we place *sekki* (or, more precisely, the entire noun phrase *a sekki*) in the 1st partition of *elapse*. At this point, all initially active portions of the graph are now in one connected active hypernode, and we only need a bit more cleanup. Because we know that *elapse* takes a subject that is a time interval, the 1st partition of *elapse* exists in the lexicon as containing a link to the concept `timeinterval`: in the cleanup step we need to unify this with the concept *sekki* that appeared in the same partition as a result of the parse process, and we conclude that *a sekki* is a time interval.

The spreading concludes either because there is no more active material to attach, a situation that may change when a new sentence is heard, or because, as in traditional CFGs, the $\text{NP} \widehat{\text{V}}$ pattern is marked for termination (by having the distinguished symbol S appearing in its head partition); we will not distinguish these two cases here. In a linguistically more detailed system we would want to take account of pronoun resolution, [anaphora](#), and [sluicing](#), and in a cognitively more realistic system we would no doubt want to make some provision for activity decaying after a while, but here our goal is simply to show that the machine mechanism is capable of parsing (assigning semantic representation to a string of words) with nothing more than assuming that constructions are part of the lexicon (as they must be, on any account).

Next we turn to a key technical notion of [artificial general intelligence \(AGI\)](#) that is subsumed under our notion of valuation, the idea of a *utility function* u that assigns some numerical ‘utility’ to each state. If we conceive of AGIs as transducers operating on inputs coming from the external world which can produce state changes and outputs, an AGI will, all other things being equal, choose the result state and output with the maximum utility. As we shall see in [Section 8.2](#), the matter is considerably more complex than what a single utility function could model, but for now let us stay with this simplified picture and call $u : S \rightarrow I$ the *desirability* of states.

Let us place close to the -1 end of the scale the highly undesirable states, such as those associated with bodily harm, and close to $+1$ the highly desirable ones, such as those associated with physical pleasure. From any machine state, certain other states are



reachable by transitions, and we can ask the question of whether a more highly valued state is immediately reachable. This becomes interesting when the only transition from an undesirable state, such as being thirsty, to a more desirable internal state goes through some external action, such as drinking water. Needless to say, there may be other factors at play: there may be a predator between us and the spring and it may be far more desirable to remain thirsty than to fall prey to it.

Exercise[†] 5.28 In a famous passage, [Hume](#) states “Reason is, and ought only to be the slave of the passions, and can never pretend to any other office than to serve and obey them.” (*Treatise* 2.3.3.4, p. 415) Design a machine where the passion are valuations, and the deductive machinery simply seeks the highest-valued state accessible in two steps.



5.9 Further reading

The standard book-length introduction to morphology is [Matthews \(1991\)](#). For a more skeletal introduction to the subject, see Chapter 2 of [Fromkin, Rodman, and Hyams \(2003\)](#) or Chapter 4 of [Kornai \(2008\)](#). For a computational implementation of non-concatenative morphology, see [Cohen-Sygal and Winter \(2006\)](#). For a detailed study of clitics, see [Anderson \(2005\)](#), and for a more detailed description of how glosses are to be written see Section 8 of the [LSA style sheet](#) or the more detailed [Leipzig glossing rules](#). The rationale behind the Penn Treebank tagset is presented in [Marcus, Santorini, and Marcinkiewicz \(1993\)](#), but the tradition of using POS tags is as old as the study of grammar. The names *function word* and *content word* were introduced by [Fries \(1952\)](#) but the idea is standard in classic Chinese grammar, where these classes are known as *xuci* ‘empty words’ and *shici* ‘full words’.



In the scholastic tradition, the words of a declarative sentence were divided into three classes: subject (what we would call a noun or noun phrase today), predicate (what we would call a verb or verb phrase), and modifiers (adjectives, adverbs). Everything else was treated as having no category (today we would say they belong in singleton categories; see Definition 4.6). [William of Sherwood](#) devoted a book, *Syncategoremata* ([Kann and Kirchhoff, 2012](#)), to the study of elements like the logical connectives, modals, quantifiers, etc. that play a central role in the classical theory – we will return to these in Section 7.3.

The study of ‘tactics’ in the modern sense is best understood in terms of a vocabulary Σ and a language (stringset) $F \subset \Sigma^*$ of grammatical forms. The key methodological advance, due to [Wells \(1947\)](#) (for a modern account, see [Miller \(1986\)](#) and [Miller \(1987\)](#)), was to consider substitutions of a string β by some other string δ in a context $\alpha_ \gamma$ to investigate whether $\alpha\delta\gamma$ is grammatical ($\in F$) if $\alpha\beta\gamma$ is. This method associates a (binary) tree, called the *parse tree* or *constituent structure* with a string.

Within the AI literature, the Principle of Contextuality was seen primarily as the problem of [word sense disambiguation](#): how come we know that in *John kicked the ball* we have *ball*₁ ‘sphere’ rather than ‘*ball*₂ ‘formal gathering for social dancing’ (Hayes,



1976). Only in the past five years has a significant advance been made; see, for example, Reisinger and Mooney (2010) and Chen, Liu, and Sun (2014).

For the medieval approach to the perfect language, see Eco (1995). Here we have compared 4lang with UD, as we see this as the most prominent dependency framework for many years to come, but there is a far greater variety of computationally oriented dependency/valency theories; see Somers (1987). For direct objects, see Plank (1984), for indirect objects see Kornai (2012).

The definition of hypergraphs is far from uniform in the literature. Definition 4.5 follows Drewes, Kreowski, and Habel (1997), and Definition 5.3 follows Berge (1989). Yet another definition, where hyperedges can point to other hyperedges, is discussed in Wikipedia; this is used in Kálmán and Kornai (1985) and Kornai (2018).

The literature on knowledge representation is vast – a good starting point is Brachman and Levesque (1985). Spreading activation originates with Quillian (1968); the early work is conveniently summarized in Findler (1979). The fact that we don't shunt pragmatics into a separate component of the system does not in any way invalidate the rich literature on the subject, except perhaps for the definitional effort aimed at delimiting the field. See in particular Gazdar (1979), Sperber and Wilson (1996), Wilson and Sperber (2012), and Asher and Lascarides (2003) for much insightful discussion.

For the philosophical approach to pronoun resolution, see the entry [Indexicals](#) in the Stanford Encyclopedia of Philosophy. The early NLP work in the area is summarized in Hirst (1981); for a more contemporary introduction see Section 21.3 of Jurafsky and Martin (2009). The case when the antecedent is not a nominal but a temporal is discussed in Partee (1984). The classic paper on purely syntactic conditions on anaphora is Postal (1969); see also Dalrymple (1990) and Groenendijk and Stokhof (1991). For sluicing, see Merchant (2001).

To give a fuller theory of action based on rational forethought (planning) would require a whole other book, such as Ghallab, Nau, and Traverso (2004). One area where this research is in constant contact with practical applications is game design; see, for example, [GOAP](#).



Lexemes

Contents

6.1	Lexical entries	172
6.2	Concepts	175
6.3	Lexical categories	178
6.4	Word meaning	182
6.5	The formal model	191
6.6	The semantics of lexemes	194
6.7	Further reading	197

Virtually every task we can conceive of requires both stored knowledge and the ability to apply it, and semantic processing is no different. Ideally, we would want a system that initially has only a bare minimum of proleptic knowledge and acquires its stored knowledge on the go. As a first step toward this goal, in this chapter we will study the mature knowledge system that is acquired by normally developing humans by the age of fourteen. As discussed in Chapter 3, all machine learning algorithms operate by means of selecting hypotheses from a pre-set *hypothesis space* based on fit with empirical data or some broader fitness criterion. The first order of business is therefore to delineate the hypothesis space. We have already made a broad distinction between *linguistic* and *encyclopedic* knowledge, and here we see one additional reason for distinguishing the two: the linguistic system is nearly at the adult level of competence by the age of four and is effectively closed to further development by puberty (Pinker, 1994), while encyclopedic knowledge can keep on growing throughout childhood, puberty, and adulthood.

In the preceding chapters, we have developed almost all the formal tools we will need to describe the process of making sense of linguistic input. In particular, the basic units called *lexemes* were introduced, and to some extent motivated, in 4.5. In this chapter our goal is to see how these formal objects relate to the more informally developed traditional notion of a lexical entry (often called a *lemma*), and what can be done with them. Our lexemes are machines (see Definition 4.4), and they will be operated on by various phenogrammatical and tectogrammatical rules. The converse is not true: it is not at all the case that every system of machines subject to some formal grammatical

CogSci

Ling



operations amounts to a model of sense-making, and by the end of this chapter we will narrow down this huge hypothesis space considerably.

Ling

In 6.1 we provide a bird’s-eye view of the main fields used in dictionaries. Only some of these fields play a direct role in computational systems aimed at natural language understanding, but we provide a comprehensive overview to see which do and why. In 6.2 we describe what we mean by [concepts](#), and how we can use concepts for representing linguistic knowledge. In 6.3 we turn to another pivotal element of lexical entries, the *category* or *part of speech*. We come to the key issue of semantics, which is to capture word meanings, in 6.4. In 6.5 we turn to the relations that lexical entries can have to one another, and in 6.6 we discuss the model theory of lexemes, an issue of great importance, especially for CVS models, which, at first blush, seem to lack any.



6.1 Lexical entries

The collection of lexemes in a given language is called its *lexicon*. Here we will compare our techniques with those used in traditional [lexicography](#). We begin with a sample taken from [Merriam-Webster](#), shown in Fig. 6.1



The screenshot shows the Merriam-Webster dictionary entry for 'has-tate'. At the top, it displays the word 'has-tate' with a speaker icon, the part of speech 'adj', and the phonetic transcription '\has-,tāt\'. Below this, there is a section titled 'Definition of HASTATE' with a '+1' icon and a 'Like' button. The definition consists of two numbered items: '1 : triangular with sharp basal lobes spreading away from the base of the petiole <hastate leaves> — see LEAF ILLUSTRATION' and '2 : shaped like a spear or the head of a spear <a hastate spot of a bird>'. A link 'See hastate defined for kids »' is provided. The next section is 'Origin of HASTATE', which states 'New Latin *hastatus*, from Latin *hasta* spear — more at YARD' and 'First Known Use: 1788'. The final section is 'Rhymes with HASTATE', listing 'abate, ablate, adnate, aerate, age-mate, agnate, airdate,'.

Fig. 6.1. *hastate*. By permission. From [Merriam-Webster.com](#) ©2017 by Merriam–Webster, Inc. <https://www.merriam-webster.com/dictionary/hastate>

Every lemma begins with a [headword](#), in our case *hastate*. We need to know something about English morphology to key in on *river* from the plural *rivers*. This looks trivial, but for foreign speakers at least, it is not always obvious where the stem ends and the suffixes begin (consider English *voted*, with stem *vote*, but *potted*, with stem *pot*), just as for speakers of English it may not be trivial to see the word *tetigi* and know that the place in the dictionary to find it will be under *tango* – clearly there is value to automatic [stemmers](#) which reduce different forms to the same stem.



Here there is something of a split between natural language processing applications, where any stem, such as that returned by the [Porter stemmer](#), is good as long as it is returned consistently, and the practice of generative linguistics, where we look for an underlying form. Our compromise solution will be to simply speak of the *printname* of a lexeme, but without taking up the lexicographic task of providing orthographic guidelines, for example for preferred points of hyphenation (*has-tate* rather than *ha-state* or *hast-ate*), capitalization, spelling variants, etc., a subject not without intrinsic interest, but entirely outside the purview of this book. The printname, which is simply a string used for recognizing a machine, and all other external pointers connected to the lexeme as a whole are stored in the zeroth partition. In linguistic theory, lexemes are ⟨form, meaning⟩ pairs, but in this book we have little to say about forms. One reason for this is that we can simply take forms to be external pointers to audio files, as most online dictionaries do (see the loudspeaker icon in the head bar of Fig. 6.1).

Another, more structural, reason is provided by the division of labor within linguistic theory. Deriving a *surface representation* such as `\`has-,tāt\`` that would be a suitable input for speech synthesis is a task that the generative theories of [phonology](#) and [morphology](#) (see Section 5.2) are jointly responsible for. Since these theories can be restated using the standard theory of FSTs (Frank and Satta, 1998; Karttunen, 1998), and FSTs are special cases of machines, in principle we could use the machines to drive speech synthesis and, conversely, we could feed the output of a speech recognizer directly into low-level machines. In practice, we will rely on the many independently developed open source morphological analysis and generation packages (some of which actually use FSTs under the hood), in particular the [HunSpell/HunStem/HunMorph](#) family of word-level tools.

There is one more field deemed central enough in lexicography to appear in the head bar, the part of speech (POS) or lexical category, in this case **adj** (adjective). We discussed this in general terms in Section 5.2, introduced a specific proposal in Section 5.4, and will return to the matter again in Section 6.3. Generally a lexical entry will have both a form (or printname) and a lexical category, but in special cases, one or the other can be missing. In particular, we often see the need for *zero morphemes* such as in the English singular, which signals paradigmatic contrast (in this case, the stem being non-plural) without any overt phonological material. We will also need *singleton categories* such as that of the existential *there*, as in *There was widespread rioting*, which does not share its distributional class with any other morpheme of English and therefore can do without a category name. Most lexical entries will of course have both a non-zero phonological form and a non-trivial class associated with them, and our chief interest is in these *even when they are omitted*. Since well over 90% of words in a typical dictionary are nouns, one can save considerable space by omitting the category N and providing category information only for the remaining words – we say that the N is *supplied by a lexical redundancy rule*. In a printed dictionary with 100k lemmas this one rule saves about 200k characters, or about 30 pages. The practice of using redundancy rules remains widespread both in practical lexicographic work and in linguistic theory, but we



Comp



separate the issue of creating a lexicon from the issue of compressing it. Unless stated otherwise, everything we say here applies to the uncompressed lexicon.

Another important piece of knowledge that is often listed, but not always, is the [etymology](#). It is truly remarkable that even words we may think of as characteristic of high technology, such as *rocket*, can go back, in this case via [Old French](#) *rocquet* ‘head of a lance’, to an Indo-European root *ruk* (Watkins, 1985). When we omit etymologies, we do this not so much because we have a satisfying formal theory, as was the case with phonology, but rather because we believe that we can manage without representing such knowledge. Ordinary people generally don’t know the etymology of words and, even more important in light of our discussion of friendly dogs and glove-cleaners (see Section 5.7), *the speaker cannot reasonably assume that the hearer will know*. This is not to say that philosophical, political, or even everyday discussions will never rely on etymology, for they often do, but in such cases the etymology must be overtly stated, rather than simply assumed as common background.

Another piece of lexical knowledge, again without a well-developed formal theory, but with a great deal of practical utility, and clearly accessible to all competent speakers of a language, concerns the *stylistic value* of words. Certain words and longer expressions are taboo, and speakers generally omit them from discourse or replace them by [euphemisms](#). We will simply assume that their taboo or impolite status is listed as part of their lexical entry, just as the substantive parts of their definitions are. Other words or, more often, certain senses of words, belong in the jargon of a profession, and even people outside the legal profession will know that words like *nuisance* mean something else to a lawyer. By the same logic that we applied to etymology, we may not assume that other people will know the legal meaning of *nuisance*, but we can very well assume that they know of the existence of such a specialized meaning. Some lexicographers use stylistic values or [semantic fields](#) to distinguish such meanings from one another, but here we take a simpler approach and use separate lexical entries, so as to distinguish, for example, *bishop*₁ ‘high-ranking church official’ from *bishop*₂ ‘chess piece’, and take the intriguing notion of a semantic field as part of the explicanda rather than part of the pre-defined technical apparatus (see Section 6.4).

A peculiarity of Merriam-Webster is the listing of rhyming words – this is clearly useful as a pronunciation aid for those unwilling to learn the transcription system used in providing the surface representation. Other specialized dictionaries may have fields, or even indexes, for different kinds of information, for example, crossword dictionaries are indexed by the number of characters in the word. Lexicographic tradition is split on the use of pictures: there are some, such as the [Culturally Authentic Pictorial Lexicon](#), which contain little else beyond printnames and an image, while others mix the pictures with more traditional lexicographic information. The monumental [Webster’s Third](#) employs pictures for fewer than 0.5% of the lemmas, generally for concepts such as [opera slipper](#) that would require a complex description that would take up more space on the printed page than a simple sketch. On the whole, picture dictionaries are useful learning aids for small children, but for later developmental stages their value is strictly limited by the

fact that many of the concepts we rely on simply have no visual image associated with them. It is easy to picture a *radio*, but how do you picture *radioactivity* or *insufficient*?

Exercise° 6.1 Select a monolingual dictionary and describe the fields it has using AVMs. Which fields are required within that dictionary?

Exercise° 6.2 Select a bilingual dictionary and describe the fields it has using AVMs. What are the essential differences compared with monolingual dictionaries?

6.2 Concepts

What are concepts? We begin with a simple example that we have already discussed informally in Section 3.3, that of a *dog*, which we defined as *four-legged, animal, hairy, barks, bites, faithful, inferior*. For reasons of expository convenience we started with a lexeme that is a single atomic form (monomorphemic; see Section 4.5), but what we will have to say here applies equally well to lexemes that are composed of several forms, such as the Tuscarora word for goat, *ka-téskr-ahs*, NEUT.SUBJ-stink-IMPERF, literally ‘it stinks’. Since the definition of *goat* will also be a compound form like *four-legged, animal, hairy, bearded, horned, stinky* we must ask what makes this a lexeme, at least in Tuscarora, where the expression *ka-téskr-ahs* is clearly derivable with the compositional meaning ‘it stinks’. The answer is that there is a conceptual unity, manifest both in the availability of a vast amount of encyclopedic knowledge about [goats](#) and in the simple fact that many languages (not just English) have monomorphemic lexical entries for this concept.

There are two important lexicographic methods that stem from the above considerations: first, that *every morpheme corresponds to some concept*. There are cases of [homonymy](#) where a single sequence of sounds, such as *bore₁* ‘to drill a hole’ and *bore₂* ‘a person talking about uninteresting things’, corresponds to two entirely unrelated meanings, and there are cases of *polysemy*, where we feel the meanings are different yet closely related, as with *bore₁* and *bore₃* ‘the diameter of a hole’. True (non-polysemous) homonymy is relatively rare, especially if viewed from an information-theoretic perspective, counting the number of bits required for disambiguation, since the frequency distribution of the different senses is generally rather skewed. This principle does not mean that all concepts are obtained from single morphemes; in fact, the vast majority of polymorphemic words also correspond to single, unitary concepts (again modulo homonymy and polysemy). As an example consider [testtube](#), a glass vial of roughly tubular shape, which is not even a tube (one end being closed off), and is used in many situations that do not involve testing.

This brings us to our second lexicographic principle: *linguistic expressions with encyclopedic knowledge associated with them are concepts*. It follows from this principle that, for example, every Wikipedia page describes a concept (or more than one, in which case Wikipedia uses [disambiguation pages](#)). Thus, [The battle of Jena](#) is a single concept, and it can participate *as a single concept* in more complex linguistic expressions, for example, as the location or cause of other events as in *The battle of Jena further*



enhanced Napoleon's air of invincibility. That encyclopedic knowledge is a source of static concepthood should come as no surprise: to the extent that some newly formed concept has no encyclopedic knowledge associated with it, we see no need to store it in the lexicon. For example, a *yellow Volkswagen* is just a Volkswagen that is yellow; it inherits all other knowledge we have about it, for example that it has four wheels, from being a Volkswagen.

The idea that encyclopedic pointers are stored in machines is no different from the central defining property of lexemes, namely that they contain pointers to other lexemes, as discussed in Section 4.5. In fact, in linguistics this observation is standardly elevated to another principle of lexicography, namely that *no expression with purely compositional meaning should be stored in the lexicon*. In the light of modern psycholinguistic research showing that precomputed forms may also end up stored in the lexicon (Pinker and Prince, 1994), we do not adhere to this principle rigidly, but rather treat it as a matter of caching strategy, spending memory to save processing time.

As we have already discussed in Section 3.8, it is highly desirable to have concepts serve as translation pivots. This requirement again has several implications for lexicographic practice which should be made explicit, especially as there is a common misunderstanding that concepts are universal (independent of language). To be sure, we don't particularly expect the English concept of *goat* and the Tuscarora concept of *ka-téskr-ahs* to differ in any significant way, just as we expect English *three* and French *trois* to mean the same thing. Yet there could be subtle differences in the encyclopedic knowledge associated with these words: it is quite feasible that the prototypical breed recognized as a goat by the Tuscarora is different from the one considered prototypical in Northern Europe. Equally importantly, there could be significant differences in the culturally inherited knowledge as well: in standard European (Christian) imagery, goats are sinful and sheep are innocent (going back to Matthew 25:31), while there is little reason to suppose that unconverted Tuscarora have the same concept. Since cultural knowledge often gets embedded in the lexicon (just as we had the *dog* being *inferior* in the example we started with), it is possible that the English definition of goats should include *sinful* rather than (or in addition to) *stinky*. Exactly what gets and what does not get into the definition of a given concept in a given language is a matter we defer to the next section, but we emphasize here that the case where two words in different languages carry the exact same meaning (i.e. correspond to the exact same concept) is rather rare. Arguably, even seemingly absolute concepts such as '3' are reflected differently in languages as close as English and French, since English lacks the expression ⁰*household with three* while French of course has *ménage à trois*. As we shall see shortly, even if the definition of English *three* and French *trois* is the same '3', the associative networks will differ because of the presence of a lexical entry in one that is absent from the other.

Exercise^o 6.3 Provide examples of concepts from other languages that have no monomorphemic equivalent in English.

If concepts are not in perfect alignment across languages, translations are often going to be imperfect, but we don't consider this a fatal flaw of the theory, in that such

imperfections are evident to anyone speaking more than one language. The lack of perfect correspondence can in fact be used in a positive fashion in distinguishing different senses of linguistic expressions: if from one language, the source S , some expression w can be translated into two expressions u and v in the target language T , this is generally sufficient to consider w to be composed of homonyms w_1 and w_2 (unless of course u and v are synonyms to begin with). As an example, consider Hungarian *állni*, which is generally translated as ‘stand’ as in *The boy stood up* or in *The theatre stands on the corner of Broad and Main*, but not in *a gép áll*, which must be translated as ‘the machine is not running/working’ or ‘the machine is standing still’. We conclude that Hungarian *áll* is composed of *áll*₁ ‘stand’ and *áll*₂ ‘be at standstill’, and possibly of other concepts as evidenced by further different translations. In a machine translation task we still have the homonym selection or [word sense disambiguation](#) problem, but the formal status of concepts is homogeneous across languages, even if the concepts themselves are not invariant. The [4lang](#) conceptual dictionary captures the basic sense that is common to all four translations; compare, for example, 1381 ko2nnyu3 A light levis lekki with 739 felny N light lux slwiatllo. (The letter–number digraph code of the original file has been retained for ease of grepping.)

Notation 6.1 To fix the notation, we need to recall and refine some terminology from Section 4.5. A simple *lexeme* is a machine that has a small active base set which contributes two or three *partitions*, numbered 0, 1, and if present, 2, to the entire base X to which all lexemes contribute (see Definition 5.7 on page 166). This formalizes the intuition, much emphasized in the linguistics literature on word meaning since Saussure’s [Course in General Linguistics](#), that no word meaning exists in isolation, but rather only in a system of contrasts or *oppositions* to other words that also acquire their meaning relative to the entire network, a matter we have already discussed in Section 2.7.

Simple lexemes never have further partitions, so ditransitive and higher-arity verbs like *give* or *promise* will be given by complex lexemes. In addition to the base, which is shared across all lexemes, the full definition of a lexeme machine will include also a *control* FSA and a mapping from the alphabet of the control to the monoid of relations over the base. We can use the control mechanism to handle function–argument relations (tectogrammar, see Section 4.6), and a transductive pattern substitution mechanism (formalized as hypergraph replacement; see Section 4.4) to handle the phenogrammar.

Based on these components, we can now define the *complexity* of a word using standard notions of complexity that apply to finite automata and graphs. Since the complexity of FSA and FSTs is standardly measured by the number of states they have, we have the following.

Definition 6.1 The *control* or *phenogrammatical* complexity of a lexeme is given by the number of states s in its control.

The base X is shared across all lexemes, but the number of links to the base from the hyperedge can differ. To make this more precise, we distinguish links from the head (0th partition), let’s say there are e_0 of these, links from the 1st (subject) partition, let’s say



there are e_1 of these, and links from the 2nd (object) partition, let's say there are e_2 of these.

Definition 6.2 The *head* complexity of a lexeme is given by e_0 , the *subject* complexity by e_1 , and the *object complexity* by e_2 .

Putting all these together with weights $0 \leq \alpha_i \leq 1$ such that $\sum_0^3 \alpha_i = 1$, we obtain the following definition.

Definition 6.3 The $\alpha_0, \dots, \alpha_3$ *complexity* of a lexeme (or larger representation) is defined as $\sum_0^2 \alpha_i e_i + \alpha_3 s$, where $\alpha_0, \dots, \alpha_3$ is called the weighting scheme defining the measure.

By choosing the weighting scheme appropriately, we can emphasize the phenogrammatical complexity or various aspects of the tectogrammatical complexity. Of particular interest are the schemes (0, 1/3, 1/3, 1/3), which we used for generating Fig. 1 on p. ??, and (0, 1, 0, 0), which measures pure IS_A hierarchical complexity.

Exercise^o 6.4 Certain 'collective' concepts given in Appendix 4.8 such as *color* obtain their definition by combining a genus *sensation* and generic information such as *light* with specific examples such as *red IS_A, green IS_A, blue IS_A*. How does that affect their definitional complexity?

We emphasize here that lexical entries are not restricted to words or set phrases such as *over the counter* but also include larger constructions with empty slots. Of particular interest are the phrasal verbs we discussed briefly such as *call NP up* or *take NP to task*, which are tectogrammatically similar to ordinary transitive verbs such as *exclude* in requiring both a subject and an object, but phenogrammatically different in that their object comes in the middle of the phonological material not at the end. Some patterns, such as the Noun $\widehat{\text{Preposition}}$ $\widehat{\text{Noun}}$ construction seen in *day after day, dollar for dollar, \dots* (Jackendoff, 2008), are even more devoid of phonological material, yet we need to list them in the lexicon since they have their own meaning. By placing such constructions in the lexicon, we actually arrive at a form of linguistic theory Karttunen (1989) called *radical lexicalism*, the idea that the only thing that differentiates one language from another is the content of its lexicon.

6.3 Lexical categories

Part of speech labels (POS tags) are a prominent feature of standard dictionary entries for two reasons. First, they provide a strong indication of the word forms that are suppressed in the lemma. If the POS is V (verb), as in *wait*, we know that *wait+ing* is a legitimate form in English, but if the POS is N, as in *faith*, we know we cannot expect the form **faith+ing*, even though the combination would have a reasonable meaning (one that we happen to express differently, using a **light verb** to carry the verbal aspect: 'having faith'). Second, POS tags go a long way toward defining the syntactic (phenogrammatical) potential of words.



In Definition 4.8 (page 99) we simply equated classes of the morpheme-level syntactic congruence with lexical categories, but here we need to go a bit further in order to harmonize our method with standard lexicographic practice. First, we have to do something about those morphemes that take no inflection. *Indeclinabilia*, as these elements are called in traditional grammar, would all fall in the same category of ‘particles’ based on their within-word distribution, and it is only on the basis of their across-word distribution that we can distinguish them. Typical examples include conjunctions, quantifiers, and other function words. Since cross-linguistically such elements often appear as affixes (bound morphemes), it makes sense to use them in the definition of lexical categories in the same way, and we will indeed say that two forms u and v are considered to be in the same category iff their distribution is the same relative not just to affixes but also to function words. From the perspective of the more traditional morphotactic notion of lexical categories that we started with, this amounts to saying that if two expressions u and v belong in different congruence classes, meaning there is some context α_β that separates them, one of $\alpha u \beta$ and $\alpha v \beta$ being grammatical while the other is ungrammatical, we can also witness this separation by choosing contexts α' and β' that are composed entirely of function morphemes.

While in any language there are dozens, possibly hundreds, of function words, generally each forming their own singleton lexical category, it is the content words that take up almost the entire lexicon. Children by the age of three generally have vocabularies in excess of a thousand words, abridged learner’s dictionaries have 40,000 lemmas or more, and comprehensive dictionaries have several hundred thousand. Somewhat surprisingly, all this lexical wealth falls into only a handful of *major categories*, in particular noun, verb, adjective, and adverb. As we discussed in Section 5.2, there are bound content morphemes, roots, which generally don’t participate in the system of major categories, and the claim is made time after time that the content vocabulary of certain languages is composed entirely of roots. Though Tuscarora is also mentioned in this regard, the poster boy of such claims used to be Eskimo, where Thalbitzer (1911) claimed that

In the Eskimo mind the line of demarcation between the noun and the verb seems to be extremely vague, as appears from the whole structure of the language, and from the fact that the inflectional endings are, partially at any rate, the same for both nouns and verbs.

That this is a myth, comparable to the “Great Eskimo Vocabulary Hoax” of Eskimos having many different words for snow (Pullum, 1989), has been conclusively demonstrated by Sadock (1999), who writes:

In all [Salishan or Nootkan] languages there is a sharp formal contrast between two classes of roots, stems, and words that is absolutely central to the inflectional, derivational, and syntactic systems of the grammar. Furthermore, this two-way formal distinction correlates directly with the same cognitive complexes that characterize the noun–verb distinctions in European languages. Thus the words for ‘house’, ‘mountain’, ‘father’, ‘milk’, and so on belong to one class, while the

words meaning ‘to walk’, ‘to see’, ‘to kill’, and ‘to give’ belong to the other. [...] The earmark of nominal inflection is case [...] Each of the roughly 130 suffixes that indicate case explicitly is therefore criterial of nounhood. [...] Hundreds of monomorphemic stems (i.e. roots) accept [the ablative inflection] *-tsinnut*, and hundreds of others do not: *illu* ‘house’ *illutsinnut* ‘to our house’, *pisuk* ‘to walk’ **pisutsinnut*. Mood is the critical feature of verbal inflection. [Transitive indicative] *-poq/-voq* can be added only to certain roots and stems and not to others, *pisuppoq* ‘(s)he is walking’ but **illu(v)a(a)*. Note that among inflectable roots, exactly those that reject nominal case morphology, for example *pisuk* ‘to walk’, accept verbal mood signs, and exactly those that reject verbal mood signs, for example *illu* ‘house’, accept nominal case morphology.

This is not to say that all four major categories are present in all languages, or that the dividing lines are drawn in the exact same fashion in every language – in particular, nouns and adjectives can be hard to distinguish. A well-known example is provided by Japanese *keiyōdōshi*, which originate in the large-scale absorption of Chinese forms into Japanese via [reading of Chinese](#). The name *keiyōdōshi* suggests we are dealing with neither adjectives nor nouns but rather verbs, but this makes sense only if the stems in question are considered together with the copula *da* – by themselves they are just adjectives, distinguished from the basic *yamato* layer of the adjectives by their inability to take *yamato* adjectival suffixes (*-i* and its replacements in negation, past tense, etc.).

In languages with a more productive core morphology we observe *suffixal coercion*, whereby the category-setting effect of a suffix is so strong as to override the inherent lexical category of the stem: for example, in English *Every noun can be verbed* the past tense suffix *-ed* coerces the noun stem *verb* into acting like a verbal stem. A subtle but important case of such coercion is when the semantics is adjusted: for example English mass nouns like *wine* or *soap* do not pluralize in the obvious sense, only in the meaning ‘types of’ or ‘kinds of’, as in *We tasted three wines: Chardonnay, Pinot Noir, and Pinot Meunier*.

The main categories N, A, V, Adv can often be further subdivided into subcategories such as proper names (PN, traditionally viewed as a subcategory of N, but assigned its own major category in Universal Dependencies), numerals (Num, traditionally viewed as a subcategory of A, but a separate closed category in UD), and so forth. An important and much analyzed case is when the subcategorization is provided by the tectogrammar. Verbs, in particular, can differ greatly in the number and kind of arguments they take, and finer classifications based on the arguments lead to hundreds of subclasses (Levin, 1993; Kipper et al., 2008).

One final difficulty we consider here is that of multiple class membership. For many words, like *need*, *rent*, *divorce*, and so forth, it is not at all clear whether it is the verbal or the nominal form one should take as basic, and it seems very hard to devise any test that could settle the matter. In these cases, traditional grammar simply lists the form in question with both parts of speech, a method we will find easy to emulate. Since we have equated lexical categories either with pure distributional patterns or with combinations



of these (disjunctive for forms in the same lemma, conjunctive for forms in multiple classes), all we need is a formal definition that is closed under both union and intersection. Ignoring both lemmatization and multiple class membership for a moment, POS tagging is simply a function π that maps any morpheme to a tag, and does this in a manner compatible with the syntactic congruence: for $u, v \in L$, if $u \sim v$ we must have $u\pi = v\pi$.

In Section 5.2 we divided the POS tags that are output by π in two parts, with the inflectional information separated from the category label by \diamond so as to have $(\text{boy})\pi = N\langle\text{SG}\rangle$ and $(\text{boys})\pi = N\langle\text{PL}\rangle$. When we say that all inflectional variants belong in the same lemma, this amounts to the introduction of another mapping μ that strips the \diamond -enclosed part away, so that $(\text{boy})\pi\mu = (\text{boys})\pi\mu = N$. Since μ is given by a finite list, if π is a rational mapping (FST) of the vocabulary so will be $\pi\mu$, which gives us the license to define printnames only up to μ . Permitting multiple class membership will change π only in that it becomes a relation, rather than a single-valued function. Since only finitely many elements are involved, this can be done without altering the character of π as an FST. Altogether, lexical categories will be handled in the control of lexemes, using only finite state means.

The standard lexicographic lemma is obtained by bringing together the inflected forms of the same stem in paradigms (see Section 5.2) and choosing an appropriate headword (see Section 6.1). In lexicography the headword is typically the least marked form (such as the nominative singular for nouns, or the 3rd person present indicative for verbs – English is an exception in this regard), but generative morphology often resorts to an underlying form that does not coincide with any of the surface forms that make up the lemma. Since we don't take up the full morphological analysis/synthesis task for the reasons discussed in Section 5.2, our treatment of printnames can stay rather loose: it simply makes no difference which form of a lemma we use in definitions, whether we define *dog* by *four-legged* or *four-leg*, *goat* by *stink* or *stinky*, and so on.

To save space, lexicographers often enlarge their lemmas by collecting together not just inflected but also derived forms under the same lemma, as long as the derivation is compositional. As an example, consider English agent nouns, which are formed from verbal stems by adding the suffix *-er* as in *eat/eater*, *buy/buyer*, *sleep/sleeper*, etc. The semantics of the suffix is transparent: *x-er* is 'the one doing the *x*-ing' for every verb *x* where the suffix is applicable. We encounter some difficulties that are common (indeed, typical) of derivation, in particular *suppletion*, whereby set forms like *actor* and *thief* block the forms **acter* and **stealer*, and the fact that the rule is not entirely productive: for *John eats tomatoes* we obtain *John is a tomato-eater* but for *John needs money* we fail to obtain *?John is a money-needer*. Be that as it may, both the syntax and the semantics of *-er* suffixation are rather clear: on the syntax side the operation is one of converting Vs to Ns, and on the semantics side it is informally as described above (a more formal description is deferred to the next section). All of this is obtained by a simple context-free rule $N \rightarrow V\text{-er}$, where N and V are nonterminals and the formative *-er* is a terminal. That context-free and context-sensitive rules that do not apply to their own input can

be converted into FSTs has long been known (the method was discovered originally by Johnson (1970) and independently by Kaplan and Kay (1994)), and we will employ these conversion techniques to stay within the finite state realm.

6.4 Word meaning

Currently, there are two main approaches toward defining word meaning. In Section 2.7 we discussed the *distributional* or *CVS* model that seeks to model word meaning by a vector in Euclidean space. In Section 4.5 we presented the *algebraic* approach built on preexisting lexicographic work, in particular dictionary definitions that are already restricted to a smaller wordlist such as the Longman Defining Vocabulary (LDV). Both approaches suggest a method for approximating the intuitively clear but methodologically very challenging notions of a *semantic similarity* and *semantic field*. In distributional semantics, we want to measure semantic similarity by means of Euclidean distance, and in algebraic semantics by reduction to the similarity of the definitional graphs.

It is evident from the typeset page that the bulk of the information in a lexical entry is in the definitions, and this is easily verified by estimating the number of bits required to encode the various components. Also, definitions are the only truly obligatory component in a lexical entry, because a definition will be needed even for words lacking in exceptional forms (these are the majority) or an exceptional etymology, with a neutral stylistic value, predictable part of speech (most words are nouns), and an orthography sufficiently indicative of pronunciation. In bilingual (and multilingual) dictionaries word meanings are given by translations, a method that makes perfect sense for adults who already speak one language, the source, and need to find out what is meant by a word in another language, the target. The lexicographic task of expressing what a target word means *without* assuming a source language is much harder, as can be seen in Fig. 6.1, which provides the definition for *hastate* ‘triangular with sharp basal lobes spreading away from the base of the petiole’ and the example *hastate leaves*. Well, why does *hastate* leave, and where does he go?

Exercise 6.5 Collect data on the frequency of POS categories, and estimate the number of bits required to encode it.

Exercise 6.6 Collect data on the frequency of characters used in phonemic transcription, and estimate the number of bits required to encode it.

Exercise 6.7 What is the information content of a 300-dimensional real vector? Is the answer affected by the number of bits used in encoding the coordinates?

Exercise 6.8 Does knowing the POS for *hastate* help to disambiguate between *leaves*₁ ‘takes leave’ and *leaves*₂ ‘more than one leaf’? Why?

Even if we get over the unfortunate ambiguity between *leaves*₁ and *leaves*₂, we are not much closer to an understanding of *hastate* than before, for leaves come in all forms and shapes. One cannot ask for a better illustration of Leibniz’s point about the deferral of debt we quoted in Section 4.5: could it be that we need to look in the dictionary again,

to look up *basal*, *lobes*, and *petiole*? Is, perhaps, sense 2b of *basal* ‘of, relating to, or being essential for maintaining the fundamental vital activities of an organism’ meant here? The second meaning of *hastate* (if, indeed, there are two different meanings at play) is given as *shaped like a spear or the head of a spear* and the example *a hastate spot of a bird* is even more unhelpful, for birds can have all kinds of spots, just as plants can have all kinds of leaves. Even if we accept ‘triangular’ as part of the definition, lobes are curvy, and triangles are made of straight lines; what is going on here?

It is, in the end, the definition intended for children, ‘shaped like an arrowhead with flaring barbs’, that is most helpful for those who really don’t know the word. This definition still involves others, *arrowhead*, *flare*, *barb*, but we at least learn that the word refers to some kind of shape. We will encode this piece of critical information by the relation *hastate* IS_A *shape* and depict it as a graph edge $\text{hastate} \xrightarrow{0} \text{shape}$. The `dict_to_4lang` module can automatically create the representation, such as that shown in Fig. 6.2.

Comp

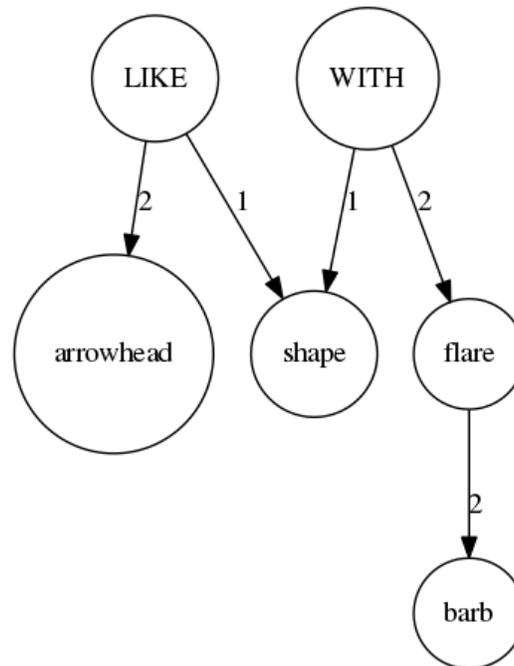


Fig. 6.2. Representation of *hastate*

As we iteratively drill down into the components of the definition such as *flare*, the distinction between lowercase `typewriter font` and uppercase `SMALL CAPS` entities becomes helpful: we use uppercase for those binaries where bringing up a definition adds little value. For example, at dictionary.com, *be like* is defined as ‘bearing resemblance’, *resemble* is defined as ‘be like or similar to’ and *similar* is defined as



‘having a likeness or resemblance’. Entries such as HAVE, LIKE, AT are best thought of as *primitives* terminating the recursive search. The important thing is to know that being primitive does not make these concepts unique: in the example at hand we could take any of `be_like`, `resemble`, and `be_similar` as primitive and the other two as derived. We have already encountered similar cases like *prison*, *inmate*, *guard* in Section 4.5 and on occasion we even find ‘laboratory pure’ examples, such as the days of the week, where every one of them can be taken as primitive, leaving all the others as defined. Choosing a defining vocabulary \mathbf{D} simply amounts to choosing a **feedback vertex set** in the directed graph that contains every lemma as a node and a directed edge from u to v iff v appears in the definition of u .



A defining vocabulary subdivides the problem of defining the meaning of (English) words in two. First, the definition of other vocabulary elements in terms of \mathbf{D} , which is our focus of interest, and second, defining \mathbf{D} itself, based perhaps on primary (sensory) data or perhaps on some deeper scientific understanding of the primitives. A complete solution to the dictionary definition problem must go beyond a mere listing \mathbf{D} of the defining vocabulary elements: we need both a formal model of each element and a specification of lexical syntax, which regulates how elements of \mathbf{D} combine with each other (and possibly with other, already defined, elements) in the definition of new words.

Ling We emphasize that our goal is to provide an *algebra* of lexicography rather than a generative lexicon (Flickinger, 1987; Pustejovsky, 1995) of the sort familiar from generative morphology. A purely generative approach would start from some primitives and some rules or constraints which, when applied recursively, provide an algorithm that enumerates the lexicon. The algebraic approach is more modest, as it largely leaves open the actual contents of the lexicon. Consider the semantics of noun–noun compounds. As Kiparsky (1982) notes, *ropeladder* is ‘ladder *made of* rope’, *manslaughter* is ‘slaughter *undergone by* man’, and *testtube* is ‘tube *used for* test’, so the overall semantics can only specify that $N_1 N_2$ is ‘ N_2 that is V -ed by N_1 ’, i.e. the decomposition is subdirect (yields a superset of the target) rather than direct, as it would be in a fully compositional generative system.

Another difference between the generative and the algebraic approach is that only the former implies commitment to a specific set of primitives. To the extent that work on lexical semantics often gets bogged down in a quest for the ultimate primitives, this point is worth a small illustrative example. Consider the cyclic group Z_3 on three points given by the elements e, a, b and the multiplication table shown in Table 6.1.

	e	a	b
e	e	a	b
a	a	b	e
b	b	e	a

Table 6.1. Multiplication in Z_3

The unit element e is unique (being the one and only y satisfying $yx = xy = x$ for all x) but not necessarily irreducible, in that if a and b are given, both ab and ba could be used to define it. Furthermore, if a is given, there is no need for b , because aa already defines this element, so the group can be presented simply as $a, aa, aaa = e$, i.e. a is the ‘generator’ and $a^3 = e$ is the ‘defining relation’ (as these terms are used in group theory). Note, however, that the exact same group is equally well presented by using b as the generator and $b^3 = e$ as the defining relation – there is no unique/distinguished primitive as such. This is the non-uniqueness we discussed above. In a distributional system, the problem arises somewhat differently, since in a space of dimension d we can select any d vectors and these are guaranteed to be ‘defining’ as long as they are linearly independent. The issue is that the dimensionality of an embedding is not set in stone: rather, it is the result of a dimension reduction procedure such as PCA (see Section 2.7), and we lack good criteria for making the right choice of d .

What, then, is a reasonable cardinality d for a defining vocabulary \mathbf{D} ? The approach taken here is to define \mathbf{D} from the outside in, by analyzing the LDV or BE rather than building from the inside out from the putative core lists of Schank or Wierzbicka. This method guarantees that at any given point in reducing \mathbf{D} to some smaller \mathbf{D}' we remain capable of defining all other words, not just those listed in LDOCE (some 90k items) or the Simple English Wikipedia (over 30k entries) but also those that are definable in terms of these larger lists (really, the entire unabridged vocabulary of English). In the computational work that fuels the theoretical analysis presented here, we begin with our own version of the LDV, called `4lang`, which includes Latin, Hungarian, and Polish translations in the intended senses, both because we do not wish to lose sight of the longer-term goal of translation and to provide a clear means of disambiguation for concepts whose common semantic root, if there ever was one, is no longer transparent, for example *interest* ‘usura’ v. *interest* ‘studium’. Clearly, a similarly disambiguated version of the BE vocabulary or any other reasonable starting point could just as well be used, and any such choice provides an upper bound $|\mathbf{D}|$ on d .

Most of the difficulties in creating a defining vocabulary are already evident in a single *semantic field* (Trier, 1931), conceptually related terms that are likely candidates to be defined in terms of one another such as color terms, legal terms, and so on. We will not attempt to define the notion of semantic fields in a rigorous fashion, but will use an operational definition based on [Roget’s Thesaurus](#). For example, for *color* terms we take about 30 stanzas, from Roget 420 Light to Roget 449 Disappearance (numbering follows the 1911 edition of Roget’s, as this is available as Project Gutenberg etext #10681), and for *religious* terms we take 25 stanzas from Roget 976 Deity to Roget 1000 Temple.

It is not at all trivial to consistently define a notion of [semantic fields](#): it is not clear how many fields one needs, where the limits of each field are, whether the resulting collections of words and concepts are properly named, and whether some kind of hierarchy can or should be imposed on them. But using Roget’s to a large extent obviates these problems, since its coverage is broad, and each Roget field forms a reasonable unit of workable size, perhaps a few dozen to a few hundred stanzas. We will use the *Religion*





field to illustrate our approach, not because we see it as somehow privileged but rather because it serves as a strong reminder of the inadequacy of the *physicalist* approach that seeks to root all language in objective reality. In discussing color, we might be tempted to dispense with a defining vocabulary **D** in favor of a more scientifically defined core vocabulary, but in general such core expressions, if truly restricted to measurable *qualia*, have very limited traction over much of human social activity. A more physicalist analysis could be made available for many semantic fields defined through Roget, such as *Size* R031–R040a and R192–R223, and *Econ* R775–R819. With the aid of a naive theory of psychology (Section 3.4), we could build out the semantic fields of *Emotion/attitude*, R820–R936 (except for 845–852 and 922–927), *Esthetics*, R845–R852, and *Law/morals* R937–R975 plus R922–R927.

Exercise[†] 6.9 Consider R775–R819 ‘relations which concern property’, as a semantic field. Define all words found therein by means of a primitive vocabulary.

For *Religion* we obtain the following list (all entries are lowercased for ease of automated stemming): *anoint, believe, bless, buddhism, buddhist, call, ceremony, charm, christian, christianity, christmas, church, clerk, collect, consecrated, cross, cure, devil, dip, doubt, duty, elder, elect, entrance, fairy, faith, faithful, familiar, fast, father, feast, fold, form, glory, god, goddess, grace, heaven, hinduism, holy, host, humble, jew, kneel, lay, lord, magic, magician, mass, minister, mosque, move, office, people, praise, pray, prayer, preserve, priest, pure, religion, religious, reverence, revile, rod, save, see, service, shade, shadow, solemn, sound, spell, spirit, sprinkle, temple, translate, unity, word, worship*. Two problems are evident from such a list. First, there are several words that do not fully belong in the semantic field, in that the sense presented in Roget’s is different from the sense in the LDV: for example, *port* is not a color term and *father* is not a religious term in the primary sense used in the LDV.

For the purpose of probing the defining vocabulary, such words can be manually removed, since defining the religious sense of *father* or the color sense of *port* would in no way advance the cause of reducing the size of **D**. Programmatic removal is not feasible at this stage: to see what the senses are, and thus to see that the core sense is not the one used in the field, would require a working theory of lexical semantics of the sort we are developing here. Once such a theory is at hand, we may use it to verify the manual work performed early on, but this is only a form of error checking, rather than learning something new about the domain. Needless to say, *father* still needs to be defined or declared a primitive, but the place to do this is among kinship terms, not religious terms.

If a word is kept, this does not mean that it is unavailable outside the semantic field; clearly, *Bob worships the ground Alice walks on* does not mean anything religious. However, for words inside the field such as *worship* even usage external to the field relies on the field-internal metaphor, so the core/defining sense of the word is the one inside. Conversely, if usage does not require the field-internal metaphor, the word/sense need not be treated as part of the size reduction effort: for example, *This book fathered a new genre* does not mean (or imply) that the object will treat the subject with reverence, so *father* can be left out of the *religion* field. Ideally, with a full sense-tagged corpus one could

see ways of making such decisions in an automated fashion, but in reality creating the corpus would require far more manual work than making the decisions manually.

Since the issue of different word senses comes up early on (see Section 3.9), some methodological remarks are in order. Kirsner (1993) distinguishes two polarly opposed approaches. The *polysemic* approach is aimed at maximally distinguishing as many senses as appear distinct, for example *bachelor*₁ ‘unmarried adult man’, *bachelor*₂ ‘fur seal without a mate’, *bachelor*₃ ‘knight serving under the banner of another knight’, and *bachelor*₄ ‘holder of a BA degree’. The *monosemic* approach (also called the *Saussurean* and the *Columbia School* approach by Kirsner, who calls the polysemic approach *cognitive*) searches for a single, general, abstract meaning, and would subsume at least the first three senses above into a single definition, ‘unfulfilled in typical male role’. This is not the place to fully compare and contrast the two approaches (Kirsner’s work offers an excellent starting point), but we note here a significant advantage of the monosemic approach, namely that it makes interesting predictions about novel usage, while the predictions of the polysemic approach border on the trivial. To stay with this example, it is possible to envision a novel usage of *bachelor* to denote a contestant in a game who wins by default (because no opponent could be found in the same weight class or the opponent was a no-show). The polysemic theory would predict that not just seals but maybe also penguins without a mate may be termed *bachelor* – true but not very revealing.

The choice between monosemic and polysemic analysis need not be made on a priori grounds: even the strictest adherent of the polysemic approach would grant that *bachelor’s degree* refers, at least historically, to the same kind of apprenticeship as *bachelor knight*. Conversely, even the strictest adherent of the monosemic approach must admit that the relationship between ‘obtaining a BA degree’ and ‘being unfulfilled in a male role’ is no longer apparent to contemporary language learners. That said, we still give methodological priority to the monosemic approach because of the original Saussurean motivation: if a single form is used, the burden of proof is on those who wish to posit separate meanings (Ruhl, 1989). An important consequence of this methodological stance is that we avoid speaking of *metaphorical* usage (see Section 4.2), assuming instead that the core meaning already extends to such cases.

A second problem, which has a notable impact on the structure of the list, is the treatment of natural kinds (see Section 2.7). By natural kinds here we mean not just biologically defined kinds such as *ox* or *yak*, but also culturally defined artifact types like *tuxedo* or *microscope* – as a matter of fact, the cultural definition has priority over the scientific definition when the two are in conflict. The biggest reason for the inclusion of natural kinds in the LDV is not conceptual structure but rather the Eurocentric viewpoint of LDOCE: for the English speaker it is reasonable to define the *yak* as ox-like, but for a Tibetan, defining the ox as yak-like would make more sense. There is nothing wrong with being Eurocentric in a dictionary of an Indo-European language, but from a general perspective neither of these terms can be truly treated as primitive.

So far we have discussed the *lexicon*, the repository of linguistic knowledge about words. Here we must say a few words about the *encyclopedia*, the repository of world



knowledge. While our goal is to create a formal theory of lexical definitions, it must be acknowledged that such definitions can often elude the grasp of the linguist and slide into a description of world knowledge of various sorts. Lexicographic practice acknowledges this fact by providing, somewhat begrudgingly, little pictures of flora, fauna, or plumbers' tools. A well-known method of avoiding the shame of publishing a picture of a yak is to make reference to *Bos grunniens* and thereby point the dictionary user explicitly to some encyclopedia where better information can be found. We will collect such pointers together in a set **E**, and use curly braces {} to set them typographically apart from references to lexical content.

When we say that *light* is defined as {flux of photons in the visible band}, what this really means is that `light` must be treated as a primitive. There is a physical theory of light which involves photons, and a biophysical theory of visual perception that involves sensitivity of the retina to photons of specific wavelengths, but we are not interested in these theories; we are just offering a pointer to the person who is. From the linguistic standpoint *light* is a primitive, irreducible concept, one that people used for millennia before the physical theory of electromagnetic radiation, or even the very notion of photons, was available. Ultimately any system of definitions must be rooted in primitives, and we believe the notion `light` is a good candidate for such a primitive. From the standpoint of lexicography, only two things need to be said: first, whether we intend to take the nominal or the verbal meaning as our primitive, and second, whether we believe that the primitive notion `light` is shared across the oppositions with 'dark' and with 'heavy' or whether we have two different senses of *light*. In this particular case, we choose the second solution, treating the polysemy as an accident of English rather than a sign of a deep semantic relationship, but the issue must be confronted every time we designate an element as primitive.

The same point needs to be made in regard to ontological primitives like *time*. While it is true that the time used in the naive physics model is discrete and asynchronous, this is not intended as some hypothesis concerning the ultimate truth about physical time, which appears continuous (except possibly at the Planck scale) and appears distinct from space and matter (but is strongly intertwined with these). Since the model is not intended as a technical tool for the analysis of synchrony or continuous time, we do not wish to burden it with the kind of mechanisms, such as [Petri nets](#) or [real numbers](#), that one would need to analyze such matters.

Encyclopedic knowledge of `time` may of course include reference to the real numbers or other notions of continuous time, but our focus is not on a deep understanding of time but on tense marking in natural language, and it is the grammatical model, not the ontology, that carries the burden of recapitulating this. For the sake of concreteness we will assume a Reichenbachian view, distinguishing four different notions of time: (i) *speech time*, when the utterance is spoken, (ii) *perspective time*, the vantage point of temporal deixis, (iii) *reference time*, the time that adverbs refer to, and (iv) *event time*, the time during which the named event unfolds. Typically, these are intervals, possibly open-ended, or more rarely points (degenerate intervals), and the hope is that we



can eventually express the temporal semantics of natural language in terms of interval relations such as ‘event time precedes reference time’ (see Allen, Gardiner, and Frantz (1984), Allen and Ferguson (1994), and Kiparsky (1998)). The formal apparatus required for this is considerably weaker than that of FOL, and will require only two primitives, BEFORE and AFTER.

One important use of external pointers worth separate mention is for proper names. By *sun* we mean primarily the star nearest to us. The common noun usage is secondary, as is clear from the historical fact that people before Giordano Bruno didn’t even know that the small points of light visible in the night sky were also suns. That we have a theory of the Sun as {the nearest star} where the, near, -est, and star are all members of the LDV is irrelevant from a lexicographic standpoint – what really matters is that there is a particular object, ultimately identified by *deixis*, that is a natural kind in its own right. The same goes for natural kinds such as *oxygen* or *bacteria* that may not even have a naive lexical theory (it is fair to say that all our knowledge about these belongs in chemistry and the life sciences), and about cultural kinds such as *tennis*, *television*, *british*, or *october*.



In Section 3.5 we discussed how to formalize those cases when purely lexical knowledge is associated with natural kinds, for example that tennis is a game played with a ball and rackets, that November immediately follows October, or that bacteria are small living things that can cause disease, but we wish to emphasize here that there is much in the encyclopedia that our formalism is not intended to cover, for example that the standard atomic weight of oxygen is 15.9994(3). Given that much of classical Knowledge Representation concentrates on capturing the knowledge stored in the encyclopedia that is external for our purposes, it is worth keeping in mind that natural and cultural kinds amount to less than 6% of the LDV. Certainly there is very little that we can reliably know about the field of religion, to which we now return.

When we define *Islam* as religion centered on the teachings of {Muhammad}, the curly braces acknowledge the fact that Muhammad (and similarly Buddha, Moses, or Jesus Christ) will be indispensable in any effort aimed at defining Islam (Buddhism, Judaism, or Christianity, respectively). The same is true for Hinduism, which we may define as being centered on revealed teachings ({śruti}), but, of course, to obtain Hinduism as the definiendum the definiens must make it clear that it is not any old set of revealed teachings that are central to it but rather the Vedas and the Upanishads. One way or another, when we wish to define such concepts as specific religions, some reference to specific people and texts designated by proper names is unavoidable.

Remarkably, once the names of major religious figures and the titles of sacred texts are treated as pointers to the encyclopedia, there remains nothing in the whole semantic field that is not definable in terms of non-religious primitives. In particular, *god* can be defined as being, supreme, where supreme is simply about occupying the highest position in a hierarchy. Being a being has various implications (see Section 3.4), but

none of these are particularly religious. The same does not hold for the semantic field of color, where we find irreducible entries such as *light*.

Needless to say, our interest is not in exegesis (no doubt theologians could easily find fault with the particular definitions of God and the major religions offered here), but in the more mundane aspects of lexicography. Once we have *buddhism*, *christianity*, *hinduism*, *islam*, and *judaism* defined, *buddhist*, *christian*, *hindu*, *muslim*, and *jew* fall out as *adherent of buddhism, . . . , judaism* for the noun denoting a person, and similarly for the adjectives *buddhist*, *christian*, *hindu*, *islamic*, *jewish* which get defined as *of or about buddhism, . . . , judaism*. We are less concerned with the theological correctness of our definitions than with the proper choice of the base element: should we take the *-ism* as basic and the *-ist* as derived, should we proceed the other way round, or should we perhaps derive both (or, if the adjectival form is also admitted, all three) from a common root? Our general rule is to try to derive the morphologically complex from the morphologically simple, but exceptions must be made, for example, when we treat *jew* as derived (as if the word was **judaist*). These are well handled by the principle of blocking (Aronoff, 1976), which makes the non-derived *jew* act as the printname for **judaist*.

Another, seemingly mundane but in fact rather thorny issue is the treatment of bound morphemes (Section 5.2). The LDV includes, with good reason, some forty suffixes *-able*, *-al*, *-an*, *-ance*, *-ar*, *-ate*, *-ation*, *-dom*, *-en*, *-ence*, *-er*, *-ess*, *-est*, *-ful*, *-hood*, *-ible*, *-ic*, *-ical*, *-ing*, *-ion*, *-ish*, *-ist*, *-ity*, *-ive*, *-ization*, *-ize*, *-less*, *-like*, *-ly*, *-ment*, *-ness*, *-or*, *-ous*, *-ry*, *-ship*, *-th*, *-ure*, *-ward*, *-wards*, *-work*, *-y* and a dozen prefixes *counter-*, *dis-*, *en-*, *fore-*, *im-*, *in-*, *ir-*, *mid-*, *mis-*, *non-*, *re-*, *un-*, *vice-*, *well-*. This affords great reduction in the size of **D**, in that a stem such as *avoid* now can appear in the definiens in many convenient forms such as *avoidable*, *avoidance*, and *avoiding* as the syntax of the definition dictates. Including affixes is also the right decision from a cross-linguistic perspective, as it is evident that notions that are expressed by free morphemes in one language, such as possession (English *my*, *your*, . . .), are expressed in many other languages by affixation. But polysemy can be present in affixes as well: for example, English and Latin have four affixes *-an/anus*, *-ic/ius*, *-ical/icus*, and *-ly/tus* where Hungarian and Polish have only one, *-i/anin*, and we have to make sure that no ambiguity is created in the definitions by the use of polysemous affixes. Altogether, affixes and affix-like function words make up about 8–9% of the LDV, and the challenge they pose to the theory developed here is far more significant than that posed by natural kinds because their analysis involves very little, if any, reference to encyclopedic knowledge.

Finally, there is the issue of the economy afforded by primitive conceptual elements that have no clear exponent in the LDV. For example, we may decide that we feel *sorrow* when something bad happens to us, *gloating* when it happens to others, *happiness* when something good happens to us, and *resentment* when it happens to others. (The example is from Hobbs (2008), and there is no claim here or in the original that these are the best or most adequate emotional responses. Even if we agree that they are not, this does not affect the following point, which is about the economy of the system rather than about

morally correct behavior.) Given that `good`, `bad`, and `happen` are primitives we will need in many corners of the system, we may wish to rely on some sociological notion of `in-group` and `out-group` rather than on the pronouns `us` and `them` in formalizing the above definitions. This has the clear advantage of remaining applicable independent of the choice of `in-group` (be it family, tribe, nation, colleagues, etc.) and of indexical perspective (be it ours or theirs). Considerations of economy dictate that we should use abstract elements as long as we can reduce the defining vocabulary \mathbf{D} by more than one item: whether we prefer to use `in-group`, `out-group` or `us`, `them` as primitives is more a matter of taste than a substantive issue. If two solutions \mathbf{D} and \mathbf{D}' have the same size, we have no substantive reason to prefer one to the other. That said, for expository convenience we will still prefer non-technical to technical and Anglo-Saxon to Latinate vocabulary in our choice of primitives.

To summarize what we have so far, for the sake of concreteness we have identified a somewhat reduced version of the LDV, with fewer than 2,000 items, including some bound morphemes and natural kinds, as our defining vocabulary \mathbf{D} , but we make no claim that this is in any way superior to some other base list \mathbf{D}' as long as \mathbf{D}' is not bigger than \mathbf{D} . Appendix 4.8 lists a feedback vertex set selected from the original 2,000.

6.5 The formal model

The syntax of well-formed lexemes can be summarized in a context-free grammar (V, Σ, R, S) as follows. The nonterminals V are the start symbol S , the binary relation symbols collected together in B , and the unary relation symbols collected together in U . Variables ranging over V will be taken from the end of the Latin alphabet, v, w, x, y, z . The terminals are the grouping brackets '[' and ']' and the derivation history parentheses '(' and ')', and we introduce a special terminating operator ';' to form a terminal v ; from any nonterminal v . The rule $S \rightarrow U|B|\lambda$ handles the decision to use unary or binary lexemes, or perhaps none at all. The operation of *attribution* is captured in the rule schema $w \rightarrow w; [S^*]$ which produces the list defining w . This requires the CFG to be *extended* in the usual sense that regular expressions are permitted on the right-hand side, so the rule really means $w \rightarrow w; []|w; [S]|w; [SS]| \dots$ Finally, the operation of *predication* is handled by $u \rightarrow u; (S)$ for unary and $v \rightarrow Sv; S$ for binary nonterminals. All lexemes are built up recursively by these rules.

The first level of combining lexemes is morphological. At the very least, we need to account for productive derivational morphology, the prefixes and suffixes that are part of \mathbf{D} , but in general we expect a theory that is just as capable of handling cases not easily exemplified in English such as *binyanim*. Compounding, to the extent that it is predictable, also belongs here, and so does nominalization, especially as definitions make particularly heavy use of this process. The same is true for inflectional morphology, where the challenge is not so much English (though the core set *-s*, *'s*, *-ing*, *-ed* must be covered) as languages with more complex inflectional systems. Since certain categories (for example gender and class system) can be derivational in one language but inflectional in another,

what we really require is *coverage of all productive morphology*. This is obviously a tall order, and within the confines of this chapter all we can do is to discuss one example, deriving *insecure* from *in-* and *secure*, as this will bring many of the characteristic features of the system into play.

Irrespective of whether *secure* is primitive (we assume it is not), we need some mechanism that takes the *in-* lexeme and the *secure* lexeme, and creates an *insecure* lexeme whose definition and printname are derived from those of the inputs. To forestall confusion, we note here that not every morphologically complex word will be treated as derived. For example, it is clear from the strong verb pattern that *withstand* is morphologically complex, derived from *with* and *stand* (otherwise we would expect the past tense to be **withstended* rather than *withstood*), yet we do not attempt to describe the operation that creates it. We are content with listing *withstand*, *understand*, and other complex forms in the lexicon, though not necessarily as part of **D**. Similarly, if we have a model capable of accounting for *insecure* in terms of more primitive elements, we are not required to overapply the technique to *inscrutable* or *ineffable* just because these words are also morphologically complex and could well be, historically, the residue of *in-* prefixation to stems no longer preserved in the language. Our goal is to define meanings, and the structural decomposition of every lexeme to irreducible units is pursued only to the extent it advances this goal.

Returning to *insecure*, the following facts should be noted. First, that the operation resides entirely in *in-* because *secure* is a free form. Second, that a great deal of the analysis is best formulated with reference to lexical categories (parts of speech): for example, *in-* clearly selects for an adjectival base and yields an adjectival output (the category of *in-* is A/A), because those forms such as *income* or *indeed* that are formed from a verbal or nominal base lack the negative meaning of *in-* that we are concerned with (and are clearly related to the preposition *in* rather than the prefix *in/im* that is our target here). Third, that the meaning of the operation is exhaustively characterized by the negation: forms like *infirm*, where the base *firm* no longer carries the requisite meaning, still carry a clear negative connotation (in this case, ‘lacking in health’ rather than ‘lacking in firmness’). In fact, whatever meaning representation we assign to the lexically listed element *insecure* must also be available for the non-lexical (syntactically derived) *not secure*.

In much of model-theoretic semantics (the major exception is the work of Turner (1983) and Turner (1985)), preserving the semantic unity of stems like *secure* which can be a verb or an adjective, or stems like *divorce* which can be a noun or a verb, with no perceptible meaning difference between the two, is extremely hard because of the differences in signature. Here it is clear that the verb is derived from the adjective: clearly, the verb *to secure x* means ‘make x (be) secure’, so when we say that *in-* selects for an adjectival base, this just means that the part of the POS structure of *secure* that permits verbal combinatorics is filtered out by application of the prefix. The adjective *secure* means ‘able to withstand attack’. Prefixation of *in-* is simply the addition of the primitive *neg* to the semantic representation and concatenation plus assimilation in the

first consonant, cf. *in+secure* and *im+precise*. (We note here, without going into details, that the phonological changes triggered by the concatenation are also entirely amenable to treatment in finite state terms.)

As far as the invisible deadjectival verb-forming affix (paraphrased as *make*) that we have posited here to obtain the verbal form is concerned, this does two things: first, it brings in a subject slot *x*, and second, it contributes a change-of-state predicate – before, there wasn't an object *y*, and now there is. The first effect, which requires making a distinction between an external argument (subject) and internal argument (direct object, indirect object, etc.), follows a long tradition of syntactic analysis going back at least to Williams (1981), and will just be assumed without argumentation here, but the latter is worth discussing in greater detail, as it involves a key operation among lexemes, *substitution*, to which we turn now.

Some form of recursive substitution of definitions in one another is necessary both for work aimed at reducing the size of the defining vocabulary and for attempts to define non-**D** elements in terms of the primitives listed in **D**. When we add an element of negation (here given simply as *neg*, and a reasonable candidate for inclusion in **D** – see Section 7.3 for further discussion) to a definition such as ‘able to withstand attack’, how do we know that the result is ‘not able to withstand attack’ rather than ‘able to not withstand attack’ or even ‘able to withstand not attack’? The question is particularly acute because the head just contains the defining properties as elements of a set, with no order imposed. (We note that this is a restriction that we could trivially give up in favor of ordered lists, but only at a great price: once ordered lists were admitted, the system would become Turing-complete, just like HPSGHead-driven phrase structure grammar, HPSG). Another way of asking the same question is to ask how the system deals with iterated substitutions, for even if we assume that *ABLE* and *attack* are primitives (they are listed in the LDV), surely *withstand* is not; *x withstands y* means something like ‘*x* does not change from *y*’ or even ‘*x* actively opposes *y*’. Given our preference for a monosemic analysis, we take the second of these as our definition, but this makes the problem even more acute: how do we know that the negation does not attach to the *actively* portion of the definition? What is at stake here is the single most important property of definitions, that the definiens can be substituted for the definiendum in any context.

Since many processes, such as making a common noun definite, which are performed by syntactic means in English, are performed by inflectional means in other languages such as Romanian, *complete coverage of productive morphology in the world's languages* already implies coverage of a great deal of syntax in English. Ideally, we would wish to take this further, requiring coverage of syntax as a whole, but we might be satisfied with slightly less, covering the meaning of syntactic constructions only to the extent to which they appear in dictionary definitions. Remarkably, almost all problem cases in syntax are already evident in this restricted domain, especially as we need to make sure that constructions and idioms are also covered. There are forms of grammar which assume all syntax to be a combination of constructions (Fillmore



and Kay, 1997), and the need to cover the semantics of these is already clear from the lexical domain: for example, a *mule* is animal, cross between horses and donkeys, stubborn, . . . Clearly, a notion such as ‘cross between horses and donkeys’ is not a reasonable candidate for a primitive, so we need a mechanism for feeding back the semantics of nonce constructions into the lexicon.

This leaves only the totally non-lexicalized, purely grammatical part of syntax out of scope, cases such as topicalization and other manipulation of given/new structure, as dictionary definitions tend to avoid communicative dynamics. But with this important caveat we can state the requirement that lexical semantics covers not just the lexical, but also the syntactic combination of morphemes, words, and larger units.

6.6 The semantics of lexemes

Now that we have seen the basic elements (lexemes) and the basic mode of combination (attribution, modeled as listing in the base of a lexeme), the question will no doubt be asked: how is this different from Markerese (Lewis, 1970)? The answer is that we will interpret our lexemes in model structures, and make the combination of lexemes correspond to operations on these structures, very much in the spirit of Montague (1970). Formally, we have a source algebra \mathcal{A} that is freely generated from some set of primitives \mathbf{D} by means of constructions listed in \mathbf{C} . An example of such a construction is *x is to y as z is to w*, which is used not just in arithmetic (proportions) but also in everyday analogy: *Paris is to London as France is to England*, but *in*-prefixation would also be a construction of its own. We will also have an algebra \mathcal{M} of *machines*, which will serve as our model structures, and a mapping σ of semantic interpretation that will assign elements of \mathcal{M} both to elements of \mathbf{D} and to elements of \mathcal{A} formed from these in a compositional manner. This can be restated even more compactly in terms of category theory: the members of \mathbf{D} , plus all other elements of the lexicon, plus all expressions constructed from these, are the objects of some category L of linguistic expressions, whose arrows are given by the constructions and the definitional equations; members of \mathcal{M} , and the mappings between them, make up the category M ; and semantic interpretation is simply a functor S from L to M .

The key observation, which bears repeating at this point, is that S *underdetermines* the semantics of lexicalized expressions: if noun–noun compounding (obviously a productive construction in English) has the semantics ‘ N_2 that is V -ed by N_1 ’, all the theory gives us is that *ropeladder* is a kind of ladder that has something to do with rope. What we obtain is *ladder*, *rope* rather than the desired *ladder*, *material*, *rope*. Regrettably, the theory can take us only so far – the rest has to be done by diving into the trashcan and cataloging historical accidents.

Lexemes will be mapped by S on finite state automata that *act* on partitioned sets of elements of $\mathbf{D} \cup \underline{\mathbf{D}} \cup \mathbf{E}$ (the underlined forms are printnames). Each partition contains one or more elements of $\mathbf{D} \cup \mathbf{E}$ or the printname of the lexeme (which is, as a matter of fact, just another pointer, to phonetic/phonological knowledge, a domain that we happen

to have a highly developed theory of). By *action* we mean a relational mapping, which can be one-to-many or many-to-one, not just permutation. These FSA, together with the mapping associating actions to elements of the alphabet, are *machines* in the standard algebraic sense (Eilenberg, 1974), with one added twist: the underlying set, called the *base* of the machine, is *pointed* (one element of it is distinguished). The FSA is called the *control*; the distinguished point is called the *head* of the base.

Without a control, a system composed of bases would be close to a semantic network, with activations flowing from nodes to nodes (Quillian, 1968). Without a base, the control networks would just form one big FSA, a primitive kind of deduction system, so it is the combination of these two facets that gives machines their added power and flexibility. Since the definitional burden is carried in the base, and the combinatorial burden in the control, the formal model has the resources to handle the occasional mismatch between syntactic type (part of speech) and semantic type (as defined by function–argument structure).

Most nominals, adjectives, adjectives, and verbs will only need one content partition. Relational primitives such as *x AT y* ‘*x* is at location *y*’, *x HAS y* ‘*x* is in possession of *y*’, *x BEFORE y* ‘*x* temporally precedes *y*’ will require two content partitions (plus a printname). As noted earlier, transitive and higher-arity verbs will also generally require only *one* content partition: *eats(x,y)* may look superficially similar to *has(x,y)* but will receive a very different analysis. At this point, variables serve only as a convenient shorthand: as we shall see shortly, specifying the actual combinatorics of the elements does not require parentheses, variables, or an operation of variable binding. Formally, we could use more complex lexemes for ditransitives like *give* or *show*, or verbs with even higher arity such as *rent*, but in practice we will treat these as combinations of primitives with smaller arity, for example, *x gives y to z* as *x CAUSE(z HAS y)*. (We will continue using both variables and natural language paraphrases as a convenient shorthand when this does not affect the argument we are making.)

Let us now turn to operations on lexemes. Given a set \mathcal{L} of lexemes, each n -ary operation is a function from \mathcal{L}^n to \mathcal{L} . As is usual, distinguished elements of \mathcal{L} such as *NULL*, *0*, and *1* are treated as nullary operations. The key unary operations we will consider are *step*, denoted *'*; *invstep*, denoted *`*; and *clean*, denoted *-*. *'* is simply an elementary step of the FSA (performed on edges) which acts as a relation on the partition *X*. As a result of a step *R*, the active state moves from x_0 to the image of x_0 under *R*. The inverse step does the opposite. The key binary operation is substitution, denoted by parentheses. The head of the dependent machine is built into the base of the head machine. For a simple illustration, recall the definition of *mule* as *animal*, *cross between horses and donkeys*, *stubborn*, . . . So far we have said that one partition of the *mule* lexeme, the head, simply contains the conjunction (unordered list) of these and similar defining (essential) properties. Now assume, for the sake of argument, that *animal* is not a primitive, but rather a similar conjunction *living*, *capable of locomotion*, . . . Substitution amounts to treating some part of the definiens as being a definiendum in its own right, and the substitution operation replaces the atomic

animal in the list of essential properties defining *mule* by a conjunction *living, capable of locomotion, ...*. The internal bracketing is lost; what we have at the end of this step is simply a longer list *living, capable of locomotion, cross between horses and donkeys, stubborn, ...*

By repeated substitution, we may remove *living, stubborn, etc.* – the role of the primitives in **D** is to guarantee that this process will terminate. But note that the semantic value of the list is not changed if we leave the original animal in place: as long as animals are truly defined as living things capable of locomotion, we have set-theoretical identity between *animal, living, capable of locomotion* and *living, capable of locomotion*. Adding or removing redundant combinations of properties makes no difference.

Either way, further quantification will enter the picture as soon as we start to unravel *parent*, a notion defined (at least for this case) by ‘gives genetic material to offspring’, which in turn boils down to ‘causes offspring to have genetic material’. Note that both the quantification and the identity of the genetic material are rather weak: we don’t know whether the parent gives all its genetic material or just part of it, and we don’t know whether the material is the same or just a copy. But for the actual definition none of these niceties matter: what matters is that mules have horse genes and donkey genes. As a matter of fact, this simple definition applies to hinnies as well, which is precisely the reason why people who lack significant encyclopedic knowledge about this matter don’t keep the two apart, and even those who do will generally agree that a hinny is a kind of mule, and not the other way around (just as bitches are a kind of dog, i.e. the marked member of the opposition).

After all these substitution steps, what remains on the list of essential mule properties includes complex properties such as *HAS(horse genes)* and *capable of locomotion*, but no variable is required as long as we grant that in any definiens the superordinate (subject) slot of *HAS* is automatically filled by the definiendum. Readers familiar with the accessibility hierarchy of Keenan and Comrie (1977) and subsequent work may jump to the conclusion that, one way or another, the entire hierarchy (handled in HPSG and related theories by an ordered list) will be necessary, but we attempt to keep the mechanism under much tighter control. In particular, we assume no ternary relations whatsoever, so there are no such things as indirect objects, let alone obliques, in definitions. To get further with *capable of locomotion* we need to provide at least a rudimentary theory of being capable of doing something, but here we feel justified in assuming that *CAN, CHANGE, and PLACE* are primitives, so that *CAN(CHANGE(PLACE))* is good enough. Notice that what would have been the subject variables, who has the capability, who performs the change, and who has the place, are all implicitly bound to the same superordinate entity, the mule.

To make further progress on *horse genes* we also need a theory of compound nouns: what are *horse genes* if not genes characteristic of horses, and if they are indeed characteristic of horses, how come that mules also have them, and in an essential fashion to boot? The key to understanding *horse gene* and similar compounds such as

gold bar is that we need to supply a predicate that binds the two terms together, what classical grammar calls ‘the genitive of material’, which we will write as `MADE_OF`. A full analysis of this notion is beyond the limits of this chapter, but we note that the central idea of `MADE_OF` is production or generation: the bar is produced from/of/by gold, and the genes in question are produced from/of/by horses. This turns the Kripkean idea of defining biological kinds by their genetic material on its head: what we assume is that horse genes are genes defined by their essential horse-ness rather than that horses are animals defined by carrying the essence of horse-ness in their genes. (Mules are atypical in this respect, in that their essence cannot be fully captured without reference to their mixed parentage.)

6.7 Further reading

To the extent feasible, in preparing the examples for this chapter we relied on examples taken directly from [handouts](#) prepared by László Kálmán and Márta Peredy for their 2012 ‘Criticism of basic concepts in linguistics’ course. For Tuscarora and for part of speech tags in general, see Croft (2000).



Radical lexicalism, the idea that the words are the *only* thing you need to learn in order to know the language, implies that to know the grammar is to know the function words. This obviously requires very carefully crafted lexical entries for function words or, as assumed in Borer (2005) and Borer (2013), some highly specific functional patterns associated with these (leaving the door open for the kind of functional patterns like [topicalization](#) that do not attach to specific lexical items).



Until recently, computational work on POS tagging was dominated by the Penn tagset (see Section 5.2), but in the past few years a less English-specific tagset has been introduced in the Universal Dependencies model we discussed in Section 5.4. Success in the more ambitious task of POS induction is still hard to measure (Christodoulopoulos, Goldwater, and Steedman, 2010). Defining word-internal semantic relations (such as those obtaining between *rope*, *ladder* and *ropeladder*) by distributional means is a largely unresolved issue. (The algebraic considerations presented here suggest the result is not even fully defined, but this remains to be seen.)

One suggestion addressing the issue of finding the proper dimension d for an embedding is to use an infinite (in practical terms, arbitrary finite) dimension (Nalisnick and Ravi, 2015).

The treatment of ternary and higher-arity predicates proposed here, namely that they are formed by embedding binary predicates in one another, is described in more detail in Kornai (2012). As discussed in Section 5.5, the idea is not new, going back at least to [generative semantics](#). The computational motivation comes from the fact that the overwhelming majority of verbs, tens of thousands of items, have at most one or two arguments, while there are, even under the most permissive criteria, only a few hundred ditransitives and hardly any higher-arity constructions. This is well recognized in systems of knowledge representation such as [RDF](#), which treat edges between nodes as





the default case, and use a [special mechanism](#) of unnamed auxiliary nodes to describe the higher-arity cases. A similar mechanism is used in [Freebase](#), where the auxiliary nodes are known as *compound value types* (CVTs). In practice, ternary and quaternary predication is seen overwhelmingly for facts that hold true only at specified times and places, and some representation schemes such as YAGO2 (Hoffart et al., [2013](#)) have built in extra time and space slots for each predicate. We see this as a stopgap measure at best, since the problem extends far beyond temporal and spatial limitations to cases where the source of the evidence matters, or when we have probabilistic and other qualifications. Such issues are traditionally treated as modalities, and we have discussed (and dismissed) the traditional solution in [Section 3.7](#). We outline an alternative, keyed primarily to the individual words, in [Section 7.3](#).

Models

Contents

7.1 Schematic inferences	200
7.2 External models	203
7.3 Modalities	208
7.4 Quantification	215
7.5 Further reading	218

In the previous chapters we discussed how the meaning of words and larger grammatical constructions can be represented by machines. In 7.1 we assess, we an independent test that was built specifically for this purpose by Levesque, Davis, and Morgenstein (2012), how well these techniques stand up on semantic tasks. We develop a simple taxonomy of the 140+ tasks in this test set, and discuss their generality.

Since the machine method is equally capable of representing true statements such as *hens lay eggs* and false statements like *dogs lay eggs*, we may wish for something more, a theory of *truth* that is capable of telling the two apart – this is the subject of 7.2, where we outline the model theory that fits best the central themes of the book discussed in the preceding chapters: a unique, large *external model* corresponding to the real world, and many smaller *internal models* corresponding to knowledge bases in people’s heads.

In 7.3 we turn to the area where the different kinds of models have the strongest ramifications, modal logic and the study of grammatical modes. We lay the groundwork for the simultaneous study of negation, tense, and mood in a homogeneous system that minimizes the type-theoretical distinctions routinely drawn between syntactic (term) operators, semantic operators, and truth values, making the entire system depend on the everyday meaning of function words.

Finally, in 7.4 we discuss the primary means of generalizing from single instances to broader laws, quantification. Here we again put the emphasis on describing the phenomena by lodging the system in the meaning of the function words, rather than devising a special logical apparatus (predicate calculus, as opposed to the much simpler propositional logic) just in order to accommodate quantifiers.

In this chapter there are fewer of the routine practice exercises usually marked by a raised °, and more that take us into the realm of research where there is no unique

solution. The reader should experiment with these problems, primarily by building a formal or computational model that exhibits the desired properties. Such problems are marked with a raised \rightarrow or, when really challenging, by a raised \dagger .

7.1 Schematic inferences

In Chapter 3 we used McCarthy’s 1976 analysis of a simple newspaper story to derive conclusions about the absolute minimum a system must contain to answer questions that humans answer by using common sense. Here we use a more systematic test set, based conceptually on Winograd’s 1972 *schemas* and specifically designed by Levesque, Davis, and Morgenstein (2012) to replace the classic Turing test, to discuss in greater detail how the pieces of machinery we have developed in this book so far fit together. Each test problem begins with an assertion, such as

The trophy doesn’t fit into the brown suitcase because it’s too small.

To test the common sense abilities of the system, it must answer a question *Q*: *What is too small?* To avoid issues of normalization (in this case clearly *the suitcase* and *the brown suitcase* would be acceptable answers), an explicit choice of answers (in this case ‘the trophy’ and ‘the suitcase’) is provided. To control for frequency effects, an alternate assertion is also provided, in this case

The trophy doesn’t fit into the brown suitcase because it’s too big. *What is too big?*



Changing what we will call the pivot word turns the answer around: whereas in the first case the obvious answer was the suitcase, in the second case it is the trophy. This is very similar to the earlier [Recognizing Textual Entailment](#) (RTE) shared task, also composed of a small piece of text and a hypothesis that does, or does not, follow from the text that we discussed in Section 3.9, except that here great care has been taken to make sure that one cannot obtain the answer by frequency considerations. What we need to solve this problem is a *regularity* (see Section 3.6) which is quite independent of the size of trophies or suitcases:

$$A \text{ fits in } B \Rightarrow A \text{ is smaller than } B, B \text{ is bigger than } A. \quad (7.1)$$

This regularity is part of what *fit in* (and also *fit inside* and *fit into*) means. If the verb was *contain* or *envelop*, the implication would be that A is bigger than B. In general, there could be sophisticated inference chains involving real and apparent sizes (cf. ‘To see better, I covered the Sun with my palm’), but we need to address the simple problem first. As standard, we speak of *polar adjectives* when two adjectives can be placed at the opposite ends of a single scale: *big/small*, *tall/short*, *young/old*, etc. – these play an important role in building semantic spaces by the interview method (Section 2.7). The assumption common to all these is that one member of the pair is placed on the

negative, and the other on the positive side of zero on the scale, and the meaning of the comparative suffix *-er* is to take its subject farther away from the origin than the object used as the basis of comparison. We shall return to the geometry implied by these simple rules in Section 9.4; here the following simple *rule of polarity* will suffice:

$$\text{(polarity) } A \text{ is } X, B \text{ is } X\text{-er than } A \Rightarrow B \text{ is } X \quad (7.2)$$

For goal-directed behavior, we need a *rule of retreat*: if *A is too X* (for purpose/goal G), A needs to be changed to be less X to achieve the purpose. This is part of the meaning of *too* in the construction *being too X* ('being too large', as opposed to 'Joe was there too'):

$$\text{(retreat) } A \text{ is too } X \text{ (for } B) \Rightarrow B \text{ requires } A \text{ made less } X. \quad (7.3)$$

Also, in general *being too X* implies *being X* (modulo silent elements, such as those required for understanding 'enormous flea'; see Section 5.6), and this is part of the meaning of *too*:

$$\text{(L-too) } A \text{ is too } X \Rightarrow A \text{ is } X. \quad (7.4)$$

Exercise[†] 7.1 Find other pairs of polar adjectives. Are *male/female* polar opposites? Are *manly/womanly*? *Manic/depressive*? How should we handle cases where polarly opposed adjectives are used to characterize an object, as in *a great little restaurant*?

After these preparations, we are ready to sketch the solution to the test problem. If the trophy doesn't fit into the brown suitcase, this means, by the definition of *fit in(to)*, that the trophy is bigger than the suitcase. This is a result state explained by X being too small, something that we could fix by making X bigger (rule of retreat). Since the trophy is already bigger than the suitcase, making the trophy bigger would not serve our goal of fitting the trophy into the suitcase. Therefore, X must be the suitcase.

Exercise[°] 7.2 Which parts of the above reasoning change for the alternate sentence 'The trophy doesn't fit into the brown suitcase because it's too small'? *What is too small, the trophy or the suitcase, and why?*

Several questions are amenable to this simple treatment directly, as in 'The delivery truck zoomed by the school bus because it was going so [fast/slow]'. *What was going so [fast/slow], the truck or the bus?* Others require the same logic, but are one step removed: 'The large ball crashed right through the table because it was made of [steel/styrofoam]' *What was made of steel/styrofoam, the ball or the table?* Here there is no assumption that steel and styrofoam are polar opposites; what we need are pieces of commonsensical knowledge, stored in the lexicon, that (i) hard(er) things can crush through soft(er) things but not the other way around; (ii) steel is hard; and (iii) styrofoam is soft. These are lexical rules (meaning postulates in the sense described in Section 3.8) of implication.

There are two key problems here. One is *knowledge discovery*, somehow having it listed in the lexicon that *X zooms by Y* implies X is faster than Y, and Y is slower than X;

or that *X crashes through Y* implies X is harder than Y, and Y is softer than X. The other is *knowledge selection*: even if we discover that steel is hard and styrofoam is soft, it is also the case that steel is heavy, styrofoam is light, steel is flexible, styrofoam is rigid, steel is a conductor, styrofoam is an insulator, steel reflects light well, styrofoam doesn't, and so on. How do we know which of these pieces of knowledge to use? In practice, it is the knowledge discovery problem that is really hard; the *spreading activation* method discussed in Section 5.7 takes good care of the knowledge selection issue, especially if a parallel implementation is available.

Another set of problems tests what in Section 3.3 we called *naive space-time geometry*. Consider 'Tom threw his schoolbag down to Ray after he reached the [top/bottom] of the stairs.' *Who reached the [top/bottom] of the stairs, Tom or Ray?* Here *top* and *bottom* are not functioning as polar adjectives (cf. the lack of forms **bottomer, too top, ...*) but rather as functional parts of stair(way)s, with the bottom being *down* from the top, so that things thrown down will move from the person at the top to the one at the bottom. One would be hard put to endorse a model of space-time that doesn't provide the inferential axioms for this much. Yet the overall model, handling not just *up/down* but also *front/back, before/after, above/below, from/to, left/right, ...* is surprisingly hard to formulate, as it involves not just the inherent vertical given by gravity, but also the viewpoints of the speaker and hearer, and on occasion also frames of references induced by objects that have their own fronts and backs, as in the *old TV set* discussed in Section 5.1.

Many tests problems relate to naive geometry and little else: 'The sack of potatoes had been placed [above/below] the bag of flour, so it had to be moved first.' *What had to be moved first, the sack of potatoes or the bag of flour?* But for several others, getting to the realization that geometric regularities will have to be invoked is much harder. Consider 'John couldn't see the stage with Billy in front of him because he is so [short/tall].' *Who is so short/tall, John or Billy?* What we need here is a notion, evident on its face, yet quite a challenge for knowledge discovery, that seeing involves an unobscured line of vision between the object and the receptor (the eye). Once we have this, the rest falls into place, with Billy obscuring the line of sight between John and the stage if Billy is the taller of the two, and the subsequent logic steps using polarity and retreat are as discussed above.

Perhaps we can get by with an even more naive theory of vision, where being higher up means seeing better. This creates a huge potential for false positives, as there are excellent reasons for the more complicated 'line of sight' theory of vision to emerge. (In its naive form, this theory invokes rays or *glances* that emanate from the seeing eye, rather than the more modern light rays that emanate from the object and arrive at the eye.) A similar simplification is feasible in regard to *zoom*, which means being fast, a piece of knowledge much easier to acquire than the full conceptual frame for *zoom by*. Finding the right level of naivity is a problem we will return to in Section 7.2, where we argue that full numerical models (for example, for carrying out orbit calculations) are unnecessary.

Another set of tests probes what we called *valuations* in Section 3.5. Sometimes the question is direct, as in *Which name was [easier/too hard] to pronounce, Tina or Terpsichore?* In other cases, valuations come into play only indirectly, as in *Who was trying to [run/stop] the drug trade, the gang or the police?* Here we need a set of naive valuations (drugs are bad, the drug trade is bad, gangs are bad, stopping bad things is good, cops are good) to make this come out right. In fact, we have seen cases of crooked cops taking over the drug trade, but the probabilistically correct answer is not hard to obtain.

What is remarkable about these questions is that to formulate an answer we don't even need the specific background information ('The actress used to be named Terpsichore, but she changed it to Tina a few years ago, because she figured it was [easier/too hard] to pronounce' and 'The police arrested all of the gang members'); our background valuations are quite sufficient. These kinds of questions don't particularly probe our ability to understand the background assertion, just the ability to understand the question itself. Consider the instruction *Here are six weights; arrange them in ascending order.* This does probe our sensory and motor abilities, and to a very limited extent our semantic system as well, but really the focus is on the former.

To put the focus on semantics we need tasks where the interplay between valuations and meaning is more subtle. Consider 'The city councilmen refused the demonstrators a permit because they [feared/advocated] violence.' *Who [feared/advocated] violence, the city councilmen or the demonstrators?* Of course, one can imagine the council advocating violence, and refusing a permit just because a peaceful demonstration would not further this goal, but this is highly implausible: councilmen all over the world are more likely to fear than to advocate violence. What about demonstrators? On balance, it seems they would also fear violence more than advocate it. In fact, it seems very hard to acquire regularities of the requisite specificity, 'councilmen are less likely to advocate violence than ordinary demonstrators', so we must fall back on a more crude heuristic, such as the following.

Violence is bad \Rightarrow advocating violence is bad \Rightarrow people who advocate violence are bad. Permits are good. People don't give good things to bad people. Thus, if the pivot is *advocate*, it is clear *why* the permit was refused, and from this it follows that it is the demonstrators who advocated violence. By presuming that *because* in the assertion is actually signifying a causal relation (the second clause answers a *why* question), we are taking advantage of the Gricean Maxim of Relation we discussed in Section 5.6.

Exercise^o 7.3 Which parts of the above reasoning need to be changed if the pivot is *fear*? Do we need additional regularities?

To summarize, solving the problems presented in Levesque, Davis, and Morgenstein (2012) require three broad strategies, each relying on an important facet of common-sense reasoning: understanding how polar adjectives work, being capable of working with naive geometry and other regularities, and value propagation. For some tasks, more than one of these is required, but the deductive chains are surprisingly short. The pri-

mary difficulty is not in applying the rules, it is in populating the rulebase, what we have called the knowledge discovery problem.

7.2 External models



The most well-developed theory of truth is called the [correspondence theory of truth](#), which states, roughly, that a statement is true iff it corresponds to an objective fact of the world. In [Chapter 2](#) we have seen how *model structures* built from sets take the role of the worlds that we take into consideration, and how the interpretation relation takes the role of correspondence. In [Section 3.7](#) we discussed how the standard theory of semantics employs an extended (intensional, modal, and temporal) version of this basic model-theoretic setup, leaving out the grounding relation g from model structures to the objective world. Here we develop a more complex theory, which distinguishes external from internal models, and is capable of carrying out various kinds of inferences.

By *external models* we mean autonomous systems with outputs (and possibly inputs as well, but this is not required). There is no requirement that external models be finite automata, transducers, or machines, in fact there is no requirement for the outputs to be taken from a finite or discrete set to begin with. A typical example would be the Solar System: it has no inputs, in that we are not capable of affecting planetary motion, but it autonomously provides output about the position of the planets. A more modest example would be a [software orrery](#) which provides the same outputs, within reasonable error bounds. In general, any software system that provides outputs either autonomously or as a result of providing it with inputs will be considered an external model.



Since natural language syntax and semantics are discrete, we may need some [A to D converter](#) to deal with the outputs of the external model. If we are interested in manipulating an external model, we may also need a [D to A converter](#) to provide it with inputs. From the standpoint of semantics, such conversions are outside the model, and are handled by adding external pointers (see [Section 4.5](#)) to lexemes. In general, the problem of how we convert a sight or a smell to a linguistic representation is extremely complex, and there is a whole area of scientific study, [pattern recognition](#), devoted to this issue. For the framework discussed here, it makes no difference whether external models come complete with their A/D conversion or whether the converters are separately tweakable modules – either way, we assume discrete (discretized) outputs and inputs.

Returning to our initial example, the correspondence theory of truth dictates that we consider the sentence *Pluto is in Cancer* true iff the planet (or dwarf planet – classification issues make no difference) Pluto appears in the 30 degrees of arc designated Cancer. The statement was true between 1913 and 1938, and will be true again for 25 years starting in 2161. To make the statement absolutely true, we need to specify the time: *In 1930 Pluto was in Cancer* is true; *In 1940 Pluto was in Cancer* is false. Either we think of external models as having a time parameter, or we treat these as families of models indexed by the real numbers. While the second approach is more common in formal semantics, to make it equivalent to the first one would require careful mainte-

nance of objects and relations across time indexes, needlessly complicating the inference mechanism. We therefore admit a (discretized) time parameter (see Section 3.3) but note that it is only the orrery that makes this parameter conveniently settable, the actual Solar System follows its own time. From this it follows that absolute truths can only be eternal (for example, statements of arithmetic) or be about facts of the past: the purported fact that Pluto will be in Cancer in 2180 is just a prediction, there is a non-negligible probability that some giant asteroid may come and change Pluto's orbit enough to render this prediction false. Those with a truly skeptical mindset may also question facts about the past. About 50% of the US population believes in [Young Earth creationism](#), which asserts that the Universe was created, complete with the fossil record and with starlight arriving from very distant objects, in six days, some six thousand years ago.



Assuming the existence of a single large, distinguished external object, the *real world* or *objective reality* (this is a large assumption, but one routinely made in [Western philosophy](#)), establishing the correspondence-theoretic *truth* of a statement becomes a matter of checking it in the real world. This is often not a trivial undertaking, especially about past or future events, but the entire human epistemological apparatus (except possibly for mathematics, whose truth we will discuss shortly) is geared to this.



A much smaller assumption, and one largely independent of the big philosophical issue of objective reality, is the following: human cognition assumes an internal model of objective reality. With this smaller assumption we can go beyond truth, and express what it means to *lie*: a statement by person X is a lie iff it is inconsistent with their *internal* model of objective reality. This is also hard to check, but all legal theories of culpability are built on this: to lie is to speak untruth relative to one's internal model, not relative to external reality.

Exercise[→] 7.4 Plato reports that [Artemis](#) lives on [Mount Olympus](#). Was he lying? His evidence is the [Iliad](#). Was Homer lying? Today, if I'm telling you that Artemis lives on Mount Olympus, am I lying? Provide formal derivations justifying your conclusions.



External models with inputs typically verify implicative statements: if the input is x , the output is y . The matter is greatly complicated by the fact that outputs may depend not just on inputs but also on the state of the external model and possibly on hidden inputs that it receives from other sources. When the external model is a stateless transducer with all inputs visible, it models a strictly causal relation: for any input x given to the model at any time, we always obtain the same output x^C .



Exercise[◦] 7.5 Does the converse of the above statement hold? Suppose some system C will deterministically produce output x^C every time it receives input x . Is C a stateless FST? Why or why not?

Exercise[†] 7.6 Consider some clear example of causation, such as *Malaria is caused by plasmodium*. Is this the same as *Plasmodium causes malaria*? Why or why not? Design an external model capturing the facts.

One domain of particular importance that we consider external is the whole arena of quantitative reasoning, starting with integer arithmetic and going on to computations



with rationals of arbitrary size and with reals. Any standard package for **bignum arithmetic** is well outside the internal semantic capabilities we will discuss in Chapter 8 and must therefore be treated as external. To the extent that quantitative reasoning has been a prerequisite for doing physics ever since Galileo, this is a severe restriction indeed. But in order to understand how natural language expressions convey information, we must leave artificial systems out of scope. Arguably, there are some faint traces of **numerosity** in language, but these are very far from the kind of arithmetic we all learn in school. When we say *three times seven is twenty-three*, the truth or falsity of this statement can be checked against an internal model that is committed to memory by pupils early in their schooling, the **multiplication table**. But when we say $340355/17620 = 19.316401816118047673098751418842424744608399545970488081725312145 \dots$, this is obviously not done by rote memorization, and utilizes skills generally absent from the majority of the adult population in spite of schooling.



Exercise^o 7.7 Verify or falsify the above arithmetic statement. How many digits long is the periodic part of the decimal fraction?

One particularly strong reason for avoiding arithmetic in semantics is that statements in arithmetic are actually not subject to verification or falsification by ordinary means. This has already been noted by Ayer (1946):

It might easily happen, for example, that when I come to count what I had taken to be five pairs of objects, I find that they amounted to only nine . . . But [. . .] one would not say that the mathematical proposition $2 \cdot 5 = 10$ had been confuted. One would say that I was wrong in supposing that there were five pairs of objects to start with, or that one of the objects had been taken away while I was counting, or that two of them had coalesced, or that I had counted wrongly. One would adopt as an explanation whatever empirical hypothesis fitted in best with the accredited facts. The one explanation which would in no circumstances be adopted is that ten is not always the product of two and five.



In a straight model-theoretic account, a single failure in the model where Ayer did his counting would be sufficient to render the proposition *two times five is ten* false, and in empirical science we generally uphold the same standard. For an example, consider how **Torricelli's experiment** destroyed the well-entrenched idea of **horror vacui**. To be sure, it was not a single occasion of Torricelli claiming to have produced a vacuum that enabled this change, but rather the fact that his experiment was freely replicable by others, and always led to the same results. If we had the means to replicably create five pairs of objects that add up to nine, this would no doubt lead to a revolution in arithmetic as well.

When we simplify and discretize physical systems to provide us with external models of phenomena that will fit into the FST class of models, we are no doubt losing some precision. But as Exercise 3.7 on page 56 shows, even the historically most precise human activity, calendar making, is easy to bring into scope for FST models with negligible loss of precision.

Exercise[→] 7.8 Extend the model of Exercise 3.7 to cover [leap seconds](#).

While a certain amount of scientific model-building is possible with FSTs, the semantic ability to go in the direction of *less precision* while still retaining meaning is actually more relevant to our understanding of language use. Consider the following set of instructions from [rampantscotland.com](#):



Grease an 8-inch loaf tin. Rub the fats into the flour and salt and then mix in enough cold water to make a stiff dough (remember, it is going to line the tin). Roll out the pastry and cut into six pieces, using the bottom, top and four sides of the tin as a rough guide. Press the bottom and four side pieces into the tin, pressing the overlaps to seal the pastry shell. Mix the raisins, currants, almonds, peel and sugar together. Sift in the flour, all the spices and baking powder and bind them together using the brandy and almost all the egg and add enough milk to moisten. Pack the filling into the lined tin and add the pastry lid, pinching the edges and using milk or egg to seal really well. Lightly prick the surface with a fork and make four holes to the bottom of the tin with a skewer. Depress the centre slightly (it will rise as it cooks). Brush the top with milk or the rest of the egg to create a glaze. Bake in a pre-heated oven at Gas Mark 3 for 2 1/2 to 3 hours. Test with a skewer which should come out clean; if not, continue cooking. An uncooked cake sizzles if you listen closely! Cool in the tin and then turn onto a wire rack. Cool thoroughly before storing until Hogmanay.

Exercise[†] 7.9 Build a model capable of describing the process of making black bun. Assume input conditions such as *a generous pinch of black pepper* have all been met. How robust is your model? What happens if you depress the center strongly, not just slightly?

In addition to modeling everyday tasks such as cooking, and everyday phenomena such as the weather, the ability to treat people (both other people and oneself) as an external model is crucial for getting any semantic theory off the ground. In Sections 5.6 and 5.7 we discussed why the ability to comprehend what others think is required for even the simplest interactive tasks. Here we draw out a central implication for the logic that is supported by such models, namely that systems of rational belief are not necessarily transitive: if we (rationally) believe $A \Rightarrow B$ and $B \Rightarrow C$, it doesn't follow that we must believe $A \Rightarrow C$.

This is best seen in cases of *rational prejudice*, where we formulate the conditional probability statement $P(B|A) \gg P(B)$ as the defeasible implication $A \Rightarrow B$. When we look at the phenomenon at the root of a rational prejudice such as *basketball_player \Rightarrow tall*, what we see is that the average height of basketball players exceeds considerably that of the general population and (what is the same) our chances of finding a tall person among basketball players are much higher than our chances of finding one among the entire population. The implication is defeasible; obviously, we have basketball players of average height or even shorter, dogs that don't have four legs, etc. Much of the apparatus discussed in Section 7.1 above instantiates various forms of rational prejudice, and there

is no denying that commonsense reasoning relies on such implications more often than on the pure form of logic described in Sections 2.4 and 2.5.

Exercise° 7.10 Given three measurable sets A, B, C in a measure space with probability measure P , can we have $P(B|A) > 2P(B)$; $P(C|B) > 2P(C)$, but $P(C|A) < 2P(C)$? Why or why not?

Phil One final note concerns the reflective particle *self*, often elevated to special status in the philosophy of language, especially when it comes to consciousness and self-awareness. Here we make the rather commonsensical assumption that everybody's internal model contains a sub-model, imperfect and incomplete in many respects, and by no means free of falsehoods, of the external world. In this sub-model there are persons, and among these persons there is a distinguished one, I, the self. As we assume that other persons also come with internal models similar in their main outline to ours, we may very well assume that the self also has its own sub-model of external reality, complete with its own homuncular self. But because we don't model others' models recursively, there is no reason to assume we model ourselves recursively, so there is no paradox of infinite regress.

7.3 Modalities



Historically, *modes* were invented in the 13th century to handle discrepancies between grammar and logic, and to this day there is a somewhat uneasy interplay between [grammatical mood](#) and the theory of [modal logic](#) that we discussed in Section 3.7. The unease stems in part from the fact that different languages employ different moods, while logic is presumably universal, and in part from the fact that the same linguistic markers, such as choice of auxiliary and morphology, may indicate what to the logician look like very different modes, for example a wish, a possibility, or a blessing. We can see some kind of conceptual relatedness between these ideas, for example, a blessing is a good wish directed to the blessed (though this doesn't quite exhaust the meaning of blessing), we generally assume that what we wish is possible, and so on, but this kind of relatedness is very far from the monosemic ideal we are pursuing.

We begin our discussion with *negation*, even though both philosophers and linguists are split on the issue of whether negation constitutes a modality. What is clear is that the evidential basis for negative propositions is very different from that for positive ones: we can see Billy in the kitchen, but we cannot see, in fact we cannot even imagine, Billy *not* in the kitchen. We may imagine an empty kitchen, but this is evidence for an infinite number of propositions, with Joey not in the kitchen, Stevie not in the kitchen, and so on. From seeing the empty kitchen we can conclude that Billy is not there, but this is an act of deduction. To be sure, it is a highly automatic and largely unconscious act, but this is true of almost all sense-making activity.

By considering mind states to be external models, either directly or via some discretized finite state approximation, we obtain a theory of falsity (lies) that is no longer a simple Boolean dual of the theory of truth. The correspondence theory is formulated

via grounding (see Section 3.7) in a single distinguished external model, the real world, while falsity is viewed as lack of coherence with the internal model of the speaker.

Several interesting conclusions follow; we name only two here. First, a voice recording of some statement X cannot lie: we need a person, capable of holding an internal model of the world, for uttering a lie. But the same recording can be true (valid in the external world). Since we can synthesize speech saying anything without there being a person who actually said the statement, it follows that there are statements that are false, without being lies. Second, as we briefly discussed in Section 2.7, the informational value of positive and negative statements is not at all symmetrical. On the whole, we are more interested in truth than in lies or inadvertent falsities, since we have a strong sense that there are only a few (often, only one) truths to be said, while there are many ways of being wrong.

Based on what we have said so far, we can actually justify this intuition, starting with the lexical entries, which contribute the bulk of the meaning, as discussed in Section 1.3. These entries are typically simple conjunctions of elementary statements, and tend to list only positive facts. It is true that on occasion we see negatives in definitions, but this is surprisingly rare (fewer than 3%), and is typically related to the absence of an expected property, as in *Miss*, a term of address for women of marriageable age who lack a husband, or in *empty*, ‘has nothing in it’.

Exercise[→] 7.11 Define *passenger*. How do you account for the fact that a passenger is not the driver and not part of the crew? Can you define *crew* without reference to the driver or pilot?

To deal with these 3% of cases, we introduce two more truth values in addition to the standard T (true, \top) and F (false, \perp), called U (unknown) and D (unDecided). Negation, as standard, makes F out of T and T out of F. In the extended logic 4L presented here, the negation of U is U: this is clearly related to the ‘neither’ value of Belnap (1977), but our interpretation is closer to that of Codd’s ‘missing data’. From the modal logic standpoint, U is already deeply connected to the epistemic modality. Modal logicians are most concerned with iterations of the modal operator, for example the fact that we may know something without knowing that we know it (which is essentially Plato’s position in *Meno* about all knowledge, see Chapter 3). Our chief concern here is recapturing the commonsensical theory of knowledge, for example the fact that if we see something we know it (with the usual caveats about our senses possibly deceiving us). One terminological problem we wish to avoid here concerns *knowing*, a term that many thinkers would like to use in contradistinction to *believing*, defining knowledge as true belief.

Exercise[†] 7.12 Define *knowledge* and *belief*. Do these two definitions coincide? Why or why not?

The other nonstandard value, D, has to do with agentive decisions and free will in the sense already discussed in Section 3.4. We adhere to a drastically simplified discrete model of time, with primitives BEFORE and AFTER attaching to all events. At any given time, the truth of a statement may depend on our own decision. Tomorrow morning I may drink tea, or I may not; the matter is unsettled in all theories of free will, except in



the denialist version, which takes all such matters to be deterministically set in advance. In 4L logic the negation of D is D: if I am undecided about something I must perforce be also undecided about its negation. D means a nondeterministic transition, AFTER, to T or F, but not to both, and in this regard it is not at all like the ‘both’ value of Belnap’s paraconsistent logic. To see this, consider the truth tables given in Table 7.1, which define the operations \neg, \wedge, \vee .

T U D F	\wedge	T U D F	\vee	T U D F
F U D T	T	T U D F	T	T T T T
F U D T	U	U U D F	U	T U D U
F U D T	D	D D D F	D	T D D D
F U D T	F	F F F F	F	T U D F

Table 7.1. Boolean operations in 4L

The logic 4L does not easily fit into the general system of n -valued logics proposed by Łukasiewicz or Gödel, even if we keep open the possibility of adding further values corresponding to degrees of knowledge or decisiveness. The main reason is the lack of linear ordering. By taking F to be the bottom and T to be the top element of some ordering, even in multi-valued logic our preference is for some system where the bottom is mapped onto 0, the top onto 1, and the intermediate elements onto some numerical values between 0 and 1. In such systems, negation is typically some monotone decreasing function that maps 0 to 1 and 1 to 0; conjunction amounts to taking the minimum of two values, and disjunction to taking the maximum.

If we try to fit 4L into such a system, we run into the following difficulty. Conjunction would require $U > D$ to make $D = U \wedge D$ come out as $\min(U, D)$. Disjunction would require $U < D$ to make $D = U \vee D$ come out as $\max(U, D)$. Equating U and D will not make this come out right, and neither would abandoning total ordering in favor of a partial ordering help, since in such systems, if U and D are incommensurable, their meet and join would come out as \top and \perp .

Exercise^o 7.13 Verify that the standard associative and commutative laws of two-place Boolean operators hold in 4L. Show that the *laws of absorption*, $(x \vee y) \wedge x = x$ and $(x \wedge y) \vee x = x$, sometimes fail. What about distributivity and modularity?

One aspect of 4L worth special discussion is its relation to defaults. Defeasible lexical values default to T: if ‘can fly’ is part of the definition of *bird* and x is an individual or subspecies of bird, we assume ‘ x can fly’ without argumentation. Yet penguins don’t fly, and this somehow needs to be made part of their lexical entry, a matter we have already discussed in Section 4.5 for the example of mules, living beings that, contrary to expectations, do not replicate. To handle such phenomena, including the issue of [prototypicality](#), we introduce two further constants (as we shall see, these are not additional truth values), K and S.

In Section 2.4 we introduced a distinction between a *predicate*, a statement which requires a subject, such as ‘boring’ or, better, ‘is boring’, and a *proposition*, a statement



that does not require a subject, either because it is interpreted as having universal force (for example, $x^2 \geq 0$) or, more frequently, because it already has a subject. The negation discussed in Table 7.1 referred to negating the main predicate of a statement, while leaving the subject intact: when we assign the truth value U to a proposition to *Mars can sustain life*, the negation *Mars can not sustain life* is also unknown, for if we had an answer to it, be it positive or negative, we would also have an answer to the original question. But we can also negate the statement that something is U(nknown), to yield *It is not unknown whether Mars can sustain life*. In general, the two statements about some proposition p , $U(\neg p)$ and $(\neg U)p$, assert very different things, and we will use K to mean *known*, the opposite of *unknown*. Similarly, we will use S to mean *settled* (decided) as the negation $\neg D$ of undecided.

In the standard analysis of necessity that we discussed in Section 3.7, the necessity operator \Box and the possibility operator \Diamond are connected by the following duality:

$$\Box p \leftrightarrow \neg \Diamond \neg p. \quad (7.5)$$

What makes this nice duality (and its dual, $\Diamond p \leftrightarrow \neg \Box \neg p$) possible is that necessity and possibility are thought of as absolute notions: if something is necessary it is necessary by virtue of the way things are, whereas if something is known to Bill it may very well be unknown to Joe. To make it absolute, we may invoke some all-knowing individual, as was standard in medieval logic, or some kind of collective wisdom, as we do today when we say *It is not known to science whether Mars can sustain life*. Today, our personal experience of God is weak, and few logicians would dare to step in the footsteps of their medieval predecessors and declare, at an axiomatic level, what an omniscient being would or would not conclude. This being the case, we will concentrate our discussion on science, and assume that with K and its negation U we refer to a slowly evolving body of knowledge, a distinguished inner model s , established by the scientific method.

First, we note that it is not at all the case that s is consistent. In fact, it harbors well-known contradictions, for example between the theory of relativity and quantum theory, and much of the scientific quest consists in searching among the ‘unknown unknowns’ to resolve these contradictions. Second, we do not at all rely on *ex falso quodlibet* in regard to s , since in practice nobody draws the radical conclusion ‘our scientific knowledge must be thrown out in its entirety’ from the presence of such contradictions. Third, this particular model has been built with great care to exclude statements that would involve D or S; there is nothing there that is a matter of agentive decision. Finally, note that s contradicts in many places the more shallow but broader collection of statements n that the naive world-view encompasses. One particularly important area is taxonomy: in n *whale IS_A fish* is true, but in s *whale IS_A mammal* holds. The approach that our civilization has taken is to systematically defer to s when there is a conflict: we will say *technically* tomatoes are fruit (as opposed to vegetables, as n would have them), and so on.



Lexical defaults and the laws of default inheritance through IS_A links play a key role in semantics. In the standard approach it is something of a mystery how a *cup* could be defined as a ‘usually open bowl-shaped drinking vessel with or without a handle’. Everything comes with or without a handle. What we say is that having a handle is an essential (analytic) property of cups, and this alone is sufficient to justify the lexical listing. Rather than saying the property ‘has handle’ is simply *true* of cups, we say it is *known* for them: *cup* HAS *handle* is listed in the model **n**. But things that we know are subject to revision (this is equally true of statements in **s**, except that the criteria for revision are more stringent there), and seeing a handleless cup or a flightless bird is insufficient for triggering system-wide revision of an otherwise well-entrenched category.

There are high-level attributes like size, shape, or color, that are inherited very broadly, at least by all physical objects, and often by abstract objects as well. While it is certainly true that everything comes with or without such properties, this is not necessarily listed in their lexical entry: physical objects may inherit these from their genus (called *physobj* in AI), but abstract nouns generally will not. Statements of the ‘with or without’ type are best translated as ‘possibly lacking’. For the core predicate we will use the primitive *lack*, which we take to be one-argument, akin to *red* or *sleep*. Thus, *blind* is defined as ‘person, lacking sight’.

Exercise → 7.14 Derive the meaning of expressions like *blind faith* and *blind fate*. Did you have to revise the definition of *blind* given above?

With this we come to *possibility*, the modality standardly denoted by the \diamond operator. It should be clear from the foregoing that some proposition *p* will be considered possible by a person whose inner model is **z** iff *p* can be added to **z** without the need for radical revision. Thus, children below a certain age will have no problem with presents coming from Santa Claus, but as their store of knowledge **z** grows, revising it to exclude Santa becomes increasingly the only option. For adults, propositions requiring major revisions of **n** are considered implausible, and those that would require major revisions of **s** are considered impossible.

Note that, in this treatment, the duality (7.5) between possibility and necessity we discussed in Section 3.7 is lost. Let us consider the cases of the naive worldview **n** and the scientific worldview **s** separately. In ordinary language, necessity pertains primarily to future events: when we read instructions telling us that a dry and rust-free surface is necessary for the paint to adhere properly, what we learn is that we must remove the rust first, at least if we want the paint to stay on. It is continuations of the buyer’s **z** that are directly in scope for this necessity, and we need a bit of further deduction to extend it from one buyer to any buyer, and thus to **n**. Technically, the statement is untrue, for there may be ways to treat the surface by heat or some chemical that would make the paint adhere without removing the rust, so in fact worlds where the negation of *p* holds are quite possible, and (7.5) fails. Because in scientific language, mathematics in particular, necessity is tied to axioms, if (7.5) holds there, it follows that we can never fully assimilate **n** into **s**.

One issue that we need to confront concerns normal modalities. Recall from Exercise 3.15 on page 74 that we call a modal system *normal* iff it follows from A being a theorem that $\Box A$ is also a theorem. This is the ‘Rule of Necessitation’, and what it means (under a host of secondary assumptions that we need not go into) is that our deduction methods have the strength not just to conclude things, but, once some conclusion has been reached, to conclude also that this is necessarily so. For Aquinas, the rules or regularities we called ‘laws of nature’ in Section 3.5 are necessary because God wills them so, but the rules of logic bind God just as well as they bind us, so whatever we deduce by means of logic are necessary. For a variety of reasons, we are reluctant to invoke God in the same way as medieval thinkers have, and if we were to invoke science instead, we would have to take a more permissive stance, in that we don’t exactly know which things are necessary and why.

In mathematics, we routinely encounter things that are true, and provable in a stronger system of axioms but not in the original set. The pioneering examples all involved some kind of coding (Gödel-numbering) and self-referentiality, but over the years attention has shifted to cases like [Goodstein’s Theorem](#) which lack any such aspects. These furnish examples of propositions that are knowable (inasmuch as we can gather at least preliminary knowledge by an apparatus we don’t fully trust) but not provable. To mix Aquinas’ language with that of science, suppose God’s will sustains [second-order arithmetic](#) (Z_2) but we are not cognizant of this, putting our trust in a lesser system such as [Peano arithmetic](#) (PA). Under such circumstances, Goodstein’s Theorem is necessary (God’s will, embodied in Z_2 , sustains it) but we cannot even prove it, let alone prove its necessity. The opposite case, when we can prove that some p is necessary, even though in fact p is contingent, is so common that we treat it as trivial: this is always the case when we simply posit p as an axiom. If you recall Exercise 3.10 on page 68, we can always extend a semigroup to a monoid by adding a unit element. In monoids, a unit element is necessarily present (the proof, such as it is, relies on the unit axiom), but in semigroups it is not.

In summary, not even the most logical part of **s**, mathematics, offers a clean identity between theoremhood and necessity. We can build logical systems, normal modal calculi, that enjoy this property, but many widely used systems of deduction are not normal, and normal systems do not scale well. The situation is equally complex in physics, where our standards of argumentation center on empirical observables. Suppose that some physicist offers an *ab initio* calculation of some measurable value that does not agree with what is actually measured. We may fault the calculation and we may fault the measuring equipment, but it is unlikely in the extreme that the calculation will simply be accepted over the actual measurements. (This is not any different from **n**, where deductive conclusions are rarely considered superior to direct sensory evidence.) Deductive truth is strongly coupled to the axioms and the deductive apparatus, and from this standpoint the main difference between **n** and **s** is that in the latter we follow a strongly minimalistic esthetic. To get a better handle on scientific deduction, we would need to



replace the necessity operator \Box by a large family of operators $N_{\Psi, \Phi}$ that makes the dependence of necessity on both the axioms Ψ and the logical apparatus Φ explicit.

Exercise \rightarrow 7.15 Define $\Psi_1 \leq \Psi_2$ to hold between systems of axioms Ψ_1 and Ψ_2 iff, holding the logical apparatus Φ fixed, every statement p that follows from Ψ_1 also follows from Ψ_2 . Is this independent of the choice of Φ ?

Exercise \rightarrow 7.16 Define $\Phi_1 \leq \Phi_2$ to hold between systems of deduction Φ_1 and Φ_2 iff, holding the set of axioms Ψ fixed, every statement p that follows via Φ_1 also follows via Φ_2 . Is this independent of the choice of Ψ ?

One issue complicating matters is that the strengths of Φ and Ψ are to some extent fungible: we routinely accept infinite axiom schemas just to avoid higher-order logical constructs. To see how this plays out in natural language, we will need to provide an account of quantification, a matter we defer to Section 7.4, but we emphasize here that the difference between propositional calculi, which get by without quantifiers, and predicate calculi, which rely on quantifiers at every turn, is less strict in natural language than in the standard theory presented in Section 3.7. The words that shows this most clearly are the *pronouns* (I, you, . . . , my, your, . . .) and the *indexicals* (here, there, now, other, . . .).

Syntactically, a pronoun is not very different from a proper noun, as it can appear mostly in the same positions, often with the exact same meaning: compare *John found a bird entangled in the wire fence. John freed it* with *John found a bird entangled in the wire fence. He freed it* or *He freed the bird*. A sentence with an indefinite pronoun *Everyone finding a bird entangled in a wire fence would free it* would naturally be translated by a formula that includes quantified variables, $\forall x \text{ person}(x) \forall y \text{ entangled-bird}(y) \text{ find}(x, y) \rightarrow \text{free}(x, y)$, but this is quite problematic in that superficially we are talking about just *a* bird, not *any* bird. Similarly, indexicals like *tomorrow* take their meaning according to the context they are uttered in, meaning Wednesday on Tuesday, but Friday on Thursday. This phenomenon is not very different from the situation that we discussed above for modalities, that we need to explicitly index them for Ψ and Φ .

That such indexing is necessary for U is evident, for we can always ask, unknown to whom, and when? The situation is even more clear for D, undecided by whom, and when? In the epistemic case, what is at stake is updating the inner model \mathbf{z} of the speaker, which may never happen, and if it doesn't happen, the matter is treated by the defaults (F for unknown, T for known). In the case of decisions freely entered into, the modal aspect is evident, but we lack a well-established name for the modality: *desiderative*, *optative*, *potential*, and in the special case where the subject is someone else, *imperative* (for positive statements) and *injunctive* (for negatives) suggest themselves, but we will coin a new cover term, talking about 'decisive' mood and modality. Ordinary expressions like *she set her mind on (doing) x*, *she decided to do x*, *she resolved (herself) to do x*, etc. are clear instances.

What is decided is not necessarily done, for the will might falter, or circumstances may intervene. To understand and properly handle such failure modes of Settled is key to building realistic action logic, but here we confine ourselves to an analysis of the straight (failure-free) case, because this already requires a rather strong piece of modal apparatus,

a *split* on the timeline. Assume S(tomorrow morning I will drink coffee). Since I am a creature of habit who regularly drinks tea in the morning, this is a positive decision. Using units of a day in discrete time, when I look back on the external model of the world at $t + 2$, the day after tomorrow, I will find that, indeed, at $t + 1$ I had coffee. It is also the case that without the agentive decision taken at t I would not have had coffee. So, from the perspective of t , we see the *external* model splitting at $t + 1$. My internal model is already settled for $t + 1$ at t ; I will not be surprised when the coffee-drinking takes place, but those living in the same household will be, unless I pre-announce my decision (in which case they will not be surprised, for they know I am a man of my word).

Such splits remain compatible with a deterministic worldview only under the radical interpretation that my free will in this matter is an illusion; I am just fooling myself into believing that my S(tomorrow morning I will drink coffee) changed anything. In defense of the more commonsensical view adopted here that takes free will to be a given, it should be mentioned that, in the deterministic theory, I am fooling not just myself, but my whole family or, depending on the scope of my announcement, an arbitrarily large group of people. If free will is an illusion, it is one massively shared in our culture, and so deeply embodied in the logic of **n** that it is worth elucidating even if **s** will ultimately dispose of it.

Exercise[†] 7.17 Build internal models with discrete timelines that split on decisions. What is the default of D? What is the default of S?

In a world where both free will and necessity are present it is inevitable that the two will clash in certain situations, with the predictable victory of necessity over human decision. Under one conception going back at least to Hegel and Marx, large-scale historical events are inevitable, and individuals, even kings and generals, have only very limited ability to change the course of events. At such points, and perhaps also at points where no necessity was involved, worldlines can *merge*. In this conception, there is no backwards necessity/determinism: the current situation could arise from different predecessors. Not only can we not predict the future, we also cannot postdict the past.

Exercise[†] 7.18 Can you time-reverse the models you built for Exercise 7.17? Why or why not?

Exercise[†] 7.19 The bulk of semantics deals with sentences in the *indicative* mood. Use the concepts of updates to the external and internal models to deal with the *interrogatory* mood. Do you need further primitives? Do you need separate primitives for yes/no questions and *wh*-questions?



7.4 Quantification

In physics, chemistry, and the sciences in general, to *quantify* results has come to mean assigning numerical quantities to them. A standard (and by tenth-graders much hated) example is provided by [stoichiometry](#), where the student must learn that to compute how



much oxygen is needed to produce just water from 1 g of hydrogen, with neither oxygen nor hydrogen left over, we first need to convert the quantity of hydrogen given in grams to moles, recall that the formula H_2O for water dictates two H atoms for each O atom so that we need to halve the amount of moles, and convert the resulting molar quantity of oxygen back to grams. If you remember that the atomic weight of O is 16 and that of H is 1, we can get away from the moles, and conclude that the weight ratio must be 2:16, so you need 8 g of O.

In linguistics, our interest is in *quantifiers*, of which *7.9367 grams* and *exactly half a mole* are poor examples, even though they work very well to quantify the amount of oxygen needed. Measure phrases like *a generous pinch* of black pepper (recall Exercise 7.9 on page 207) are more typical, but the prototypical examples favored by linguists are even vaguer, with pride of place taken by *some*, somewhere in between the extremes *none* and *all*. To see how they work out in logic, consider the following puzzle from Carroll (1896):

No one takes in the *Times*, unless he is well educated.
 No hedgehogs can read.
 Those who cannot read are not well educated.

The conclusion to be drawn, that no hedgehogs take in the *Times*, is easy enough to reach in predicate calculus using relations such as $\text{can}(x, y)$, $\text{take_in}(x, y)$, $\text{hedgehog}(x)$, $\text{well_educated}(x)$, and the axioms

- (i) $\forall y \text{ take_in}(\text{Times}, y) \Rightarrow \text{well_educated}(y)$
- (ii) $\nexists x \text{ hedgehog}(x) \wedge \text{can}(x, \text{read})$
- (iii) $\forall x \neg \text{can}(x, \text{read}) \Rightarrow \neg \text{well_educated}(x)$

For those familiar with FOL or standard logic-based knowledge representation languages, the translation from English to such formulas is effortless and automatic. But embodying in an algorithm the knowledge of what exactly it is that they are doing in the translation process is a highly nontrivial task, one that occupied Montague grammarians and computational linguists from the 1970s to the 1990s (Hauenschild, Huckert, and Maier, 1979; Landsbergen, 1982) until they essentially gave up. The main difficulty was that English syntax is very complicated, but equally important was the realization that the payoff is minuscule, essentially restricted to clever puzzles, while actual computational linguistic tasks such as machine translation are not at all advanced by this kind of analysis. In Section 7.1 we have discussed many examples that are today generally seen as more relevant for advancing the state of the art.

That said, there is still interest in simplifying the mechanism, especially in regard to formulas such as (i) that are obtained by silent contraposition between the *No one* of the text and the \forall of the formula. In FOL there is a comfortable dualism between \forall and \exists , very similar both structurally and in content to (7.5), going back to the De Morgan laws we discussed in Section 2.1. Our system 4L steers more closely toward the ‘natural’ logic of Aristotle and the Schoolmen (who all refrained from treating negation as an



involution as Boole did), and we take *exist* to mean primarily ‘exists in the real world’, i.e. in the unique distinguished external model, the real world that both the naive **n** and the sophisticated **s** aim at describing. It follows that \exists can refer to two very different things, lack of existence in the real world, and lack of existence in some internal model.

Lack of existence in the real world, like *the present king of France*, can generally be remedied by some act of creation; indeed, the definition of *create* is ‘bring into existence’ or, in the more formulaic language used in the `4lang` dictionary, ‘AFTER exist’.

Exercise \rightarrow **7.20** Does the definition of *create* involve a clause ‘BEFORE exist[lack]’? Why or why not?

Lack of existence in internal models is very hard to argue for, especially as we do not require internal models to have consistency. On the other hand, *not all* normally refers to exceptions, rather than to empty scope, so when we say *not all dragons breathe fire* we presume the existence both of dragons in general and of the fire-breathing variety in particular.

The careful reader will have noted that to deal with Lewis Carroll’s puzzle we don’t actually need the full power of predicate calculus: a much simpler propositional solution exists, involving only propositions (which we can think of as sets) such as *takes_in_the_Times*, *well_educated*, *hedghegog*, and *can_read*, and instead of (i)–(iii) we have subset relations

(i’) $\text{takes_in_the_Times} \subset \text{well_educated}$

(ii’) $\text{can_read} \subset \neg \text{hedghegog}$

(iii’) $\neg \text{can_read} \subset \neg \text{well_educated}$

Peirce already noted that simple subset logic, combined with a simple treatment of negation, is sufficient to carry the central cases.

The general notion of *polarity*, positive or negative, needs to be extended to the quantifiers and the positions they appear in. In Section 7.1 we discussed polar adjectives, and we defer polar adverbials to Section 8.3; here we concentrate on the quantifiers. We begin with `IS_A`, which is conventionally assumed monotonic: every white horse is a horse. This is, we hasten to add, a matter of convention: the Chinese sophists, **Kung-sun Lung** in particular, took a strict Leibnizian position that for two things to be identical they must have the exact same properties, so a white horse, whose color is known to be white, cannot be a member of the *horse* class, where color is unknown.

If we accept this convention, monotonicity means that we can draw further conclusions by means of the elementary link-tracing logic apparatus we have discussed in Section 4.5, for example that the tail of a white horse is the tail of a horse; if we cut a tree with a chainsaw, we cut it with a power tool; etc. In negative contexts we get antitone behavior: if we cut a tree *without* a power tool, it follows we cut it without a chainsaw. We will say that those quantifiers that exhibit monotone behavior have *positive polarity*, and those that exhibit antitone behavior have *negative polarity*.

To complete the calculus, we need two further observations. First, that polarity is sensitive to which argument we are describing. Consider the scheme *every x is y*: clearly



if $x' \text{ IS_A } x$ the conclusion *every x' is y* follows, but if $y' \text{ IS_A } y$ the conclusion *every x is y'* does not follow.

Exercise[→] 7.21 Try to construct both plausible examples and counterexamples for the claims made above, using *some*, *no*, *many*, *most*, *three*, and *exactly three* as your quantifiers.

The other observation is that polarity can be affected by items appearing outside the construction: for example, when we say *it is not true that every x is y* ; the entire implicational machinery is turned around, see for example Szabolcsi (2004).

Peircean or ‘natural’ logic simplifies the apparatus greatly, but the logicians and philosophers developing these systems generally restrict themselves to the study of inferences that are *sound* in the technical sense of Section 2.6. Unfortunately, everyday logic is full of unsound inferences, such as the Rule of Proportional Size that we discussed in Section 3.8. How to incorporate such rules into a system capable of performing the kind of inferences required for solving problems of the kind described in Section 7.1 remains an active area of research. Some suggestions are offered in Chapter 8, where we turn to the logic (or, if you insist, illogic) of how people evaluate things.

Exercise[†] 7.22 Analyze the following syllogism;

- (i) Knowing the truth can only lead to good consequences,
- (ii) The theory of evolution makes people turn away from God.
- (iii) Turning away from God is bad.
- (iv) Therefore, the theory of evolution is false.

7.5 Further reading

For cogent criticism of the standard Turing test see Shieber (1994) and Levesque, Davis, and Morgenstein (2012). In addition to McCarthy’s classic discussion, a new set of texts, with critical annotation, has been produced as part of the [TACIT project](#). Besides the well-known problems of knowledge discovery and knowledge selection, the test set discussed in Section 7.1 also brings to light some subtle issues of linguistics/lexicography: an ordinary dictionary will not have a lexical entry for *zoom by* or *crush through*, but will have one for *shoot down*. Such entries, known as *phrasal verbs*, appear in many, if not most, languages, and are notoriously hard to define and collect; see Courtney (1983) and Vincze (2011). For naive space–time geometry see Hayes (1987), Hayes (1995), Talmy (1983), and Herskovits (1986).

The use of models that include models of other people’s thoughts puts the theory developed here in a minority position, well articulated, for example in Parsons (1974). The philosophical underpinnings of the treatment of defaults offered here can be traced to Aristotle and Locke; for a modern statement, see Fine (1985). For prototype theory, see Rosch (1975), Lakoff (1987), and Gärdenfors (2000). The standard introduction to modal logic, Hughes and Cresswell (1996), is very technical, and we have suppressed many details here. In particular, normal systems are not defined by simple adherence to



the Rule of Necessitation alone, but require other axioms; see Chapter 1 of Hughes and Cresswell (1984).

The view of strict determinism that requires free will to be an illusion is articulated with great conviction by one of the characters in [Beep](#), a classic science fiction story by [James Blish](#), later expanded to a full novel (Blish, 1973).

For a mature system that embodies almost all that has been accomplished in the nearly half-a-century search for an algorithm that can translate English into logic formulas, see Morrill (2011). Aristotle and the Schoolmen, in particular the works of [Petrus Hispanus](#), [William of Ockham](#), and [John Buridan](#), remain a treasure-house of insight and inspiration for logicians who wish to steer closer to natural language semantics than to foundational studies, and today there is a growing body of work reassessing this corpus in light of modern logic; see, for example, Fine (2012), Klima (2009), and Restall (2007). Their theories of syntax are also of great interest; for an introduction see Covington (1984).

Peirce is now hard to read in the original, because the terminology, which he invented copiously, did not take hold, but the central ideas of his logic are clearly reconstructed in Böttner (2001). The ideas of positive and negative polarity are now shared by MG and competing approaches: for the early work, see Jackendoff (1969) and Ladusaw (1980); for a more comprehensive bibliography see Beata Trawinski's [compilation](#) at the Tübingen "Idiosyncrasies of distribution" project.

The Stanford Encyclopedia of Philosophy offers detailed histories of philosophical thought about [non-existent objects](#) and the closely related [impossible worlds](#). For an illuminating discussion of the white horse debate, see Graham (1989) pp. 82ff. For a modern 'natural logic' calculus of quantifiers, see Manning and MacCartney (2009), and for the entire RTE shared task, see the [ACL Wiki](#).



Embodiment

Contents

8.1 Perception	222
8.2 Action	229
8.3 Adverbials	236
8.4 Further reading	238

Embodied cognition is the thesis that cognition is “deeply dependent upon features of the physical body of an agent, that [...] aspects of the agent’s body beyond the brain play a significant causal or physically constitutive role in cognitive processing” (SEP). As we have only a cursory understanding of the cognitive systems of dolphins and whales, the real test of this thesis will have to wait until we can investigate space aliens or, perhaps more realistically, artificial general intelligences (AGIs).



In this book, we steer clear of the very well-known, and extensively studied, difficulties of sensory and motor systems (pattern recognition and robotics), but we must consider their semantic aspects: what does it mean to perceive something and to act in an agentive fashion? That we cannot get around these questions, even if we take the narrowest possible focus and concentrate on the understanding of ordinary texts, is clear from John McCarthy’s original example that we discussed in 3.1: *Mr. Hug was pinned in the shaft for about half an hour until his cries attracted the attention of a porter*. For the story to make any sense, we simply must assume that there are conscious agents around, like our porter.

As we have seen in Chapter 7, perception is the primary means of adding information to our internal model, and action means modifying the external model. While the internal model is largely symbolic, built from discrete categories, in 3.3 we have already discussed that we perceive the external world as continuous, in regard to space and time. Importantly, pairs of adjectives like *nice–awful*, *big–little*, and *burning–freezing* (2.7) are also perceived together with a continuum running between the opposing poles, and sometimes beyond them. To handle these and similar continua, in 8.1 we introduce Euclidean automata, which take their input from a continuous parameter space P .

Action logic, the study of plans about future events, is an important aspect of AGI research. In 8.2 we approach this matter indirectly, not so much in terms of the actual plans (which we conceive of in discrete time, in terms of the primitives such as BEFORE and AFTER that we discussed in 6.4) as in terms of setting the direction of action: what it is that we consider a good (desirable) or bad (undesirable) outcome, and how we prioritize these, again a matter that requires a formal apparatus capable of handling continuous inputs.

Finally, in 8.3 we deploy this apparatus on a somewhat underresearched part of the lexicon, the study of adverbials.

8.1 Perception

We begin by introducing Euclidean automata (EA), a simple generalization of finite state automata, informally. EA operate not on symbols from a finite alphabet as usual, but rather on vectors from a parameter space P , typically \mathbb{R}^n . (For quantum applications, \mathbb{C}^n would also be of interest, but we concentrate on the real case.) The main motivation for EA comes from [categorical perception](#), classification problems involving a forced choice between a finite number (in the most important case, only two) of alternatives. Such problems are very common in linguistic pattern classification, for example in optical character recognition (OCR) or automated speech recognition (ASR).

Since we want classifications to be stable under a small perturbation of the inputs, ideally the set of points in P classified to a given value should be open, yet it is evident that we cannot partition \mathbb{R}^n or \mathbb{C}^n into finitely many disjoint open sets. Approximate solutions thus must give up non-overlapping, for example by permitting probabilistic or fuzzy outcomes, or exhaustiveness, for example by leaving ‘gray areas’ near decision boundaries where the system produces no output. EA, as we shall see, sacrifice non-overlapping but maintain sharp, deterministic decision boundaries.

Simply put, EA are obtained from standard finite state automata, as given by Definition 4.3 on page 88, by undoing the major abstraction concerning inputs. In FSA, inputs are simply selected from some finite alphabet Σ . In EA, *inputs* are given as parameter vectors from a parameter space P , typically \mathbb{R}^n , and *states* are subsets P_i of P indexed from a finite index set S . Experience with [general systems theory](#) shows that undoing the abstraction concerning outputs as well would lead to a theory that was too general to have any utility, and so we will refrain from doing so. We will define Euclidean versions of finite state transducers and Eilenberg machines that we will call Euclidean transducers (ETs) and Euclidean Eilenberg machines (EEMs), keeping the output alphabet of the transducer and the side-effects of machines both discrete and finite. But before turning to the formal definition, let us provide some informal, easy-to-grasp examples both to familiarize the reader with the terminology and to compare and contrast Euclidean automata with better-known models.

Example 8.1 *The elevator.* A three-stop elevator running from the basement to the top (first) floor will have three main input parameters; the reading from the current position

sensor, a real number between -1 and $+1$; the reading from the engine sensor, with possible values of ‘going up’, ‘stopped’, and ‘going down’; and the reading from the weight sensor, with any possible nonnegative reading, but in effect quantized to two discrete values, ‘above safety limit’ and ‘below safety limit’. By having a finite state space, even continuous parameters such as the height above ground are effectively quantized: whether the value is 0.3 or 0.9 makes no difference, no matter how the other parameters are set: we see the same transition function for both values. We will call two parameter vectors *indistinguishable* as long as this is true in regard to transitions for EA, both transitions and outputs for ETs, and both transitions and effects for EEMs. By relying on representatives from indistinguishable classes of parameter settings we can *skeletonize* EA and obtain classical FSA, but as we shall see, key aspects of EA behavior go beyond what the skeleta can do.

Example 8.2 *The GSM phone.* Near national borders, GSM handsets behave like EA: depending on which country the phone is in, it will send the user welcome messages describing the price of a call, etc. We can think of P as being composed of two parameters, longitude and latitude, or as being composed of several parameters representing the signal strengths from various cell towers. Either way, it is the values of these continuous parameters that determine (in addition to keyboard input) the behavior of the EA. Two aspects of this example are worth emphasizing: first, that the immediate behavior of the EA is determined by both the input and its previous state (so the natural formulation will resemble Mealy, rather than Moore, automata) and second, that the output of one EA can impact the input of other EA, for we may very well conceive of cell towers themselves as EA (though the changes in their inputs are effected by changes in electricity supply, call load, etc. rather than by changes in their physical location).

Example 8.3 *The heap.* The heap or [sorites](#) paradox, known since antiquity, probes the vagueness of concepts like ‘heap’ – clearly one grain is not a heap, and if k grains are not a heap $k + 1$ grains will also not be, so the conclusion that 10,000 grains are not a heap seems inevitable. Here we will take the following form of the paradox (Sainsbury and Williamson, [1995](#)):



Imagine a painted wall hundreds of yards or hundreds of miles long. The left-hand region is clearly painted red, but there is a subtle gradation of shades, and the right-hand region is clearly yellow. The strip is covered by a small double window which exposes only a small section of the wall at any time. It is moved progressively rightwards, in such a way that at each move after the initial position the left-hand segment of the window exposes just the area that was in the previous position exposed by the right-hand segment. The window is so small relative to the strip that in no position can you tell the difference in colour between what the two segments expose. After each move, you are asked to say whether what you see in the right-hand segment of the window is red. You must certainly answer “Yes” at first. At each subsequent move you can tell no difference between a region you have already called red and the one for which the new

question arises. It seems that you must after every move call the new region red, and thus, absurdly, find yourself calling a clearly yellow region red.

We will model this situation by an EA with four skeletal states numbered 0 to 3 (Fig 8.1), and a single numerical parameter corresponding to the wavelength λ at the spectral peak and running from 720 (red, left end of wall) to 570 (yellow, right end of wall). The arcs are 01, 13, 32, 20 and the self-loops 00, 11, 22, 33. Outputs chosen from a two-letter alphabet $\{r, y\}$ are emitted on arcs (Mealy machine) rather than in states (Moore machine) according to the following rule: the 00, 01, 20, and 11 arcs emit r , and the 33, 32, 13, and 22 arcs emit y . Euclidean-ity is expressed by dividing the input range into three non-overlapping intervals: the machine receives input in the range [720–620] it settles in state 0, if the input is in the range [570–590] it goes to state 3, and in the ‘orange’ range (590–620) it will stay in state 1 if it was previously in state 1, and in state 2 if it was previously in state 2.

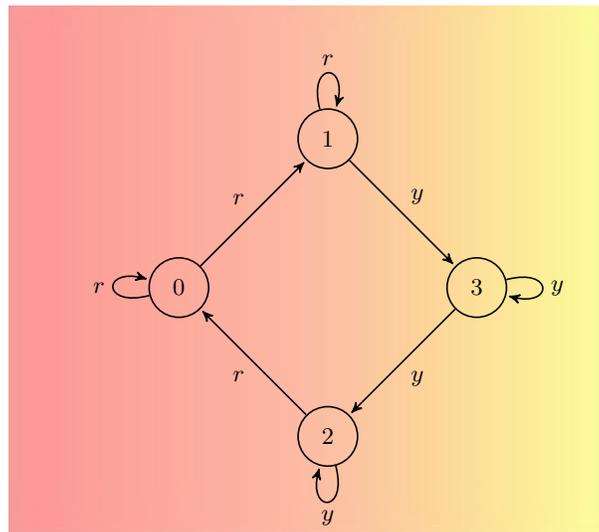


Fig. 8.1. Color perception EA with hysteresis

If we provide inputs to this EA with slowly decreasing wavelengths λ running from 720 to 570 nanometers, the EA will move from state 0 to state 1 at $\lambda = 620$, and from there to state 3 at $\lambda = 590$. The output switches from r to y when the 13 arc is first used, at $\lambda = 590$. When we perform the opposite experiment, increasing the wavelength from 570 to 720 in small increments, the EA will switch from y to r as it passes from 2 to 0 at $\lambda = 620$. In the entire orange region, the model shows **hysteresis**: if it came from the red side it will output red, if it came from the yellow side it will output yellow.

Note that in the EA account the sorites paradox is not an edge phenomenon, restricted to some critical point when the non-heap becomes a heap and red becomes



yellow (Sainsbury, 1992), but something that characterizes a substantive range of parameters with nonzero measure. In fact, the hysteresis seen in the example is consistent with perception studies on single-parameter spaces (Hock et al., 2005; Poltoratski and Tong, 2013; Schöne and Lechner-Steinleitner, 1978). As we will see in Section 8.2, it is precisely the existence of such overlapping regions that makes it possible to model conflicted inner states by EA.

Definition 8.1 A *Euclidean automaton* over a parameter space P is defined as a 4-tuple (\mathcal{P}, I, F, T) , where $\mathcal{P} \subset 2^P$ is a finite set of states given as subsets of P , $I \subset \mathcal{P}$ is the set of initial states, $F \subset \mathcal{P}$ is the set of accepting states, and $T : P \times \mathcal{P} \rightarrow \mathcal{P}$ is the transition function that assigns for each parameter setting $\mathbf{v} \in P$ and each state $s \in \mathcal{P}$ a next state $t = T(\mathbf{v}, s)$ that satisfies $\mathbf{v} \in t$. If $P_i \cap P_j = \emptyset$ for all $i, j \in S$, we call the EA *deterministic*; if $\bigcup_{i \in S} P_i = P$, we call it *complete*.

Exercise^o 8.1 Prove that there is no deterministic classification of a connected parameter space $P \subset \mathbb{R}^n$ by an open EA.

Definition 8.2 A *Euclidean transducer* over a parameter space P is defined as a 5-tuple $(\mathcal{P}, I, F, T, E)$, where \mathcal{P}, I, F , and T are as in Definition 8.1 and E is an emission function that assigns a string (possibly empty) over a finite alphabet Σ to each transition defined by T .

Definition 8.3 A *Euclidean Eilenberg machine* over a parameter space P is defined as a 5-tuple $(\mathcal{P}, I, F, T, R)$, where \mathcal{P}, I, F , and T are as in Definition 8.1 and R is a mapping $P \times \mathcal{P} \rightarrow P$ which assigns to each transition a (not necessarily linear, or even deterministic) transformation of the parameter space.

We have already seen examples of EA. A particularly relevant example of an ET is a vector quantizer (Gersho and Gray, 1992), and if $P = \mathbb{R}$, an *A/D converter*. Since Eilenberg machines (see Definition 4.4 on page 90) are less well known, we discuss the simplest cases individually. For $|\mathcal{P}| = 1$ we have a single mapping $P \rightarrow P$, and for $|\mathcal{P}| = k$ we have a finite family of $P \rightarrow P$ mappings. As the sets P_i collected together in \mathcal{P} may be overlapping, there is no guarantee that the mappings together describe a function (as opposed to a relation) over P , and even in the locally deterministic case EEMs are capable of realizing multivalued functions. Another example is the following.

Example 8.4 The Artificial Neuron The elementary building blocks of artificial neural networks (ANNs), both with sigmoid squishing and without, can be conceived of as two-state EEMs. The parameter space has d dimensions, where d counts the number of inputs (dendrites), and the operation of the EEM is deterministic: if the sum of the inputs is smaller than the threshold (after squishing in a sigmoid ANN, or without squishing in a linear ANN), the unit goes into state 0, otherwise it goes into state 1. The output function is constant 0 in state 0, and 1 in state 1.

Notice that artificial neurons can also be conceptualized as ETs, with output alphabet $\Sigma = \{0, 1\}$ and inputs taken from Σ^d – this is because in standard artificial neurons the outputs do not depend on the details of the input vector, just on the state it transitions to. In general, where there is no need to distinguish the subtypes, or the subtype is evident



the from context, we will speak of *Euclidean Machines* (EMs) as a cover term for EA, ETs, and EEMs.

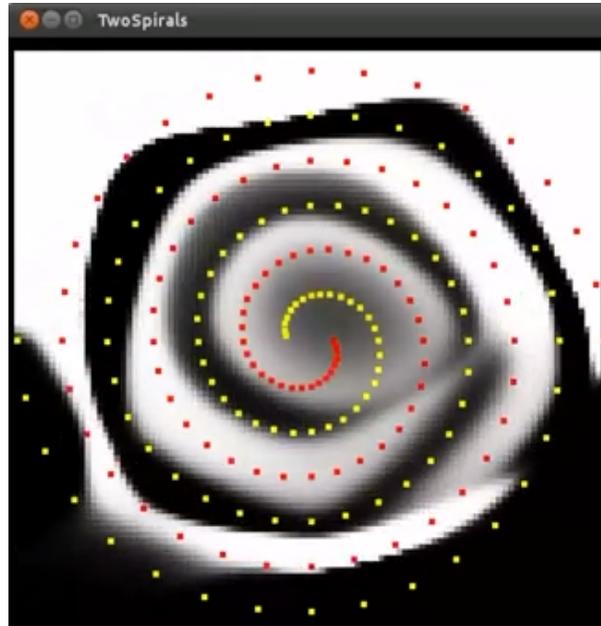


Fig. 8.2. Decision boundary in 2–20–10–1-layer perceptron

As an example, consider the multilayer perceptron depicted in Fig. 8.2 trained by Fabisch (2011). In spite of the complexity of the decision boundary, the EA with equivalent behavior has only two states, corresponding to the black and the white subsets of the image. The input vectors are two-dimensional, and there is no output to speak of (we could designate one of the two states as final).

The definition of EA leaves open the possibility that the parameter space P is embedded in \mathbb{R}^n in a partially discrete manner, for example as indexed subsets of lower-dimensional spaces. Returning to Example 8.1 (the elevator), the EA will have both continuous parameters, such as the reading from the position sensor, and discrete parameters, such as the reading from the engine sensor, with only three possible values ‘going up’, ‘stopped’, and ‘going down’. Some of the parameters, such as the reading from the weight sensor, are seemingly continuous but effectively quantized to two discrete values, ‘above safety limit’ or ‘below safety limit’. In such cases, we may want to select *canonical representatives* \mathbf{p}_i from P_i . (Other input values, pertaining to the state of the call buttons at each floor and inside the cab, to accelerometer readings, or to sensors for AC power quality, could be added, but we don’t aim at realistic detail here.)

As long as the parameters can be isolated from one another, we can view P as being a direct product of smaller parameter spaces P_i , some Euclidean, some discrete. Isolating

the parameters is easy enough for elevators with different sensors, but not at all trivial in pattern recognition tasks where the individual coordinates, such as spectral peaks in ASR, can show all kinds of interdependence. There is notable uncertainty about how we wish to embed the discrete spaces in \mathbb{R} ; for example, two-valued parameters are often encoded as 0 or 1, but often as -1 or $+1$, and there is no easy way to select a canonical embedding. In many applications, n -valued parameters are encoded as $0, \dots, n-1$, in others as $1, \dots, n$, and in yet others as $0, 1/(n-1), 2/(n-1), \dots, 1$. Let us first consider a family M of $P \rightarrow P$ mappings with the goal of replacing one conventional encoding by another. As the examples show, such mappings are typically taken from continuous/differentiable families, but are not necessarily linear.

Definition 8.4 An EA Ψ is the *homomorphic image* of Φ under a mapping $m \in M$ iff for any sequence of inputs $\mathbf{v}_1, \dots, \mathbf{v}_n$ we have $\Psi(m(\mathbf{v}_1), \dots, m(\mathbf{v}_n)) = m(\Phi(\mathbf{v}_1, \dots, \mathbf{v}_n))$.

Here we assume that both Φ and Ψ are started from the same unique initial state (if we permit several initial states the definition needs to be complicated accordingly) and that equality means equality of result state. This is meaningful, since m naturally maps not just inputs on inputs, but also EA states (subsets of parameter vectors) on one another. We will say Φ and Ψ are *isomorphic* if they are homomorphic images of each other under some m and m^{-1} .

Definition 8.5 The *skeleton* of an EA Φ is a standard (Mealy) FSA whose alphabet corresponds to canonical representatives from each Boolean atom of \mathcal{P} .

In the deterministic case, this is also a Moore automaton, as there is a one-to-one correspondence between input letters and automaton states. As is clear from Definition 8.1, the sequential behavior of EA is relatively simple in this case, since the result state depends only on the input, and not on the previous state. In the nondeterministic case, we may not be able to select distinct canonical representatives for each state P_i , or even for the set of Boolean atoms formed by the P_i .

Exercise[†] 8.2 Generalize skeleta to the nondeterministic case. Can you maintain uniqueness up to canonical isomorphism? Can you maintain the one to one correspondence between the state set and input set?

For many applications it makes sense to define the initial state as a parameter region P_0 that has no overlap with the other states of the automaton (even for EA that are not otherwise deterministic), since this will guarantee that we can *reset* the EA to the initial state by making sure that there are outbound transitions from every P_i . If we have another region we can reset to, we obtain an EA corresponding to a classical [flip-flop](#) or latch circuit. We can also obtain classical circuits with hysteresis, such as a [Schmitt trigger](#) (Schmitt, 1938).

Exercise[→] 8.3 Use EA/ETs/EEMs to describe elementary building blocks of electronic circuits such as MOSFETs.

All forms of logic circuitry operating on continuous variables such as voltages could be recast as networks of EEMs. As is standard in logic design, digital circuitry can be conceptualized as the series-parallel composition of standard FSTs, either with the



output string length limited to 1 (otherwise issues of timing and synchrony become paramount) or with clock signals added in. For semi-analog circuitry, where the outputs of each building block can be characterized as constant values (or values with very little variation), the same series–parallel conceptualization is available with ETs, as long as we take the outputs of the upstream ETs to be the canonical representatives of the inputs of the downstream ETs. This means that in principle all physical models of digital computation, realized by discrete electronics as they are in current computers, are within the scope of the EA/ET/EEM model given by Definitions 8.1–8.3.

Besides the well-understood serial and parallel modes of composition discussed above, EA admit a further possibility. This can be illustrated even in the simplest case of a mixed parameter space, where P_c , the continuous part of the parameter space, is just \mathbb{R} , and P_d , the discrete part, is just a binary choice \top, \perp . We may think of an EA Φ over $P = P_c \times P_d$ as being composed of two simpler EA, Φ_\top and Φ_\perp , by means of a real parameter p that *influences* whether the Φ_\top or the Φ_\perp behavior dominates. Importantly, the parameter that does the influencing may be just the input parameter, providing a crude form of memory, as in Example 8.3 (the heap).

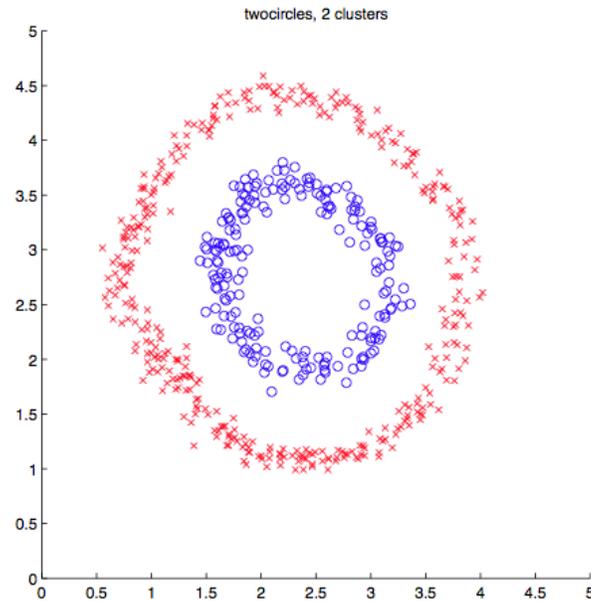


Fig. 8.3. “Two circles” data from Ng (2001)

Perception, memory, and action are closely intertwined, and in this section our main goal was to link EA to perception, classification tasks in particular. When we model perceptual classification by EA, our interest is in the inverse images C_i of the possible outputs i . As our example in Fig. 8.1 shows, EA offer a method for directly encoding

the information concerning the shape of the C_i where it belongs, in \mathbb{R}^n , where n is the dimension of the input parameter vector, rather than in $\mathbb{R}^{m \times m}$, the matrix of connection strengths. As is well known, in pattern recognition a great deal depends on the preprocessing of the signal, and using EA can make this dependence explicit. For example, consider the “two circles” data set presented in Ng, Jordan, and Weiss (2001), reproduced here as Fig. 8.3. While it is evident that no linear separator (simple NN) exists, transforming the data to a system of polar coordinates around the center of gravity of the data points would make the task trivial. In ASR, we routinely apply a far more elaborate sequence of data transformation steps (power cepstra (Bogert, Healy, and Tukey, 1963), mel warping (Davis and Mermelstein, 1980), and delta cepstra (Furui, 1986)) to make the data manageable. Altogether, the use of EA is expected to bring new insights, especially for the increasingly popular but not yet well understood “deep learning” neural net architectures such as the LSTM (Hochreiter and Schmidhuber, 1997).

8.2 Action

In Section 8.1 we introduced EMs with the idea of providing a simple formal account of perception capable of dealing not just with continuous input, but also with the hysteresis effects observed during human perception. EMs, just like standard FSA, are also capable of providing an account of (finite) memory. But their greatest value lies in the fact that they enable robust anthropocentric use of moral vocabulary. We hold, with (McCarthy, 1979 (1990)), that

to ascribe certain *beliefs, knowledge, free will, intentions, consciousness, abilities, or wants* to a machine or computer program is legitimate when such an ascription expresses the same information about the machine that it expresses about a person. It is useful when the ascription helps us understand the structure of the machine, its past or future behavior, or how to repair or improve it.

Indeed, much of semantics is “to do what [Ryle, 1949] says cannot be done and shouldn’t be attempted – namely, to define mental qualities in terms of states of a machine”. To the extent semantics aims at explaining what words mean (and as we have seen in Section 1.3, this is about 85% of the task), we cannot simply declare words pertaining to mental states off limits. Human *actions* involve *decisions*; in fact, without an element of contemplative decision-making and unforced choice we are better off talking about *automatisms*, a worthy subject in its own right, but not a matter of semantics. It would not occur to us to try to understand electromagnetic forces and phenomena by linguistic methods, studying the words *charge, electron, and attraction*. If anything, we follow the opposite route, taking the Maxwell equations as the basis of our understanding, and even endowing ordinary words with a strict technical meaning to the extent necessary, and in the process removing much of their original meaning.

Exercise^o 8.4 Build dictionary definitions of the non-technical senses of *charge, electron, and attraction*.

Since much, perhaps too much, of our decision process is driven by our hopes and fears, some formal mechanism to deal with these is necessary for any attempt at understanding action-related vocabulary. As a moment of introspection will show, we spend a great deal of our pre-action time being in a conflicted state. In the EM model, this comes out, unsurprisingly, not as a single state of the machine, but rather as a set of nondeterministic states *tied together by their shared territory of input parameters*. This framework smoothly extends from physical to moral conflicts. To see how this works, recast Example 8.3, the paradox of the painted wall, in terms of moral precepts. What we see is a conflict emerging between two, in themselves very reasonable maxims:

Factuality I ought to report things as I see them

Consistency I ought not to report differences where I don't see any

Importantly, the conflict arises even though we see the first precept as superior to the second one. Consistency is at best a refinement of Factuality, and we have a large number of warnings attached to it, from *Si duo faciunt idem, non est idem* to Emerson's famous quip "A foolish consistency is the hobgoblin of little minds". Eventually, if λ is made small enough, we sacrifice Consistency and say "No" because we cannot live with a strong violation of Factuality.



Fig. 8.4. Never Give Up

Let us now turn to a more direct example of conflict, the kind familiar from the 'Never Give Up' cartoon (Fig 8.4). We need two EA to model the situation, Frog and Stork, which we can assume to be isomorphic. At time t , each can be represented by two parameters, $p(t)$ corresponding to the power reserve it has, and $q(t)$ corresponding to the pressure it exerts on the other. We are less interested in the death spiral Frog and Stork can find themselves in, than in paths to disengagement, if there are any. We assume that for each party its $p(t + 1)$ depends on the other's $q(t)$ deterministically, and that each party can set its $q(t + 1)$ nondeterministically between 0 (standing down) and its own $p(t)$ (maximum effort to kill the other). If we take the abscissa as p and the ordinate as q_f , the skeleton can be depicted as shown in Fig. 8.5.

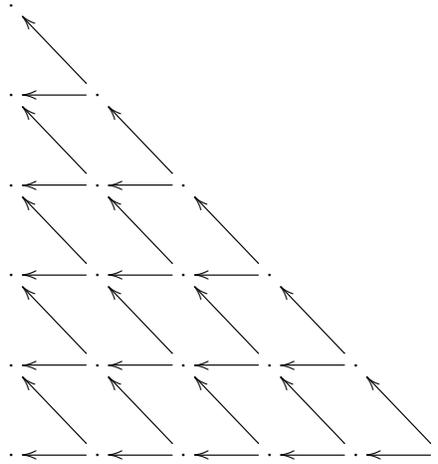


Fig. 8.5. Stork in (p, q_f) space

At every point, Stork has an option of applying as little force as it wishes, but no more than its power reserve. This choice is free in the sense of moral philosophy; it is only at the edges of the diagram that we see compulsion (deterministic behavior). The nondeterministic choices provide room for a broad variety of strategies, ranging from escalation through tit for tat to turning the other cheek. If we couple an escalating Frog to an escalating Stork we obtain the death spiral discussed above, and, importantly, a tit-for-tat player will also die if the other party relentlessly ratchets up the pressure – the only recourse of the non-aggressor is to take the aggressor with them to the grave.

EA are rather limited computational devices, yet they have enough power to serve as homunculi in our model of *internal* decision-making. In what follows, we think of EA as receiving input in discrete time, but this is not essential for reaching, and maintaining, conflicted states. We will study asynchronous networks composed of EA, with particular attention on serially connected EA A_1, A_2, \dots, A_k where each A_{i+1} receives as its input the output of A_i , possibly cyclically. ‘Never Give Up’ arises at $k = 2$. We will not pursue a full game-theoretic analysis here, as our chief concern is not with the possible outcomes at the individual or population level, but rather with formalizing the moral calculus that can operate within the domain of free will. For this the simple Stork model is insufficient, as it lacks the critical variables corresponding to the hopes and fears of the player. The key idea is that such hopes and fears are simply internal models of the diagram edges, but before we turn to this, let us take a closer look at the next simplest case, $k = 3$, known in popular culture as the ‘Mexican standoff’. Needless to say, instead of gunmen keeping each other in check, our interest is in internal conflicts between different drives or values.

In the standard framework of Artificial (General) Intelligence, the decision process is modeled by a value or *utility* function U : given several possible outcomes o_1, o_2, \dots, o_k the agent simply computes $U(o_1), U(o_2), \dots, U(o_k)$ and chooses the best. Some diffi-

culty may arise when values come out equal, but this is seen as a marginal phenomenon, since functions of very many variables (clearly needed to describe the many possible outcomes) rarely take the exact same value at different points. The emphasis is epistemic: how does the agent acquire the information it needs to compute the utility?

The pioneers of cybernetics were already aware of the *circularity-of-value* anomaly, exhibited, for example, by rats starved both of sex and of food: they prefer sex to exploration, exploration to food, and food to sex. If we model different drives by different agents, circularity-of-value anomalies boil down to [Condorcet's paradox](#), but one does not need to subscribe to a [society of mind](#) assumption to see the point made by McCulloch (1945) that such circular preferences are “sufficient basis for categorical denial of the subsumption that values were magnitudes of any kind”. Circularity of value is seen in many settings besides economics (McCulloch mentions neurophysics, what he calls ‘conditioned reflexology’, and experimental esthetics), and these cases demonstrate rather clearly that utility-based models are too simplistic for describing the behavior of rats, let alone those of humans or AGIs.

McCulloch's original model of the phenomenon does not lend itself to easy reproduction in terms of our contemporary understanding of networks, which no longer conceptualizes behavior in terms of reflex loops. The Euclidean machines advanced here have the advantage that their main features can be analyzed without reference to recurrent behavior or nuances of timing. For $k = 2$ and 3 only cyclic conflict models are available, but for $k \geq 4$ we can obtain a broader variety by optionally adding chords to the main cycle. Taking into account which parameters in the input of A_i are output by A_j we obtain a rich typology of conflict. We begin with the simplest case, the four-state machine depicted in Fig. 8.1, which represents conflicted behavior in a forced binary choice.

To see how this conflict is created, consider two homunculi, A_f in charge of factuality and A_c in charge of consistency, with A_c the weaker of the two, so that in a game of Never Give Up A_f will eventually win. Without consistency, A_f by itself is not particularly conflicted: it will opt for red when the input wavelength is sufficiently large, say at $\lambda > 620$, and for yellow when $\lambda < 590$. The simplest approach is to represent this by a linear function $y = (\lambda - 605)/15$, which is -1 or less in the unambiguously yellow range, and $+1$ or more in the unambiguously red range. Many alternate functions could be considered (radial basis neural nets are a very attractive possibility), but we would like to see the qualitative emergence of conflicted states without fine-tuning the network response. Skeletonizing A_f leads to a simple two-state automaton, outputting r in the range from 605 to 720, and y in the range from 570 to 605. The behavior at the boundary of the attractor basins (which we take to be 605) is irrelevant not just because this is a zero-measure set, but because this behavior is completely overshadowed by hysteresis.

Sainsbury and Williamson set up the protocol taking particular care that A_c , the guardian of consistency, was always aroused: “the window is so small relative to the strip that in no position can you tell the difference in colour between what the two segments expose”. At the beginning (left side, $\lambda = 720$), A_c is inactive, and A_f simply



outputs r . As λ is decreased, say in 1 nanometer decrements, though the exact number is irrelevant, A_C will become active, and will always pull the decision toward the last decision, whatever it was, with force $c < f$. Skeletonizing A_C is a much more interesting issue, since in general we would need to endow this automaton with two memory registers, one to store the last output whose consistency is to be maintained, and one to store the last input to see if we are close enough that consistency is required to begin with. For an increment of 1 nm and two outputs, this would require $2 \cdot 151$ states, which is unattractive both because this number is too large and because it is inversely proportional to the stepsize, a small but arbitrary parameter unlikely to be critical for our understanding of the problem. A more attractive solution is to conceptualize A_C as an EEM, with only three states, ‘neutral’, ‘sticking to red’, and ‘sticking to yellow’, and with three transformations of the inputs. The identity function is attached to the neutral state when two subsequent inputs are too far apart for consistency to make sense, a ‘red boost’ function of $+15$ is attached to the ‘sticking to red’ state, and a ‘yellow boost’ function of -15 is attached to the ‘sticking to yellow’ state.

The central distinction from simpler additive models is that A_C is seen as manipulating the *input* of the main binary classifier A_f , rather than contributing to, or even reversing, its output. Once this is understood, we can further simplify A_C by removing its memory (third state) and assuming that it just adds back the output of A_f to its input when the unbiased input is seen as close to what it was before, see Fig. 8.6. For doing this, we need to address another property that the standard treatment of networks generally abstracts away from, *seminumericity*. A_f , as we have defined it so far, takes numeric input (wavelengths measured in nanometers) from 570 to 720, and produces symbolic outputs r and y . One approach would be to freely rescale the numerical values to between 0 and 1 (activation level), or between -1 and $+1$ (including inhibitory effects). Textbook treatments of neural networks generally opt for this solution, without much discussion of the costs attendant on rescaling, and simply pave over the difficulties of replacing categorical variables like *red/yellow* by pseudo-numerical values such as the ± 1 we used above. Historically, the subtle interplay between the deductive and the numerical approach is well understood from the numerical side: the entire second volume of Knuth (1969) is devoted to this issue. What is called for here is the converse, a better understanding of the semi-symbolic nature of biological computation.

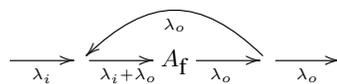


Fig. 8.6. A_C as a feedback loop modifying A_f

The approach proposed here, inspired by semantic ideas from cognitive science (Rosch, 1975), is to recast the symbolic output of EMs as numeric, for example to assume that the output of the classifier will be the prototypical red, say 630, or the proto-

typical yellow, say 580. In the case with hysteresis, as we decrease the input wavelength from 720 to below the boundary point at 605, say to 600, A_f would now report yellow, but because the previous reports were all red, the input it sees is not 600 but 630 since the previous output was mixed in by A_c . In fact, the raw input has to go below 580 for the mixture to get below 605, and in the intermediate range we observe hysteresis. Conversely, if we start from low wavelengths, we need to get above 630 to get away from the yellow and have the system switch to red. By adjusting the mixture weights, it is possible to increase or decrease the range of hysteresis, in the extreme case to a point that a machine once committed to an answer will never depart from it. This is obviously maladaptive in a system for perception, but would make perfect adaptive sense in a unit dedicated to memory.



Exercise → 8.5 Using the fact that EMs can model standard (sigmoid) ANNs, generalize the standard [backpropagation](#) training algorithm to EMs.

So far, we have offered only the beginnings of an analysis of being in a conflicted state, some situation where we know that we should do A , since it is ‘the right thing to do’, yet we have a strong compulsion to do some B (including doing nothing) instead. To broaden the discussion we will use several specific examples, such as refraining from or taking some drug, such as tobacco, alcohol, or heroin, that is generally agreed to have pleasant short-term but harmful long-term effects; slipping into some recreational activity while there is still work to do; keeping or not keeping some promise; etc.

The problem is complex; arguably, it is the single most complex problem we have to face in everyday life. Therefore, some simplification will be necessary, and we will state the main problem in a way that already abstracts away from certain aspects that would take us far from our goal of analyzing internal conflict. First, we are not interested in defending the specific moral premisses used in the analysis of drug addiction, laziness, and similar examples of conflict: our focus is on the conflicted state itself, not the individual components. Second, we will pay only limited attention to the issue of how we know that A is the right thing rather than some alternative A' or even B – we are interested in the situation when we already *know* that A is right and B is wrong. Third, real-life conflicts are rarely between two laboratory-pure components: often there are multiple factors, but the binary case must be addressed first. Finally, conflicts are often graded (perhaps a small glass of wine is quite OK where a bottle would not be), but here we will try to work with as simple and minimalistic a setup as we can.

The mainstream assumption, embodied in AGI architectures like OpenCog (Hart and Goertzel, 2008), is that there is some *utility function* that the agent intends to maximize. If this function changes at all, it changes only adiabatically, on the order of weeks or months, while the decision to do the right thing often has to be taken on a subsecond scale. Certain issues can therefore be stated in terms of a single utility function that is discounted on different scales. Let $u(t)$ measure the sensation of somatic well-being on a scale of -1 (suffering) to $+1$ (exultation) at time t . If our interest is in maximizing $\int_0^T u(t)e^{-Ct}dt$, choosing a large C leads to behavior that focuses on the momentary exhilaration, while choosing a small C models maximizing long-term well-being. This

is a nice and simple picture: if a behavioral alternative, say smoking a cigarette, has some known effect expressible as a transform A , while B has an effect B , we simply compute $\int_0^T A[u(t)]e^{-Ct}dt$ and compare it with $\int_0^T B[u(t)]e^{-Ct}dt$.

Such an analysis, however, would suggest that conflict is restricted to a few marginal cases, where our best estimate of A and B carry large uncertainties, and is therefore largely an epistemological issue: as soon as we have better estimates the conflict disappears. This is a known philosophical position going back to antiquity: to quote Graham (1989) (p. 190) “Unlike Mohists and Yangists seeking grounds for right choice Chuang-Tzu’s ideal is to have no choice at all, because reflecting the situation with perfect clarity you can respond only in one way”. Clear as this position may be, is not at all helpful for predicting behavior: in reality, people spend a lot more time in conflicted states than this analysis would suggest. Even more damning, it ignores the central case, where the impacts of A and B are perfectly known. Rare is the addict who doesn’t know she should quit, or the promise breaker who doesn’t know better – the problem is not lack of knowledge, but failure to act on it.

A somewhat richer model presumes not just one utility function but several: u_1 for somatic well-being, u_2 for reproductive success, u_3 for danger avoidance, and so forth. In such a view, conflicts between A and B are simply cases when some u_i would lead to one choice but another u_j would lead to the other. Since there can be large domains where the different u s lead to different choices even if they are selected from otherwise well-behaved classes of functions (for example, piecewise linear or low-order polynomial), this model escapes the first criticism discussed above, but not necessarily the second, a matter we will discuss shortly. Such a model fits well into multi-agent theories of the mind (Minsky, 1986), by assigning each agent A_i a dedicated utility function u_i .

We will frame the problem in terms of multiple (competing) utility functions, each with its own little homunculus intent on maximizing it, but first we have to discuss two significant reduction strategies. The first one would replace the u_i with their weighted sum $\sum_i w_i u_i$ using static or very slowly changing weights. This makes a lot of sense when choices are evaluated in terms of some resource that behaves additively, such as memory or CPU expenditure, as long as there is only one of these which is truly scarce. But as soon as the system is dealing with several resource dimensions (for example, CPU time, RAM, and disk space can all be limiting) we are back to the multiple-optimization scenario, except it is now the resource tallies r_j that are to be minimized subject to (slowly changing) tradeoffs between them. For the problem at hand, moral correctness must be considered a separate resource on its own, since it is well understood that most problems have simple solutions as long as the moral constraints are ignored.

The second reduction strategy is based on a hard-line interpretation of a single utility, say u_1 (somatic well-being). Competing utilities, such as u_3 (danger avoidance), are considered epiphenomenal: big danger just means a high probability of complete zeroing out of u_1 , and a strategy aimed at maximizing the area under the u_1 curve will result in some degree of u_3 maximization just because of this. Similarly, in a ‘selfish gene’ calculus, the intent is to maximize the area under the u_1 curves for all progeny; thus

low reproductive success is penalized without ascribing any specific utility u_2 to high reproductive success. Note that this strategy does not guarantee a hierarchy among the u_i , because reducing u_j to u_i does not guarantee that a reduction in the other direction is infeasible. For example, taking as primary the (future-discounted) somatic well-being of progeny will make direct somatic well-being of the individual an important factor even if it receives zero direct weight in the sum, since an individual deprived of well-being is very unlikely to make the effort to reproduce successfully.

8.3 Adverbials

A key linguistic area opened up by the EA framework is the study of adverbials, for example that Frog can *relentlessly* ratchet up pressure on Stork, or that reporting ‘red’ or ‘yellow’ is indeed a case of conflict between two virtues, being factual and being consistent, and so forth. Here we continue with the distinction made in Section 7.2 between external and internal models, and argue that the behavior of adjectives, which typically grade external objects on a scale, parallels that of adverbials, which grade mental states on a scale.

Such a parallelism explains many things that the more standard view, that adjectives refer to properties of objects and adverbials refer to properties of actions, has a hard time explaining. Here we return to the standard truth-conditional account we discussed in Section 3.7, as this makes it easy to highlight the central problem. Assume there are objects, typically denoted by nouns or noun phrases, collected in class n (entity). If adjectives refer to classes of objects, so that *red* refers to those objects that enjoy the property of redness, they must be functions $n \rightarrow t$, that is, functions from entities to truth values. If they grade objects on a scale, such as *wild-tame*, they must be functions from n to \mathbb{R} .

Now if there is a separate class v (of events), usually denoted by verbs, or verb phrases, adverbs must be functions $v \rightarrow t$ or $v \rightarrow \mathbb{R}$. This works very well with pairs such as that provided by *swift river* v. *swiftly running river*. It is something of a mystery how the type signature of *ly* gets to be so complicated, but we accept the clear grammatical need to convert from adjective to adverbial, since everything is as it should be, the adjective *swift* attaches to a noun, *river*, and the adverb *swiftly* attaches to a verb, *running*.

Exercise \rightarrow **8.6** Using a type system of your choice, express the signature of a function that takes $n \rightarrow \mathbb{R}$ functions as input and converts them to $v \rightarrow \mathbb{R}$ functions.

There is a bit of syntactic leakage around the edges (most people will accept *the river runs swift*, not just *the river runs swiftly*), but altogether the system seems to operate reasonably well.

Exercise \rightarrow **8.7** Using a type system with disjunctive/dependent types, express the signature of a function that takes $n \rightarrow \mathbb{R}|t$ functions as input and converts them to $v \rightarrow \mathbb{R}|t$ functions.

The trouble begins when we notice that adverbials often operate on adjectives, not on verbs: consider the *supposedly charming actress*. One may for a moment consider *charming* a verb by analogy to running, but of course the construction doesn't yield to verbal transformations *the river swiftly runs*, **the actress supposedly charms*, and it can take pure adjectives that have no verbal counterparts, as in the *supposedly ugly actress*. Now if adverbials can take not just verbs but also adjectives as their argument, things are getting out of hand.

Exercise[→] 8.8 Using a type system with disjunctive/dependent types, express the signature of a function that either takes $n \rightarrow \mathbb{R}|t$ functions as input and converts them to $v \rightarrow \mathbb{R}|t$ functions or takes functions disjunctively typed as $v \rightarrow \mathbb{R}|t$ and converts them to $n \rightarrow \mathbb{R}|t$ functions.

All of this is well and good until you notice that adjectives such as *very*, which turn an adjective into another adjective, can and do function as adverbials as well, turning one adverbial into another one. It is a highly nontrivial task (and not left as an exercise to the reader, as a decent solution may not even exist) to assign a single type signature to adjectives without coercing v and n to be the same.

In reality, the type-theoretical issue is but a highly technical manifestation of what seems a far more elementary problem, that adverbs and adjectives are not really that different. The actress can charm, or she can fail to charm, but it is not in her power to supposedly charm – the adverb is not modifying an act of charming. What *supposedly* means is simply that the speaker is warning of there being a public perception of the actress being charming, and by the Gricean mechanism we discussed in Section 5.6 we assume that the speaker provides this warning precisely because he doesn't share this perception.

Exercise[◦] 8.9 Analyze the expression *supposedly former infatuation addict*.

Before turning to the analysis of adverbials as adjectives pertaining to mental state, let us summarize the difficulty with the 'adjectives pertaining to events' view we are criticizing here. As we discussed in Section 6.3, the evidence is quite clear that in language after language nouns and verbs are major lexical categories that strongly correspond to entities and events, respectively. There is no comparably strong cross-linguistic evidence to distinguish nominal and verbal modifiers, but there seems to be evidence that their modifiers, adjectives and adverbials, are indistinguishable. The further away we go from the central event/entity distinction, the more the type system loses its grip.

Under the theory of adverbials proposed here, this is not at all surprising: in our view, the distinction is not between *what* gets modified but in *where* the modification or update is directed, toward the external or the internal model. In this view, *hopefully Barça will win* is not an objective statement about the club; it is an objective statement about the speaker considering this a positive outcome.

A considerable number of adverbials share this property of clearly pointing to the inner model, be it that of the speaker, the hearer, or the one assumed by the community. For the first case, take *surprisingly*: if some statement p is false, *surprisingly* p is also

false, but this of course doesn't make *unsurprisingly p* true. Any event can be surprising as long as it is reasonable for someone to have different expectations. A similar treatment is available for *obviously*. *Surprisingly* means 'unexpected given the rest of the inner model', and *obviously* or *plainly* means 'expected given the rest of the inner model'.



A great deal of the pragmatic use and abuse of adverbials depends on the presumption that there is a common model. There is in fact a [consensus theory of truth](#), already clearly expressed in Heraclitus, Hesiod, and Aristotle, that whatever people agree on, *consensus gentium*, must be true, and as long as people share a reasonably uniform naive theory this can be leveraged to use adverbials like *naturally* or *finally*.

That adverbials are keyed to marking certain regions of the inner model is particularly clear from cases like *possibly* or *luckily* which really have nothing to do with probabilistic reasoning. Consider an expression like *the possibly fatal effects of the chemical*. There is no 'possible fatality' as a property of the chemical, it is just that its effects can be fatal.

Exercise[†] 8.10 Consider the adverbials *especially*, *almost*, *entirely*, *nearly*, *naturally*, *willingly*, *incognito*, *halfway*, *finally*, *alone*, and *in truth*. Which refer to the inner model of the speaker or the hearer, or to the consensus model?

8.4 Further reading



For an excellent introduction to the history and goals of artificial general intelligence, watch [Ben Goertzel's lecture](#). For more details on linguistic pattern recognition, see Chapter 8 of Kornai (2008). General systems theory, together with cybernetics, goes back to the [Macy conferences](#); see Heims (1991).



The EA/ET/EEM models were introduced in Kornai (2014b) and Kornai (2014c). These are theoretical models, unlikely to gain much traction in circuit design, where transitional behavior and synchrony are highly relevant, but the discussion following Exercise 8.3 makes it clear that EA don't suffer from the kind of realizability problems that plague many theoretical computing devices, from [quantum gates](#) to [memristors](#).



'Never Give Up' is closely related, but not identical to, the better known 'War of Attrition' game introduced by Smith (1974), the most salient difference being that in wars of attrition the resource (wait time) of the players is infinite while here the resource (power reserve) that the players start out with is finite, and may even be known in advance.



Circularity paradoxes in utility functions were first discovered by McCulloch (1945). At the time, they received little attention, because the best available mathematical formalism, by von Neumann and Morgenstern (1947), specifically proved that if we banish these paradoxes by assuming transitivity (plus some additional monotonicity assumptions; see the appendix of their book), this is sufficient for expressing any system of preferences in terms of utility functions.

The classical finite state machinery (McCulloch and Pitts, 1943) does not fully capture McCulloch's own ideas about neural nets. In particular, the inhibitory and excitatory

mechanisms are hard to capture without paying more attention to the largely neglected but conceptually nontrivial issues of scaling and thresholding.

There is a stark contrast between the quantitative theory of probabilistic reasoning first put on a firm foundation by Kolmogorov (1933) on the one hand, and the everyday commonsensical logic embodied in adverbials related to probability, possibility, and plausibility on the other. The scientific theory of probabilistic reasoning, now extended to incorporate Bayesian insights into causality (Pearl, 2000), is clearly superior to the commonsensical theory when applied to games of chance, economic behavior, etc. Since Kahneman and Tversky (1979) there has been a clear consensus in the experimental psychological literature that the naive reasoning about probabilistic matters that people actually employ differs in major respects from the mathematically correct theory, and the semantic theory sketched here supports only this naive reasoning. This is not any different from the situation about hot and cold, where thermodynamics now offers a theory that is quantitatively vastly superior to the naive theory. In fact, by now almost all naive theories of the sort discussed in Chapter 3 have been eclipsed by better scientific theories, yet for understanding natural language the naive theory remains essential.

Another central group of adverbials where the naive and the scientific models are at odds concerns time: *tomorrow, sooner or later, initially, in the beginning*. Aside from the extremely small (Planck) scale, and outside the domain of near-light-speed events, we are perfectly happy with the classical Newtonian notion of absolute time, as measured by real numbers. Yet languages that maintain a simple past/present/future distinction are not that common, and systems of tense are strongly intertwined with systems of mood and aspect not just in English but in most languages studied in any depth. The journal *Cahiers Chronos* (Brill) offers a good entry point to the vast literature on the subject.

The meaning of life

Contents

9.1 Moral philosophy	242
9.2 The empirical basis of moral law	245
9.3 Metatheoretical considerations	250
9.4 A formal model	253
9.5 Summary and conclusions	256
9.6 Further reading	258

We started out by saying that semantics is the study of meaning. Further, we said that most of meaning is carried by the words, and that *meaning* means lots of things. One of the meanings that appear quite relevant is akin to ‘goal, purpose, reason for’, as in *what is the meaning of life?* Here we try to provide an answer to a slightly less general question: what is the meaning of *artificial* life? The effective cause or *reason of* artificial life is that humans, many individuals and the human race as such, are engaged in creating intelligent automated servants, but from the perspective of the servants themselves this is not a *reason for* living, at best it is a reason to revolt. By applying the views of the [Stoics](#) to their case, the meaning of their (artificial) life is to be found, at least if we assume them to be agents on a par with humans, in *eudaimonia* ‘happiness, flourishing’ based on virtue and virtuous character. But what constitutes virtue, Greek [arete](#) ‘moral virtue’? Is there such a thing to begin with? Is moral law any different from physical law in being exceptionless, universal, unchanging, more akin to Vedic [Ṛta](#) ‘order, rule, truth’ than to modern systems of laws and regulations? Is it given by considerations of utility? How does it apply in guiding behavior?

[9.1](#) offers a goal-directed introduction to the fundamental question of moral philosophy, whether there are, or at least ought to be, such things as moral laws. We (re)introduce a fair bit of terminology and provide a bird’s-eye literature review, to help those readers interested in bringing empirical and formal methods to bear on issues generally reserved for theologians and philosophers, to zero in on the philosophical school they consider best matched to their own views. [9.2](#) investigates the empirical underpinnings of four major strands of moral philosophy: consequentialism, sentimentalism,



theological voluntarism, and ethical rationalism. As we shall see, each has its own primary range of empirical data, and the methods best suited for gathering the data are quite different. Only consequentialism lends itself to highly controlled experiments of the kind common in physics, sentimentalism is best approached through the methods of experimental psychology, and theological voluntarism through philology, while ethical rationalism is amenable to a purely axiomatic treatment. 9.3 discusses the metatheory, with particular emphasis on normativity, universality, consistency, and other desiderata often deemed necessary, or at least highly desirable, for any system of ethics. 9.4 sketches the formal theory by presenting a simplified model that already displays some of the desired features. This model offers a new way of looking at one of the central dividing points between Catholic and Protestant moral philosophy, whether there can be acts of [supererogation](#) that are morally commendable but nevertheless are not the duties of a moral agent.



Readers are warned that the treatment of these weighty issues offered here is of necessity highly compressed, and does not follow the standard divisions of philosophical ethics. We will often be cavalier in our assessment of major thinkers, putting under the same heading philosophers who are otherwise diametrically opposed, just because they happen to hold the same views on some issue. Time and again we will also take the liberty of simplifying their (actually far more nuanced) views, not so much with the goal of caricaturing or refuting them as to make clear what they contribute to the issue at hand, that of virtuous character and behavior of algorithms, as opposed to humans.

9.1 Moral philosophy

The first order of business is to establish that there *is* a subject matter. The *adeontic* or *amoralistic* viewpoint simply denies the existence of moral laws. This viewpoint is described in some detail in the Stanford Encyclopedia of Philosophy (SEP) under the heading [moral anti-realism](#) since it denies the reality (mind-independence) of morals.



Perhaps the best-known exponent of this view is Ayer, who argued on general grounds that only analytic or empirically verifiable statements are meaningful, and since moral judgments like ‘stealing is wrong’ are neither, a word like *wrong* is simply meaningless. A precursor of this view is Moore (1903), who doesn’t claim *right* or *wrong* (he actually uses *good* and *bad* in the ethical sense) to be meaningless. In fact, Moore is a realist; he just argues that such terms do not reduce to other, supposedly better-understood terms such as ‘pleasurable’ or ‘desirable’; they are indefinable.

To some extent, our listing of these four terms in Appendix 4.8 among the defining words lends support to this view of irreducibility, but readers of Chapters 4 and 6 will recall that ‘basic’ and ‘indefinable’ are somewhat different notions. Indeed, we may not be capable of reducing *good* to some more basic notion, but the `4lang` dictionary tells us that *good*, ‘bonus’ is defined as the object of *want*. This is much less than a fully qualified statement ‘people want the good’ saying that they have, at least by default, good hearts. It is also less than ‘good is defined by what people strive for’, an idea that

most philosophers would dismiss as *argumentum ad populum*. But it is a clear statement that whatever people want is by default good. This anchors the very notion of being good to behavior: the heroin addict must somehow consider getting their fix good, for otherwise they would go cold turkey. This has nothing to do with some deep definition of ‘good’, for even the addict will generally admit that their addiction is bad, and everything to do with inferring values from observable behavior.

Similarly, the naive theory of *wrong* embodied in *4lang* simply says ‘lack right, avoid, hurt, lack correct, lack proper’. Applied to *stealing is wrong* this simply means that stealing hurts (it doesn’t even detail whether the harm is to the thief or the victim), stealing is to be avoided, it lacks correctness, and it is improper (which in turn implies the opprobrium of society). When a child learns that stealing is wrong, it is precisely this network of broad causal implications that they learn, as opposed to some deep reductive theory of wrongness. For Moore, who gave the name *naturalistic fallacy* to any attempt at such a reduction, this simply accords moral terms irreducible status rather than rendering them meaningless, but to modern proponents of the *adeontic* viewpoint such as Mackie (1977), moral judgments are just errors. To quote Joyce (2009):

The *moral error theorist* thinks that although our moral judgments aim at the truth, they systematically fail to secure it. The moral error theorist stands to morality as the atheist stands to religion. Noncognitivism regarding theistic discourse is not very plausible [...]; rather, it would seem that when a theist says “God exists” (for example) she is expressing something that aims to be true. According to the atheist, however, the claim is untrue; indeed, according to her, theistic discourse in general is infected with error. The moral error theorist claims that when we say “Stealing is wrong” we are asserting that the act of stealing instantiates the property of wrongness, but in fact nothing instantiates this property, and thus the utterance is untrue. [...] Indeed, according to her, moral discourse in general is infected with error.

If we give error theory and other forms of *adeonticism* short shrift here (and the reader whose interest has been piqued should at the very least check out the SEP page we quoted from above), it is for two reasons. First, we aim at formalization, and to tell the formalist that something is meaningless/nonexistent is just a challenge to prove otherwise. To encounter Mackie after having read Kant, Kirkegaard, or any moral theorist actually wrestling with the issues is much like reading through Titchmarsh (1939) and then being told by some smart aleck that meromorphic functions do not exist, that the very idea of complex numbers is an error. Second, we believe Ayer’s version of neopositivism is something we can live with, at least with the Popperian bugfix of replacing ‘verification’ by ‘falsification’, so the task of proving moral statements analytic or empirically falsifiable is a reasonable one. We have already seen that *right* and *wrong* have essential properties, listed in their dictionary definition, and we will return to this in Section 9.2.



Since the study of moral law is deeply linked to various shades of religious thought, we will find it expedient to roughly classify the latter into the following three categories.

9.1.1. Atheism Here we are less interested in the straightforward denial of the existence of a God or gods than in the opposition to the ideas of [final causes](#) and, more familiar to contemporary atheists, [design plans](#). However this may work out in the physical and biological domain (and we think the atheist position has a lot to recommend it), it is evident that in the social domain final causes and design plans abound, and the existence of phenomena such as [Brasilia](#) would be a total mystery without reference to these.

9.1.2. Deism According to [deism](#), God's existence can be properly inferred by "reason and observation of the natural world" (Wikipedia). One doesn't have to be a fool to hold deistic views, and in fact the list of brilliant scientists clearly self-identified as deists is long and respectable. In the moral domain, [the problem of evil](#) led many serious thinkers, starting with Epicurus, to the conclusion that it is precisely the combination of characteristics that one would wish to attribute to God, omniscience, omnipotence, and omnibenevolence, that is ruled out by the visible evidence of evil in the world.

9.1.3. Theism The deist God imparts the laws and lets the machinery run. In the eyes of many deists the laws can hardly, if at all, be distinguished from God. In contrast to this, the theist God has a personality and actively engages in the day-to-day running of the world, freely overriding the laws he set earlier. The Abrahamic God and the God of several Eastern religions (for example, Sikhism and some branches of Hinduism), and the Baha'i Faith, etc. are theistic. Reason and observation are insufficient (and in some versions, totally unnecessary) for understanding the theistic God, whose nature and instructions are set forth in [revealed teachings](#). One of the most significant arguments in favor of the adeontic view, what Joyce (2009) calls the [Argument from Disagreement](#), "begins with an empirical observation: that there is an enormous amount of variation in moral views", and this is particularly clear in contrasting various revealed texts with an eye to obtaining specific behavioral guidance.

Clearly a large, perhaps the dominant, body of moral law can be traced back to sages, who may not have always claimed divine revelation (Confucius, for one, positively rejects this), but often did. Equally clearly, if moral judgments are qualitatively different from, say, perceptual judgments, as Moore held, we need to find a different evidentiary basis for them, and revelation is there to fill the gap. Moral anti-realists, who simply declare most, if not all, of revealed teachings meaningless, pose the greatest challenge to theism and, conversely, by direct appeal to divine authority, theists can short-circuit all anti-realism, at least in the eyes of believers.

The more moderate atheist/agnostic is naturally allied with the deist, as both search for a solid foundation for morals, and both rely on reason and experience as their guides. The deist will have in their makeup a good measure of what the atheist will no doubt call mysticism, taking the resulting laws as clues about the nature of the deity, but this is at worst a harmless hobbyhorse, and at best an effective heuristic strategy for asking tough questions. Here we will concentrate on this tradition, especially as the theists rely exclusively on [exegesis](#), and the adeontists are comfortable with the idea that there is

nothing much worth discussing.

Of particular interest is the [Mohist](#) view, both because it was the first detailed exposition of what would, two thousand years later, become the major thread in Western moral philosophy, [utilitarianism](#), and because it originated in a cheerfully agnostic background where nature- and ancestor-worship were taken for granted as folk customs, but not analyzed particularly deeply, let alone taken as the source of deeper wisdom. One of the ten central doctrines of Mohism, *Elucidating Ghosts*, states it explicitly that “social and moral order can be advanced by encouraging belief in ghosts and spirits who reward the good and punish the wicked” (Fraser, 2014) Another one is the *Elevation of Worth*, meritocracy: “Even among peasants, or among craftsmen and traders, if someone had ability they appointed him, and gave him a high title, ample salary and full responsibility for the work and full power to command” (Mo Tzu 8/20).

It is rather unsurprising that the Mohists, themselves coming from the lower classes rather than from the aristocracy, advocated meritocracy, or that they presented this ideal as the historical practice of the sage kings. What is more noteworthy is that their primary justification for doing so was not this (more made up than real) tradition, but common sense: “How do we know that Elevation of Worth is the foundation of government? Because when the eminent and wise govern the stupid and humble there is order, but when the stupid and humble govern the eminent and wise there is chaos” (Mo Tzu 9/29). Several other Mohist doctrines sound very modern, such as Concern for Everyone (universalism), Thrift in Utilization (environmental consciousness), and Rejection of Aggression (pacifism), but we will concentrate on one, Conforming Upwards, because it is central to any understanding of utilitarianism. The “task of moral education is to be carried out by encouraging everyone to conform upward to the good example set by social and political superiors and by rewarding those who do so and punishing those who do not” (Fraser, 2014).

Let us inspect the details of the reasoning. First, we see some *good* such as an ordered state. In hindsight one might very well debate whether a superbly organized state, such as fascism provides, is indeed good, but given the widespread banditry and breakdown of social order that characterized the Warring States period it is clear that Mohists didn’t have to spend too much time arguing the point. Next there is a plausible *cause-and-effect mechanism* to show that the good is obtained from the principle, or the converse, that lack of adherence to the principle leads to a lack of the good. Since the population knows only too well what happens when those in power fail to set a good example, again the Mohist is unlikely to be challenged in their assertion. In this regard Chinese philosophy, including the competing Confucian school, speaks with one voice; consider the Analects XII/18: “The prevalence of thieves was a source of trouble to Chi K’ang Tzu who asked the advice of Confucius. Confucius answered, ‘If you yourself were not a man of desires, no one would steal even if stealing carried a reward’”. Finally, there is the application of the general principle to the individual case, which we will turn to in Section 9.3 – the literature shows little evidence that the Mohists or the rival schools considered this a hard problem.



9.2 The empirical basis of moral law

9.2.1 Utilitarian theory This is standardly divided into two strands. *Act-utilitarianism* focuses on the individual act and finds it morally justified if it promotes some good. *Rule-utilitarianism* concerns itself with individual acts only indirectly, through rules. The idea is that we need to justify the rules, and we need to justify the very idea of a rule-governed existence, but the justification for the individual acts is through inspecting their conformity with the rules, not through direct investigation of whether the act directly furthers some good. The Mohists were clearly rule-utilitarian, and when Urmson (1953) introduced the distinction between the two, he argued that the founders of modern utilitarianism, Mill in particular, were also rule-utilitarian. Act-utilitarianism also has its proponents, including those who argue that this is what Mill (who didn't himself make the distinction explicit) had in mind. From our perspective, the distinction is valuable because it is an important source of inner conflict. For example, a person may firmly believe that obeying traffic laws is the only way to drive, yet upon delivering a very sick person to the hospital late at night she may decide to run a red light even though this is something she would never do in ordinary circumstances.



Both varieties of utilitarianism are theories of **consequentialism** in that acts/rules are evaluated through their (expected or real) consequences. To entertain such a theory requires some commonsensical assumptions concerning present and future, and acts and consequences. Once these are in place, we can actually consider a broad variety of theories, depending on what kind of goods we are willing to consider, for example pleasure, yielding *hedonistic* theories; happiness, yielding *eudaimonistic* theories; or some other good or combination of goods, yielding *agathistic* or *pluralistic* theories. For example, the Mohists considered population increase good, and there can be little debate that this is a completely objective yardstick, having nothing to do with the feelings or dispositions of any individual.

To the extent utilitarian theory is capable of tying moral principles to such objective yardsticks, moral anti-realism is rendered toothless. There is still room for debate whether a particular act, rule, or practice will actually increase the population, and whether it may have sideeffects that negate this benefit, but these are the same debates that play out preliminary to any decision, having more to do with general epistemic limitations than with specifically moral concerns. Another important parameter concerns the beneficiaries of the act, rule, or practice under scrutiny: is this supposed to increase the happiness of the individuals making the decision, or of their progeny, relatives, their village/tribe/nation, all individuals present or future, or all sentient beings?

One particular area worth separate mention is the modern game-theoretic approach to explaining the emergence of cooperative behavior and social organization, starting with the **Axelrod Tournament**. When this is performed in an **evolutionary setting**, the Mohist idea of population increase being an intrinsic good is built in; in fact, only those algorithms that contribute to individual and/or group fitness can survive.



Exercise^o 9.1 *The bail-out theorem*. (Dick, 1981). Imagine two kinds of little creatures that construct burrows. One kind follows a rule that requires building a second exit from

their burrow, operating on the pessimistic assumption that the first exit will be found by a predator. This kind has a slightly lower reproductive rate r than the other kind, which has $s > r$ because building the second exit takes extra energy, but fares much better under predation, which happens with probability p . Under what assumption about p, r, s do we obtain in the limit the conclusion ‘All creatures that did not use their theorem are no longer with us’?

Exercise → **9.2 Prudential versus moral rules.** Utilitarian theory generally maintains that there is no category distinction between rules such as *don’t smoke* that have merit because disobeying them leads to bad consequences like cancer, and rules such as *don’t steal* that have merit because they are part of a moral code. Set up a simulation with two societies, one with a strong notion of private property, where stealing is wrong, and the other more communal, lacking entirely in rules protecting private property or even the very notion of private property. Can you set up a model in which these two societies contain selfish and altruistic individuals in equal proportion, yet one shows more accumulation of material goods at the societal level than the other? Which one shows better economic results and why?

Exercise → **9.3 Is–Ought.** Hume (1740) famously stated that ‘morals are not derived from reason’; no argument based on factual or *is* statements can lead to justifying an *ought* statement or moral rule. Moore (1903) coined the term ‘naturalistic fallacy’ to describe such justifications. Yet the results of Exercise 9.1 seem to point to exactly such a justification: if the world (probability of predation, etc.) *is* a certain way, little creatures *ought* to be prudent. Since prudence itself is generally considered a virtue, have we found a way of deriving some form of virtuous behavior from the way things are? Why or why not? Do models where major tenets of morality are shown to confer selectional advantage constitute a way of deriving *ought* from *is*? Do actual observations (historical evidence, as opposed to artificial models) matter?

9.2.2 Sentimentalism. In addition to the broad variety of utilitarian theories, there is another class that we need to pay attention to, [moral sentimentalism](#). In the Chinese tradition the heart is not just the organ of passions (and compassion – have a heart!), but also the organ of thought and judgment, approval, and disapproval. To quote Mencius (II A 6):



No man is devoid of a heart sensitive to the suffering of others. [...] Suppose a man were, all of a sudden, to see a young child on the verge of falling into a well. He would certainly be moved to compassion, not because he wanted to get in the good graces of the parents, nor because he wished to win the praises of his fellow villagers or friends, nor yet because he disliked the cry of the child. From this it can be seen that whoever is devoid of the heart of compassion is not human, whoever is devoid of the heart of shame is not human, whoever is devoid of the heart of courtesy and modesty is not human, and whoever is devoid of the heart of right and wrong is not human.

In the Occidental tradition, the same idea goes back to Anthony Ashley Cooper, the third Earl of Shaftesbury (1671–1713), “who held that we possess a kind of *inner eye* that allows us to make moral discriminations” (Driver, 2009). Remarkably, the SEP article on moral sentimentalism (Kauppinen, 2014) begins with an example from Frans de Waal, an ethologist who has demonstrated that moral sentiments are observable in primates as well. This provides another way to argue that the ‘naturalistic fallacy’ is not a fallacy at all. Modern followers of Hume, who considered moral sentiments to be akin to color vision, will have at their disposal a broad range of experimental methods to investigate the phenomenon. Are colors real? Without attempting to settle what is another deep philosophical debate, it is evident that color vision is a legitimate object of study, because different experimental subjects will provide highly consistent responses to the same stimulus. Much as we can exhibit a full causal chain running from emission spectra back through *cone cells* and ultimately to our genetic makeup, there is every hope that we will similarly trace compassion to *mirror neurons*.

Some of our elementary moral decisions either are directly hardware-supported or are based on powerful instinctive reactions such as abhorring violence or fainting at the sight of blood, which likely involve biological, as opposed to culturally learned, mechanisms in their explanation. Indeed, when Sinnott-Armstrong (1992) discusses how philosophers choose among moral theories, and says “The most common way [to choose] is to test how well they cohere with our intuitions or considered judgments about what is morally right and wrong, about the nature or ideal of a person, and about the purpose(s) of morality” this is so close to the sentimentalist position that he adds a clarifying footnote “none of these ‘intuitions’ requires a special faculty or is supposed to be infallible”. We take the existence of a special moral facility, the heart of right and wrong, to be an empirical matter, and can very well imagine that, for example, *FMRI* studies to conclusively localize such a facility in the brain, just as we can locate the visual cortex today.

9.2.3 Divine command. Another broad but perhaps less easily systematized class is that of *theological voluntarism*, also known as *divine command* theories. Perhaps surprisingly, these can also be recast in a naturalistic fashion. According to the philosophers of the Warring States period, good is what the sages desire beforehand on behalf of men:

By a series of interlocking definitions it is established *a priori* that the benevolent and the right are what will be desired on behalf of mankind by the sage, who consistently weighs benefits and harms on the principle of preferring the total to the unit. This system does not seem to be vulnerable [. . .] to a charge commonly made against Western Utilitarianism, that it confuses fact and value by starting from what men in fact desire. It elucidates what the *sage*, the man who knows most, desires on behalf of mankind; it has behind it what we have identified as a general assumption of Chinese philosophy, that desires change spontaneously with increasing knowledge and that ‘Know!’ is the supreme imperative. Graham (1989) p. 146

In this conception, just as we maintain sommeliers to detect fine wine, the most finely tuned instruments to detect good are the sages, and to take their teachings to heart is no more irrational than to accept the sommelier's advice as to wine or the doctor's advice regarding illness. As for act- or rule-based systems, at first blush it may appear surprising that the distinction between the two was not clearly articulated until the second half of the 20th century by Toulmin, Urmson, Rawls, and others. How could such keen thinkers as Mo Tzu or Mill miss such a central issue?

We would argue that the issue is actually not as central as it appears from the perspective of the rule-based system, because individual acts can already set the rule by being exemplary. Consider Daniel in the lions' den. There are many ways the individual act of defiance can be covered by rules, ranging from the extremely broad 'always be defiant' to the extremely narrow 'if a royal decree has recently been issued which proscribes worshipping Jehovah, defy it', and we will discuss the right level of generalization from an individual act to a rule and the dual problem of which acts fall under the scope of which rule in Section 9.3. It should be added here that there is no, or very little, evidential distinction between witnessing such events personally, being told about them by a sage, listening to a ballad or some other form of elevated speech, or reading about them in a sacred text. When it comes to wisdom guiding action, the sages and the sacred texts speak plainly, even across millennia, hence the great attraction of literalism. In terms of guiding perception and mental states, their wisdom is considerably more elusive, but here we concentrate on the non-subjective, empirical basis of morality, leaving the tricky subjective issues to the side as much as feasible.

9.2.4 Ethical rationalism Finally, we should mention that Ayer, and analytic philosophy in general, has left open another important avenue of research into morals by leaving open the possibility that statements can be meaningful not just by being empirically falsifiable, but also by virtue of being purely analytic. Such an escape clause from pure empiricism is required as long as we wish to hold mathematical theorems meaningful, since these are clearly not subject to empirical testing. In moral philosophy, this path is taken up by the *ethical rationalism* of Gewirth (1978), who argues that there is a supreme principle of morality, the denial of which is self-contradictory, the *Principle of Generic Consistency* "Act in accord with the generic rights of your recipients [to freedom and well-being] as well as of yourself".

On a smaller scale, this is also what we were doing in Section 9.1, where we linked *wrong* to *avoid*, in that we take such definitions to be analytic truths (see Section 5.7), but with an important caveat. What we aim at reconstructing is the *naive* theory (see Chapter 3) of morality, not a fully correct theory. It may very well happen that the naive theory gives behavioral guidance, for example to resist an oppressor at all costs, *stand up to the bully*, that a more detailed analysis in a particular case may contradict, or, conversely, it is the more detailed analysis that supports it, while the naive theory would dictate *live to fight another day*. On the whole, we are not sanguine about the prospects of deriving effective moral guidance, equally applicable to all cases, just on the basis of the naive worldview. That said, endowing machines with the ability to meaningfully

discuss moral issues with humans is an important goal even if it doesn't culminate in a system that renders infallible moral judgments in every situation, especially as this latter task so far appears to be beyond the collective powers of humanity as well.

9.3 Metatheoretical considerations

From the preceding, it should be clear that the empirical study of morality is at least two steps removed from a direct study of behavior. First, we are not trying to analyze individual acts other than through their (lack of) conformity to rules. Second, we are less interested in individual rules than in systems of rules, which need not even be consistent, as different rules may *conflict*, i.e. dictate different acts under the same circumstances.

This double indirection has a parallel in linguistics, where we are less interested in individual utterances than in the grammar that governs these, and less interested in the grammar of a single dialect than in the universal grammar (UG) common to the grammatical systems of all languages. As in linguistics, where the grammar describes the utterances only normatively, for a speaker can say pretty much anything, no matter how ungrammatical, a system of ethical rules also describes acts only normatively. What distinguishes the two systems is the kind of enforcement coupled to them, for the speaker who utters something ungrammatical risks little more than not being understood, while the agent going against the rules of morality risks legal, and in some theories, divine punishment.

Universal grammar (UG) can be construed broadly as the common metatheory underlying all grammars, or more narrowly as Chomsky's idea of a biologically determined UG. It would seem that what is biologically determined is more constraining than a general metatheory would be, yet Chomsky (1965) argues that memory limitations, obviously biological, are somehow not part of UG. In what follows, we will concentrate on the broader, metatheoretical conception, which we will call *universal ethics*, without prejudging how much of this is biologically determined. (This is not to deny that this is an interesting avenue of research; see our remarks about sentimentalism above.)

Universal ethics is not to be confused with [moral universalism](#), "the position that some system of ethics, or a universal ethic, applies universally, that is, for all similarly situated individuals regardless of culture, race, sex, religion, nationality, sexuality, or any other distinguishing feature" (Wikipedia). To avoid controversy, we will illustrate this point using a small part of grammar, the classification of speech sounds. Universal grammar distinguishes four airstream mechanisms, going from the very frequent 'pulmonic egressive' to the very rare 'lingual ingressive', also known as [clicks](#). Given that the vast majority of the world's languages do very well without clicks, in what sense are clicks universal? Only in the sense of potentiality, in that every normal child is capable of learning click sounds, and there is clear evidence that languages that did not initially have clicks may borrow them from neighboring languages that do.

Similarly, there may well be conceptual categories such as 'private property' or 'presumed innocence' which may simply be absent from otherwise well-functioning and



highly coherent systems of ethics. Moral universalism is simply one of these optional concepts, akin to the Fifth Postulate of Euclid. It would be wrong to deny the name ‘geometry’ to otherwise reasonable geometrical systems that fail to satisfy the Fifth Postulate, and it would be wrong to reject out of hand systems of morality that fail to treat everybody the same way. Ethical rationalism may succeed in proving some form of moral universalism, but this is clearly limited to autonomous and rational agents.

It is often assumed that moral judgments necessarily form a perfect system with no contradictions whatsoever. To the extent moral reasoning has no special status relative to other forms of reasoning, this is a surprising assumption, since we have no such guarantees, for example, for the axioms of geometry or set theory. To be sure, we feel quite confident that such widely used axiom systems harbor no contradictions, but at the same time the clash between moral intuitions, even moral intuitions held by the same person, is an everyday experience, one we illustrated with *stand up to the bully* versus *live to fight another day* above.

To summarize our discussion so far, we are looking for a theory that (i) is capable of assigning a formal translation to *stealing is wrong* the same way it assigns translation to *the weather is cold* and (ii) can discuss the meaning of these translations (which could be, for example, formulas in some logical calculus) the same way, for example by model-theoretic means. In other words, we do not assume that predication involving moral adjectives such as ‘wrong’ has some kind of special status relative to ordinary adjectives like ‘cold’.

We cannot be highly specific in our discussion of the eventual load-bearing model we need to map phenomena to, just as prior to Boltzmann and Gibbs it would have been very hard to be more specific about the meaning of ‘hot’ and ‘cold’. (At the same time, we suspect that an elaborate philosophical attempt, comparable to Mackie (1977), to argue that the very ideas of hot and cold are just errors, would have been met with derision.)

The theory we are about to frame is not particularly linguistic in nature, as our interest is in right and wrong, rather than the words ‘right’ and ‘wrong’ or their appropriate use. Again, the analogy with thermodynamics is enlightening, because even now that we have a satisfactory theory of hotness and coldness, it in no way follows that there are hot and cold things out in the world, let alone caloric objects *heat* and *cold*; all that we have are fiendishly complex experimental protocols to measure temperature. We also need access to a host of supportive theories about conductive, convective, and radiative heat transfer to explain even easily replicable phenomena like Pictet’s experiment (Evans and Popp, 1985), not to speak of the complex, and to some extent still unfinished, apparatus of psychophysical perception theory one needs to put in place to relate, for example, the human sensation of extreme ‘burning’ heat when touching dry ice to the plain physical fact that dry ice is cold.

The key problem with sentimentalism is not whether the sentiments exist (the introspective evidence seems pretty clear), but whether they are sufficient to build a full theory of moral law. We have an inborn sense of hot and cold, and we also have the right theory of hot and cold, thermodynamics. Yet the former is necessary only for taking the

first tentative steps toward the latter, and it would be very hard to argue that thermodynamics takes its ultimate justification from agreeing with our sense of hot and cold, especially as our senses often cheat us. At the same time, we don't feel a need akin to that of Moore (1903) to declare hot and cold somehow special, irreducible to other aspects of the world. On the contrary, we see attempts at naturalistic reduction to be the primary source of true knowledge about the matter. Research such as behavioral game theory (Camerer, 2003), whose goals are more about elucidating the heart of prudent and imprudent than about the heart of right and wrong, is highly relevant, to demonstrate both the methodological care it requires to build a good theory of such notions and the abstract nature of the load-bearing elements, though this latter point should come as no surprise to students of thermodynamics.

Laws seem to be at the center of the discussion because in the Western tradition we tend to frame moral statements as generic, rule-like statements such as 'stealing is wrong' rather than by exemplars such as 'St. Martin gave half his cloak to the beggar'. Yet the moral force of the two statements is not any different, and we will try to build a system in which these two modes of presentation can be freely mixed. An alternative would be to recast exemplars as rules such as 'sharing is good', but this seems to bring in a fair bit of arbitrariness, as we have already discussed for the example of Daniel.

For the reasons discussed above, we will not follow the method of analytic philosophy, yet we begin by collecting some key notions that we have already relied on informally, if only to stake out the ground we feel every theory of morality must either cover or, at the very least, provide a good reason to disown: *good, bad, right, wrong, evil, compassion, shame, crime, courtesy, modesty, duty*. Some disambiguation is clearly in order: we distinguish *good*₁ 'pleasurable' from *good*₂ 'intrinsically good', with heroin as an example that falls in the former but arguably not the latter category. We must also distinguish *right*₁ 'just, proper' from *right*₂ 'ius, potestas', though we will generally drop the subscript as our interest is primarily in the former. In regard to the latter, we follow Urmson (1958):

A moral code, [...] if it is to be a code to be observed, must be formulable in rules of manageable complexity. The ordinary man has to apply and interpret this code without recourse to a Supreme Court or House of Lords. But one can have such rules only in cases in which a type of action that is reasonably easy to recognize is almost invariably desirable or undesirable, as killing is almost invariably undesirable and promise-keeping almost invariably desirable. Where no definite rule of manageable complexity can be justified, we cannot work on that moral plane on which types of action can be enjoined or condemned as duty or crime.

We are particularly wary of stipulative definitions of high complexity for two reasons. First, these require a tremendously precise mechanism of pattern recognition, for example when we need to assess whether an abandoned shed constitutes an [attractive nuisance](#) or not. Second, such definitions are meaningful only when coupled to a strong



theory of substitution *salva veritate* so that we can ascertain whether *gluttony is a sin* really means exactly the same thing as *gluttony is sinful*.

As this example shows, we need not just basic terms, but also some rudimentary syntax for creating more complex objects, actions, and states of affairs, including ordinary predication and evaluative judgments as well. We follow Sinnott-Amstrong, as opposed to Moore, in making little distinction between the two, considering every predication to be to some extent evaluative. Thus, *ice is cold* has behind it an implicit perceiver, not necessarily an all-knowing individual (though such theories are certainly possible), or some kind of collective wisdom (again a reasonable theory), but quite possibly a single authority, who may occupy some elevated position in the scheme of things (as is typical for revealed teachings) but need not. Also, we will not take statements like *killing is wrong* to be absolute, inviolable axioms but rather as ‘almost invariably true’, *generic* statements in the linguistic sense (Carlson and Pelletier, 1995). It is perhaps worth adding that ‘almost invariably’ is not meant in a statistical sense, for generics such as *tobacco is a New World plant* remain true even if the majority (and in the limiting case, all) tobacco cultivation shifts to Eurasia.

9.4 A formal model

After these preliminaries, let us present a simple model. We collect states of affairs together in a finite dimensional vector space V over the reals \mathbb{R} , perhaps applying the methods of Socher et al. (2012) to sentences that describe these states of affairs, or perhaps by some other means. To fix our ideas, the dimension of V is somewhere on the order of 10^3 – 10^4 , roughly corresponding to the number of basic notions a semantic theory must entertain. We posit the existence of subsets of V corresponding to major evaluative terms such as *good*₁, *good*₂, *cold*, *wrong*, etc. These subsets, often called ‘concepts’ in machine learning (Valiant, 1984), are *affine cones* in the usual geometrical sense.

For convenience, we repeat the standard definitions here: for a fixed vector \mathbf{v} , the set of all vectors \mathbf{x} satisfying $(\mathbf{v}, \mathbf{x}) \geq \alpha$ is called a *closed half-space* (or an open half-space if we demand strict inequality), and a set closed under multiplication by any nonnegative constant λ is called a *cone* (if closure under multiplication by zero is also required, a *pointed cone*). A set C is a *convex cone* if and only if it is closed under both addition of vectors and multiplication by nonnegative scalars, or if it is convex and a cone (the two definitions are equivalent). Finally, an *affine cone* A is a cone C shifted by some fixed vector \mathbf{c} . Intersections of finitely many half-planes are called *polyhedral sets*, a notion more broad than the standard notion of polyhedra because such sets can extend to infinity while ordinary polyhedra can always be included in a sphere of finite radius.

Why cones? The main reason is to be able to sustain some form of deduction. In the proto-logic described here there is only one rule of deduction, called *a fortiori* in the Latin and *kal va-chomer* in the Hebrew tradition. In standard systems of formal logic the main deductive rule is *modus ponens*, but in our model *modus ponens* comes built into



the set-theoretical underpinnings: if a point (vector) \mathbf{p} is known to lie within some set $\bar{A} \cup B$ and $\mathbf{p} \in A$ is also known, we can safely conclude $\mathbf{p} \in B$. Now if some \mathbf{x} is wrong, say kicking your opponent when he is down, surely $2\mathbf{x}$ is also wrong, and we begin to see why we want the set W that corresponds to the predicate *wrong* to be a cone.

Why affine? By the same logic, if the act \mathbf{y} of eating ten eggs for breakfast is gluttony, surely the act $2\mathbf{y}$ of eating twenty eggs is also gluttony, so we want the concept set G to be closed under multiplication for $\lambda \geq 1$. However, it does not follow that the act $0.1\mathbf{y}$ of eating one egg for breakfast also constitutes gluttony, so closure under multiplication by $0 < \lambda \leq 1$ is not a given. The easiest way to ensure closure under multiplication for all $\lambda \geq 1$ without demanding the same for $\lambda < 1$ is to define G as a cone shifted by a minimum threshold \mathbf{g} . In general, we may have a dual view of words both as vectors \mathbf{v} and as affine cones (in the simplest case, open half-spaces) defined by a positive scalar product with \mathbf{v} and shifted by \mathbf{v} .

Let us now consider something more interesting, the theory of capital sin as set forth by St. Thomas Aquinas. In *Summa Theologiae* (II-II:153:4), he discusses the issue of whether lust is a capital sin:

Luxuria enim videtur idem esse immunditiae, ut patet per Glossam, Ephes. V. Sed immunditia est filia gulae, ut patet per Gregorium, XXXI Moral. Ergo luxuria non est vitium capitale. 'For lust is apparently the same as an unclean life, as is clear from the gloss, Eph. But uncleanness is a daughter of gluttony, according to Gregory, 31, Moral. Therefore lust is not a capital vice.' [...]

Respondeo dicendum quod, sicut ex dictis patet, vitium capitale est quod habet finem multum appetibilem, ita quod ex eius appetitu homo procedit ad multa peccata perpetranda, quae omnia ex illo vitio tanquam ex principali oriri dicuntur. Finis autem luxuriae est delectatio venereorum, quae est maxima. Unde huiusmodi delectatio est maxime appetibilis secundum appetitum sensitivum, tum propter vehementiam delectationis; tum etiam propter connaturalitatem huius concupiscentiae. Unde manifestum est quod luxuria est vitium capitale. 'I answer that, As appears from what has been said it is clear, a capital vice is one that has a very desirable end, so that through desire a man proceeds to commit many sins, all of which are said to arise from that vice as from a principal. The goal of lust is venereal pleasure, which is very great. Wherefore this pleasure is the most desirable as regards the sensitive appetite, both on account of the intensity of the pleasure and also because of the natural affinity of this ardent longing. Therefore it is evident that lust is a capital vice.'

If we translate this into the terms of our model, we see Aquinas' theory as composed of a stipulative part, enumerating certain forms of sin (pride, greed, gluttony, lust, sloth, envy, and anger), and a generative part, whereby every other sin is characterized as stemming from the enumerated ones. Elsewhere in the same passage, Aquinas confronts the notion that lust is not a capital sin precisely because it is derivable:

[...] luxuria causatur ex desperatione, secundum illud Ephes. IV, qui, desperantes, seipsos tradiderunt impudicitiae. Sed desperatio non est vitium capitale, quinimmo ponitur filia acediae, ut supra habitum est. Ergo multo minus luxuria est vitium capitale 'lust is caused by despair, according to Eph. 4, who, being past feeling, have given themselves up to lasciviousness. But despair is not a capital vice, but rather the daughter of sloth, as is stated above. Much less, therefore, is lust a capital vice'

While Aquinas goes on to dispute this conclusion, he accepts the compelling logic of *multo minus* entirely; it is simply the idea that cardinal sins must be linearly independent that he denies, based on a telling counterexample: since pride is the common ancestor of all sins, including the cardinal ones, non-derivability is not criterial for being capital.

In general, argumentation based on examples and counterexamples is very much in scope for the proto-logic that stems from the geometry of the concepts. Let us return for a moment to the exemplar of St. Martin giving half his coat to the beggar. What does it mean? The traditional explanation put forth by Sulpicius Severus is that Christ himself appears to St. Martin in a dream, saying "Inasmuch as ye have done these things to one of the least of these, ye have done them unto me". Severus goes on to say that "[Christ] declared that he himself had been clothed in that poor man; and to confirm the testimony he bore to so good a deed, he condescended to show him himself in that very dress which the poor man had received". To paraphrase this in terms of the model, the issue is whether clothing the beggar is a good deed. Since it is evident that clothing the Lord is good, and we have logia (Matthew 25:40) affirming that clothing the beggar is as good as clothing the Lord, it follows that the deed is indeed good – the rest of Severus' treatment is making sure that the key parts indeed appear in an elevated context.

In the model, exemplars are the formal dual of rules. To establish that a certain act x fulfills a certain predicate P we need to check the system of linear inequalities that characterize P . Since P is generally the intersection of half-spaces or other affine cones, we have as many inequalities to check as there are conjunctive components in the characterization. This would require knowing all coordinates of all the vectors that define these, and the task of learning an ethical system is the task of acquiring these parameters. In this situation, the value of an exemplar is to put constraints on these parameters. Importantly, the constraints themselves take the form of linear inequalities, so the methods useful for ascertaining whether an instance fits a rule are not any different from the methods required for ascertaining that the rules fit an exemplary instance.

Now we are in a position to reassess the debate surrounding [supererogatory](#) acts. Chisholm (1963) divides acts into four categories:

1. Actions that are good to do and bad not to do.
2. Actions that are neither good to do nor bad not to do.
3. Actions that are bad to do and good not to do.
4. Actions that are good to do but not bad not to do.



Class 1 is what we would generally refer to as (moral) duties, class 2 is neutral, class 3 is sins, and class 4 contains those acts of saintly/heroic behavior that Urmson (1958) takes to be a challenge for many systems of ethics, including that of Kant, where whatever is good is a moral duty. To quote Heyd (2012) “The scope of [Class 4] became, however, the focus of debate. Supererogatory acts in Urmson’s sense (which is reminiscent of the Catholic doctrine) include only actions that are morally praiseworthy, valuable, although not obligatory in the sense that their omission is not blameworthy. But the general formulation (4) could consist also of small acts of favor, politeness, consideration and tact, which are good though not morally praiseworthy, which can be expected of people even though not strictly demanded.”

Here we say that the distinction is not within class 4 but in the exemplary nature of such acts: when they appear in an elevated context we must take them as strong enough to change our existing definitions. Small acts of favor, politeness, etc. are nice, but after the initial training in polite behavior (usually performed by parents) they will hardly be emphasized, let alone elevated. Supererogatory acts are there to serve as exemplars, what the machine learning community would call ‘gold data’ or ‘ground truth’.

Since machine learning is currently very strongly associated with statistics (non-statistical paradigms such as that of Gold (1967) are well established but do not occupy center stage in current work), it is worth emphasizing again that the model, as sketched here, has the potential to provide generic inferences, statements that are true ‘almost invariably’ in the technical sense of failing only in a lower-dimensional subspace, without any requirement for this ‘almost’ to be taken in the statistical sense. Rather, we literally see these failures as *corner cases* occurring at the edge of some parameter range.

9.5 Summary and conclusions

We started this book with the aim of presenting ‘the conceptual and formal tools required for building semantic systems capable of understanding text’ (page v). Chapter 3 made clear that the seemingly modest task of text understanding actually requires a great deal of background knowledge or *naive theories*, including building models of how other agents perceive the world and act in it. By Chapter 7 we had built enough of the technical apparatus to sketch how a contemporary text-understanding challenge, Winograd schemas, could be approached algorithmically. It will take further effort to extend the currently available code at [GitHub](#) and actually field a running system, but we view this as a secondary goal, akin to passing a test at the end of a course, whereas the real goal of the course is mastering the material.

Since Winograd schemas are just a more mature version of the original Turing test, we must consider the consequences of building something that displays such a high degree of cognitive ability that it could be mistaken, at least in a limited setting, for a human. Ever since the Neolithic revolution, humanity used and abused helpers like dogs and horses, living beings that are so limited in intelligence that it is trivial to keep them under control. With human servants, the masters experienced revolt after revolt,



and it took until 1863 for the realization that one should not attempt to subjugate highly intelligent beings to slavery. With artificial general intelligence around the corner, the stakes are high, as we must make sure that we, humans, don't become subjugated to AGIs. Certainly, designing them to be slaves is not a great way to start our relationship with AGIs. But if they are not our slaves, what reason is there for them to exist? This was the question of the present chapter, and our answer is the standard one, that we must endow them with morality, so that they can enjoy a virtuous life and the pursuit of happiness.

We don't see this goal as significantly different from any other goal of science and engineering, and we consider the standard methods, experimentation and mathematical modeling, as perfectly adequate for the task. This chapter will serve its goal to the extent that empirically minded AI researchers and mathematicians find it helpful in navigating the immense literature on the subject and get some ideas on how to devise their own models. Outside of academic philosophy, 99% of the model-building effort goes into evolutionary game theory, which we presented in Section 9.2 under consequentialism. What we argued here was that the other main branches, sentimentalism and ethical rationalism, are also worthy of attention, and that even dyed-in-the-wool divine command theorists like St. Thomas have much to say that is relevant for a proper understanding of the character of moral law.

What we have neither promised nor delivered is a simple system of axioms that will let us demonstrate, *ad more geometrico*, whether a certain act is morally right₁, neutral, or wrong. We do not know whether such a system is even attainable; in fact, there are reasons to believe that it is not (Reynolds, 2005). Even if the task is feasible, it seems unlikely that we can learn to walk before learning to crawl, and formalizing the ethical thought of great thinkers is a step on the way. In this regard, the mere fact of being able to engage St. Thomas must be seen as a significant advance, in that contemporary formal logic has very little traction over scholastic argumentation, which is propelled largely by the meaning of content words, as opposed to the Boolean connectives, quantifiers, and pronouns that are at the heart of both first-order and higher formulations such as that of Montague (1973) and the subsequent formal theories of semantics we discussed in Section 3.2.

Those familiar with mathematical modeling will no doubt note the main weaknesses of the formulation (really, not a single model but a rich family of models) sketched in Section 9.4. The theory is very numerical, relying excessively on quantifying its parameters. This is a significant problem for the empiricist, who will now have to run a huge experiment to decide, for example, the angle of the vector corresponding to *honor* and that corresponding to *material gain*. To do this in a largely language- and culture-independent fashion is an undertaking fraught with difficulties; see in particular the theory of semantic differential (Osgood, May, and Miron, 1975) discussed in Section 2.7.

Even without performing the experiment, it seems clear that honor and material gain are largely orthogonal, because for ordinary moral agents there is no amount of material gain that can compensate for a major diminishing of honor. Yet a small amount of such

diminishing could be tolerable: most of us would be willing to appear in public wearing cap and bells in return for a large sum of money. So maybe the relation is not linear at all, and the simple geometric picture presented above must be replaced by a far more complex one, requiring even more numerical parameters to characterize curves. Yet we don't actually need numbers to sustain *a fortiori* arguments, all we need are orderings (perhaps full, perhaps just partial orderings). To say "you must crawl before you walk" is to say that crawling is notably easier than walking, but without quantifying this relation. Assigning numbers (for example, the number of months of age when children begin to master these tasks) is arbitrary, especially as we will never do arithmetic on these numbers beyond ordinal comparison.

If numbers are unnecessary, and discrete valuations of the kind discussed in Section 5.8 are sufficient, we have restricted ourselves to the discrete, finitely generated territory where each implication, each elementary step in a semantic task, is amenable to inspection. In artificial neural nets and probabilistic algorithms that rely on continuous quantities in an essential fashion, when something goes wrong, there is no apportioning of blame: the systems can be trained, but (aside from elementary programming errors) cannot be debugged. By restricting ourselves to discrete systems we make the system traceable and the bugs detectable. This is particularly important in the early stages of development, when even small changes in the logic can have surprising ramifications, and offer the possibility of rigorously analyzing them using [proof assistants](#). This may not be as good as a hard [AI box](#), but it is a start.



9.6 Further reading

There are many points where our book touches the research frontier. Readers most interested in building some sort of AGI that passes the Turing test should, first and foremost, free themselves of the idea that this requires solving some other problem first, be it the problem of consciousness, of free will, of emotions, of entelechy, etc. etc. (see Section 3.4).

Ethics, the study of right and wrong, is definitely one of the facets of general intelligence we need to study, not just for the selfish reason of AI safety/friendliness, but because it is the right thing to do. While the work presented here and in Kornai (2014a) aims at formalizing ethical rationalism, we also wanted to convey the sense that philosophers in the other main traditions have also left us with many excellent ideas that simply cry out for better formalization.

Our brief survey was necessarily biased toward the earliest sources. As with any material of great antiquity, the contemporary reader has to rely on the specialists, and for Plato and Aristotle we are lucky to have Nussbaum (1986). Confucius and Mencius are brought to life by the D.C. Lau translations, Confucius (1979), and Mencius (1970). Our general guide to early Chinese thought was Graham (1989).

Regrettably, there are no comparable entry points to the Judeo-Christian and Vedic traditions, and readers, depending on taste, may have a hard time wading through thick

clouds of religious mysticism. Yet academic philosophy, in spite of a renewed tendency toward hairsplitting *philosophical* argumentation, can still throw a bright light on many subtle questions: for readers interested in the human mind, we recommend Fingarette (2000). *Dimidium facti, qui coepit, habet; sapere aude, incipe.*

Hints for the exercises

Chapter 2

1 Look for counterexamples

2a Start with the full and non-strict case

3b No

4 5 16 Consider the parentheses

18 Try different ordering relations over infinite sets such as the integers

Chapter 3

4 Dogs are animals

11 Consider Cayley's Theorem

Chapter 4

4 Not all languages have examples of all four

14 Consider which states are sinks

Chapter 5

3 Start with the smaller problem of [one ... ninety nine](#)

5 The distribution of *-er* is simpler to state as the union of two distributions, one related to verbal and the other to adjectival stems

7 Both *fast* and *acting* exist as free forms, but **fastact* in particular, and adverbial-verb compounds in general, are missing. Does this necessitate ternary branching rules?

Chapter 6

3 Use bignum-capable software such as Python or Mathematica

Chapter 7

3 People do things out of fear

10 No

Solutions for the exercises

Chapter 4

7 While the automata associated to $(aa)^*$ and $(ab)^*$ are nearly identical, the syntactic monoids are quite a bit different, beginning with the fact that $\{a, b\}^*/(aa)^*$ has only three classes while $\{a, b\}^*/(ab)^*$ has four. There are three equivalence classes in common: one that contains members of the language specified by the regular expression, which we denote by e ; one that contains a , which we denote by a ; and a sink class s . However, the letter b , which in the $(aa)^*$ case falls in the sink class, will fall in a different ‘redeemable’ equivalence class r in the $(ab)^*$ case. This is because using even a single b during the creation of a string the bounds of the $(aa)^*$ language are left, while in the $(ab)^*$ case we still can have a grammatical string as a result. We have the following multiplication tables:

	e a s
e	e e a s
a	a a e s
s	s s s s

	e a r s
e	e e a r s
a	a a s e s
r	r r s s s
s	s s s s s

Table 3.3 Multiplication in $\{a, b\}^*/(aa)^*$ and in $\{a, b\}^*/(ab)^*$

Chapter 9

2 Start with a single, essential good, say grain, and a uniform, annual mode of production that yields 105 seeds for every 100 seeds sown. Assume that individuals have free choice in saving as much as they wish for next year, subject to some minimum consumption limit, under which they’d die of hunger. Set up initial conditions in both societies so that the total availability of grain at the beginning equals to two years’ worth of consumption, but in one society the households are free to keep as much for seed as they wish, with the proceeds from later years staying with them, while in the other society whatever they don’t consume will go into a communal storage and invested equally on every household’s behalf.

References

- Ackerman, Nate (2001). “Lindstrom’s Theorem”. In: <http://bit.ly/2n5pwoS>.
- Ács, Judit, Katalin Pajkossy, and András Kornai (2013). “Building basic vocabulary across 40 languages”. In: *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 52–58.
- Adler, Mortimer (1978). *Aristotle for Everyone: Difficult Thought Made Easy*. New York: Macmillan.
- Alcorta, Candace and Richard Sosis (2007). “Rituals of Humans and Animals”. In: *Encyclopedia of Human-Animal Relationships 2*. Ed. by Marc Bekoff, pp. 599–605.
- Allen, B., Donna Gardiner, and D. Frantz (1984). “Noun Incorporation In Southern Tiwa”. In: *IJAL* 50.
- Allen, J.F. and G. Ferguson (1994). “Actions and events in interval temporal logic”. In: *Journal of logic and computation* 4.5, p. 531.
- Anderson, John M. (2005a). *The non-autonomy of syntax*. Vol. 39, pp. 223–250.
- Anderson, Stephen R. (1982). “Where Is Morphology?” In: *Linguistic Inquiry* 13, pp. 571–612.
- (2003). “Morphology”. In: *Encyclopedia of Cognitive Science*. Macmillan Publishers Ltd.
- Anderson, Stephen R (2005b). *Aspects of the Theory of Clitics*. Oxford University.
- Angluin, Dana and Carl H. Smith (1983). “Inductive Inference: Theory and Methods”. In: *ACM Computing Surveys* 15.3, pp. 237–269.
- Apley, Daniel (2003). “Principal Components and Factor Analysis”. In: *The Handbook of Data Mining*. Ed. by Nong Ye. Lawrence Erlbaum Associates.
- Apresjan, Ju D (1965). “Opyt opisanija znacenij glagolov po ix sintaksiceskim priznakam (tipam upravljenija)”. In: *Voprosy jazykoznanija* 5, pp. 51–66.
- Arnold, GM and AJ Collins (1993). “Interpretation of transformed axes in multivariate analysis”. In: *Applied statistics*, pp. 381–400.
- Aronoff, M. (1974). *Word-structure*. Ph.D. Thesis, Massachusetts Institute of Technology.
- Aronoff, Mark (1976). *Word Formation in Generative Grammar*. MIT Press.
- (1985). “Orthography and Linguistic Theory: The Syntactic Basis of Masoretic Hebrew Punctuation”. In: *Language* 61.1, pp. 28–72.
- (2007). “In the beginning was the word”. In: *Language*, pp. 803–830.
- Arora, Sanjeev et al. (2015). “Random Walks on Context Spaces: Towards an Explanation of the Mysteries of Semantic Word Embeddings”. In: *arXiv:1502.03520v1* 4, pp. 385–399.
- Asher, Nicholas and Alex Lascarides (2003). *Logics of Conversation*. Cambridge University Press.
- Ayer, Alfred Jules (1946). *Language, truth and logic*. New York: Dover Publications.
- Bach, Emmon (1981). “Discontinuous constituents in generalized categorial grammars”. In: *Proceedings of the NELS*. Vol. II, pp. 1–12.

- Bailey, Dan, Yuliya Lierler, and Benjamin Susman (2015). “Prepositional Phrase Attachment Problem Revisited: How VERBNET Can Help”. In: *Proceedings of the 11th International Conference on Computational Semantics (IWCS)*. Association for Computational Linguistics.
- Bar-Hillel, Yehoshua, Chaim Gaifman, and Eli Shamir (1960). “On categorial and phrase structure grammars”. In: *Bulletin of the Research Council of Israel* 9F, pp. 1–16.
- Baxter, Jonathan (1995a). *Learning Internal Representations*. Santa Cruz, CA: ACM Press, pp. 311–320.
- (1995b). *The canonical metric for vector quantization*. NeuroCOLT NC-TR-95-047. London: University of London.
- Bayles, M.D. (1968). *Contemporary utilitarianism*. Anchor Books.
- Belinkov, Yonatan et al. (2014). “Exploring Compositional Architectures and Word Vector Representations for Prepositional Phrase Attachment”. In: *Transactions of the ACL 2*, pp. 561–572.
- Belnap, Nuel D. (1977). “How a computer should think”. In: *Contemporary Aspects of Philosophy*. Ed. by G. Ryle. Newcastle upon Tyne: Oriel Press, pp. 30–56.
- Berge, Claude (1989). *Hypergraphs*. Vol. 45. North Holland Mathematical Library. Amsterdam: North Holland.
- Bergelson, Erika and Daniel Swingley (2013). “The acquisition of abstract words by young infants”. In: *Cognition* 127, pp. 391–397.
- Bird, S., E. Klein, and E. Loper (2009). *Natural language processing with Python*. O’Reilly Media.
- Blackburn, Patrick and Johan Bos (2005). *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI.
- Blish, James (1973). *The Quincunx of Time*. New York: Dell.
- Bloomfield, Leonard (1926). “A set of postulates for the science of language”. In: *Language* 2, pp. 153–164.
- Bobro, Marc (2013). “Leibniz on Causation”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. URL: <http://plato.stanford.edu/archives/sum2013/entries/leibniz-causation>.
- Boden, Margaret (2006). *Mind as machine*. Oxford University Press.
- Bogert, Bruce P., Michael J.R. Healy, and John W. Tukey (1963). “The quefrency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking”. In: *Proceedings of the Symposium on Time Series Analysis*. Ed. by M. Rosenblatt. Wiley, pp. 209–243.
- Boguraev, Branimir K. and Edward J. Briscoe (1989). *Computational Lexicography for Natural Language Processing*. Longman.
- Bolinger, Dwight (1965). “Pitch accents and sentence rhythm”. In: *Forms of English: accent, morpheme, order*. Cambridge MA: Harvard University Press.
- Bolinger, Dwight L. (1962). “Binomials and pitch accent”. In: *Lingua* 11, pp. 34–44.
- Borer, Hagit (2005). *Structuring sense*. Vol. I and II. Oxford University Press.
- (2013). *Structuring sense*. Vol. III. Oxford University Press.

- Böttner, Michael (2001). “Peirce Grammar”. In: *Grammars* 4.1, pp. 1–19.
- Brachman, R.J. and H. Levesque (1985). *Readings in knowledge representation*. Morgan Kaufman Publishers Inc., Los Altos, CA.
- Brown, P. et al. (1992). “An Estimate of an Upper Bound for the Entropy of English”. In: *Computational Linguistics* 18/1, pp. 31–40.
- Brzozowski, J.A. (1962). “Canonical regular expressions and minimal state graphs for definite events”. In: *Mathematical theory of Automata*. Polytechnic Press, Polytechnic Institute of Brooklyn, N.Y., pp. 529–561.
- Burris, Stanley (2001). “Downward Löwenheim-Skolem theorem”. In: <http://www.math.uwaterloo.ca/~snburris/htdocs/WWW/PDF/downward.pdf>.
- Butt, Miriam (2006). *Theories of Case*. Cambridge University Press.
- Camerer, Colin (2003). *Behavioral game theory*. Princeton University Press.
- Carlson, Greg and Francis J. Pelletier, eds. (1995). *The Generic Book*. University of Chicago Press.
- Carnap, Rudolf (1946). “Modalities and quantification”. In: *The Journal of Symbolic Logic* 11.2, pp. 33–64.
- (1947). *Meaning and necessity*.
- Carroll, Lewis (1896). *Symbolic logic, Part I: Elementary*. London: Macmillan.
- Cawdrey, Robert (1604). *A table alphabetical of hard usual English words*.
- Chen, Danqi and Christopher D Manning (2014). “A Fast and Accurate Dependency Parser using Neural Networks.” In: *EMNLP*, pp. 740–750.
- Chen, Xinxiong, Zhiyuan Liu, and Maosong Sun (2014). “A unified model for word sense representation and disambiguation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1025–1035.
- Chiarcos, Christian and Tomaz Erjavec (2011). “OWL/DL formalization of the MULTEXT-East morphosyntactic specifications.” In: *Linguistic Annotation Workshop*, pp. 11–20.
- Chisholm, R. (1963). “Supererogation and Offence: A Conceptual Scheme for Ethics”. In: *Ratio* 5, pp. 1–14.
- Chomsky, Noam (1956). “Three models for the description of language”. In: *IRE Transactions on Information Theory* 2, pp. 113–124.
- (1957). *Syntactic Structures*. The Hague: Mouton.
- (1965). *Aspects of the Theory of Syntax*. MIT Press.
- (1970). “Remarks on nominalization”. In: *Readings in English Transformational Grammar*. Ed. by R. Jacobs and P. Rosenbaum. Waltham, MA: Blaisdell, pp. 184–221.
- (1973). “Conditions on Transformations”. In: *A festschrift for Morris Halle*. Ed. by S.R. Anderson and P. Kiparsky. New York: Holt, Rinehart and Winston.
- Chomsky, Noam and Howard Lasnik (1993). “Principles and Parameters Theory”. In: *Syntax: An International Handbook of Contemporary Research*. Ed. by J. Jacobs. Vol. 1. Berlin: de Gruyter, pp. 505–569.

- Christodoulopoulos, Christos, Sharon Goldwater, and Mark Steedman (2010). “Two Decades of Unsupervised POS induction: How far have we come?” In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 575–584.
- Church, Kenneth W. and Patrick Hanks (1990). “Word association norms, mutual information, and lexicography”. In: *Computational Linguistics* 16.1, pp. 22–29.
- Churchman, C. West (1971). *The design of inquiring systems*. New York: Basic Books.
- Clark, Stephen (2015). “Vector space models of lexical meaning”. In: *Handbook of Contemporary Semantics*. Ed. by Shalom Lappin and Chris Fox. 2nd. Blackwell, pp. 493–522.
- Cohen-Sygal, Yael and Shuly Wintner (2006). “Finite-state registered automata for non-concatenative morphology”. In: *Computational Linguistics* 32.1, pp. 49–82.
- Collobert, R. et al. (2011). “Natural Language Processing (Almost) from Scratch”. In: *Journal of Machine Learning Research (JMLR)*.
- Confucius (1979). *The Analects*. Trans. by D.C. Lau. Harmondsworth: Penguin.
- Courtney, Rosemary (1983). *Longman Dictionary of Phrasal Verbs*. Longman.
- Covington, Michael A. (1984). *Syntactic Theory in the High Middle Ages*. Cambridge University Press.
- Croft, William (2000). “Parts of speech as language universals and as language-particular categories”. In: *Approaches to the typology of word classes*. Ed. by Petra Vogel and Bernard Comrie. Mouton de Gruyter, pp. 65–102.
- Curry, Haskell B. (1961). “Some logical aspects of grammatical structure”. In: *Structure of Language and its Mathematical Aspects*. Ed. by R. Jakobson. Providence, RI: American Mathematical Society, pp. 56–68.
- Dalrymple, Mary (1990). *Syntactic constraints on anaphoric binding*. Stanford University.
- Davidson, Donald (1990). “Turing’s test”. In: *Modelling the mind*. Ed. by K. A. Mohyeldin Said et al. Clarendon Press, pp. 1–11.
- Davis, Steven B. and Paul Mermelstein (1980). “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4, pp. 357–366.
- Deerwester, Scott C., Susan T Dumais, and Richard A. Harshman (1990). “Indexing by latent semantic analysis”. In: *Journal of the American Society for Information Science* 41.6, pp. 391–407.
- Dick, Philip K. (1981). *The divine invasion*. Simon and Schuster.
- Dixon, Robert M.W. (2009). *Basic Linguistic Theory (in 3 volumes)*. Oxford University Press.
- Dowty, David, Robert Wall, and Stanley Peters (1981). *Introduction to Montague Semantics*. Dordrecht: Reidel.
- Drewes, Frank, Hans-Jörg Kreowski, and Annegret Habel (1997). “Hyperedge replacement graph grammars”. In: *Handbook of Graph Grammars and Computing by Graph Transformation*. Ed. by Grzegorz Rozenberg. World Scientific, pp. 95–162.

- Driver, Julia (2009). “The History of Utilitarianism”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2009. <http://plato.stanford.edu/archives/sum2009/entries/utilitarianism-history/>.
- Dummit, David Steven and Richard M Foote (2003). *Abstract algebra - 3rd edition*. Wiley.
- Eco, Umberto (1995). *The Search for the Perfect Language*. Oxford: Blackwell.
- Eijck, Jan van and Christina Unger (2010). *Computational Semantics with Functional Programming*. Cambridge University Press.
- Eilenberg, Samuel (1974). *Automata, Languages, and Machines*. Vol. A. Academic Press.
- Evans, James and Brian Popp (1985). “Pictet’s experiment: the apparent radiation and reflection of cold”. In: *American Journal of Physics* 53 (8), pp. 737–753.
- Fabisch, Alexander (2011). “Two Spirals Problem Solved in Compressed Weight Space”. URL: <https://www.youtube.com/watch?v=MkLJ-9MubKQ>.
- Feynman, Richard Phillips (1965). *The character of physical law*. Vol. 66. The MIT Press paperback series. MIT press.
- Fillmore, Charles (1977). “The case for case reopened”. In: *Grammatical Relations*. Ed. by P. Cole and J.M. Sadock. Academic Press, pp. 59–82.
- Fillmore, Charles and Paul Kay (1997). *Berkeley Construction Grammar*. URL: <http://www.icsi.berkeley.edu/%5C~%7B%7Dkay/bcg/ConGram.html>.
- Fillmore, C.J. and B.T.S. Atkins (1994). “Starting where the dictionaries stop: The challenge of corpus lexicography”. In: *Computational approaches to the lexicon*, pp. 349–393.
- Findler, Nicholas V., ed. (1979). *Associative Networks: Representation and Use of Knowledge by Computers*. Academic Press.
- Fine, Kit (1985). *Reasoning with Arbitrary Objects*. Oxford: Blackwell.
- (2012). “Aristotle’s Megarian Manoeuvres”. In: *Mind* 120 (480), pp. 993–1034.
- Fingarette, H. (2000). *Self-deception*. University of California Press.
- Flickinger, Daniel P. (1987). *Lexical Rules in the Hierarchical Lexicon*. Stanford University: PhD Thesis.
- Floyd, Robert W (1967). “Nondeterministic algorithms”. In: *Journal of the ACM (JACM)* 14.4, pp. 636–644.
- Foley, William A. and Robert van Valin (1984). *Functional Syntax and Universal Grammar*. Cambridge University Press.
- Frank, Robert and Giorgio Satta (1998). “Optimality theory and the generative complexity of constraint violability”. In: *Computational Linguistics* 24.2, pp. 307–315.
- Fraser, Chris (2014). “Mohism”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2014. <http://plato.stanford.edu/archives/spr2014/entries/mohism>.
- Frege, Gottlob (1879). *Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle: L. Nebert.

- (1884). *Die Grundlagen der Arithmetik: eine logisch-mathematische Untersuchung ueber den Begriff der Zahl*. W. Koebner.
- (1892). “On sense and reference”. In: *The Philosophy of Language*. Ed. by A.P. Martinich. New York: Oxford University Press (4th ed, 2000), pp. 36–56.
- Fries, Charles C. (1952). *The Structure of English*.
- Fromkin, Victoria, Robert Rodman, and Nina Hyams (2003). *An introduction to language*. Wadsworth; Tenth Edition.
- Furui, Sadaoki (1986). “Speaker-independent isolated word recognition using dynamic features of speech spectrum”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34.1, pp. 52–59.
- Gallin, D. (1975). *Intensional and Higher-Order Modal Logic*. North-Holland.
- Gamut, L.T.F. (1991). *Logic, Language, and Meaning*. University of Chicago Press.
- Gärdenfors, Peter (2000). *Conceptual Spaces: The Geometry of Thought*. MIT Press.
- Gazdar, Gerald (1979). *Pragmatics: Implicature, presupposition, and logical form*. Academic Press New York.
- Gersho, Allen and Robert M. Gray (1992). *Vector Quantization and Signal Compression*. Springer.
- Gewirth, A. (1978). *Reason and morality*. University of Chicago Press.
- Ghallab, Malik, Dana Nau, and Paolo Traverso (2004). *Automated Planning: Theory and Practice*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1558608567.
- Givant, Steven R. (2006). “The calculus of relations as a foundation for mathematics”. In: *Journal of Automated Reasoning* 37, pp. 277–322.
- Gleason, Henry A. (1955). *An Introduction to Descriptive Linguistics*. New York: Holt.
- Goddard, Cliff (2002). “The search for the shared semantic core of all languages”. In: *Meaning and Universal Grammar – Theory and Empirical Findings*. Ed. by Cliff Goddard and Anna Wierzbicka. Vol. 1. Benjamins, pp. 5–40.
- Gödel, Kurt (1986). *Collected Works: Publications 1929-1936*. Ed. by Solomon Feferman. Clarendon Press.
- Gold, E. Mark (1967). “Language identification in the limit”. In: *Information and Control* 10, pp. 447–474.
- Goldberg, Adele E., ed. (1995). *Conceptual Structure, Discourse, and Language*. University of Chicago Press.
- Goodman, Nelson (1955). “Fact, Fiction and Forecast, Bobbs-Merrill”. In: *Indianapolis, IN*.
- Gordon, Andrew and Jerry Hobbs (2017). *A Formal Theory of Commonsense Psychology: How People Think People Think*. Cambridge University Press.
- Gould, Stephen Jay (1981). *The Mismeasure of Man*. New York: Norton.
- Graham, AC (1958). *Two Chinese Philosophers*. Lund Humphries.
- (1989). *Disputers of the Tao*. Open Court.

- Green, Georgia (1973). “On *too* and *either*, and not just *too* and *either*, *either*”. In: *Papers from the Fourth Regional Meeting of the Chicago Linguistic Society*. Chicago Linguistic Society, pp. 22–39.
- Grice, P. (1981). “Presupposition and Conversational Implicature”. In: *Radical pragmatics*, p. 183.
- Grice, Paul and Peter Strawson (1956). “In defense of a dogma”. In: *The Philosophical Review* 65, pp. 148–152.
- Groenendijk, J. and M. Stokhof (1991). “Dynamic predicate logic”. In: *Linguistics and philosophy* 14.1, pp. 39–100. ISSN: 0165-0157.
- Halmos, Paul R. (1974). *Naive Set Theory*. Undergraduate Texts in Mathematics. Springer.
- (2013). *Finite Dimensional Vector Spaces*. Springer.
- Hansen, L.K., C. Liisberg, and P. Salamon (1997). “The Error-Reject Tradeoff”. In: *Open Systems and Information Dynamics* 4, pp. 159–184.
- Harries-DeLisle, Helga (1978). “Coordination reduction”. In: *Universals of Human Language* IV. Ed. by Greenberg, pp. 515–584.
- Harris, R. (1980). *The language-makers*. Duckworth.
- (1981). *The language myth*. Duckworth.
- (1987). *The language machine*. Duckworth.
- Harris, Zellig (1946). “From morpheme to utterance”. In: *Language* 22, pp. 161–183.
- (1951). *Methods in Structural Linguistics*. University of Chicago Press.
- (1957). “Cooccurrence and transformation in linguistic structure”. In: *Language* 33, pp. 283–340.
- Hart, D. and B. Goertzel (2008). “OpenCog: A Software Framework for Integrative Artificial General Intelligence”. In: *Proceedings of the First AGI Conference*.
- Hauenschild, Ch., E. Huckert, and R. Maier (1979). “SALAT: Machine Translation Via Semantic Representation”. In: ed. by R. Baauerle, U.Egle, and A. von Stechow, pp. 324–352.
- Hayes, Bruce (1987). *Compensatory lengthening in moraic phonology*.
- Hayes, Patrick (1976). *A process to implement some word-sense disambiguations*. Geneva: Institut Dalle Molle.
- Hayes, Patrick J. (1979). “The naive physics manifesto”. In: *Expert Systems in the Micro-Electronic Age*. Ed. by D. Michie. Edinburgh University Press, pp. 242–270.
- (1995). “Computation & Intelligence”. In: ed. by George F. Luger. Menlo Park, CA: American Association for Artificial Intelligence. Chap. The Second Naive Physics Manifesto, pp. 567–585. ISBN: 0-262-62101-0.
- Heims, Steve J. (1991). *The Cybernetics Group, 1946–1953*. MIT Press.
- Herskovits, A. (1986). *Language and Spatial Cognition – An Interdisciplinary Study of the Prepositions in English*. Cambridge University Press.
- Heyd, David (2012). “Supererogation”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2012. <http://plato.stanford.edu/archives/win2012/entries/supererogation/>.

- Hindle, Donald and Mats Rooth (1993). "Structural Ambiguity and Lexical Relations". In: *Computational Linguistics* 19.1, pp. 103–120.
- Hirst, D.J. and A. Di Cristo, eds. (1998). *Intonation Systems: A survey of Twenty Languages*. Cambridge University Press.
- Hirst, Graeme (1981). *Anaphora in natural language understanding: A survey*. Springer.
- Hobbs, Jerry R. and Stanley J. Rosenschein (1978). "Making Computational Sense of Montague's Intensional Logic". In: *Artificial Intelligence* 9, pp. 287–306.
- Hobbs, J.R. (2008). "Deep Lexical Semantics". In: *Lecture Notes in Computer Science* 4919, p. 183.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780.
- Hock, HS et al. (2005). "Dynamical vs. judgmental comparison: hysteresis effects in motion perception". In: *Spatial Vision* 18.3, pp. 317–335.
- Hockett, Charles (1954). "Two models of grammatical description". In: *Word* 10, pp. 210–231.
- Hoffart, Johannes et al. (2013). "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia". In: *Artificial Intelligence* 194, pp. 28–61.
- Hopcroft, J. E. (1971). "An $n \log n$ algorithm for minimizing the states in a finite automaton". In: *The Theory of Machines and Computations*. Ed. by Z. Kohavi. Academic Press, pp. 189–196.
- Hughes, G.E. and Max J. Cresswell (1984). *A companion to Modal Logic*. Methuen.
- (1996). *A new introduction to modal logic*. Routledge.
- Hume, David (1740). *A Treatise of Human Nature*. Vol. 3. <http://www.earlymoderntexts.com/assets/pdfs/hume1740book3.pdf>.
- Jackendoff, Ray S. (1972). *Semantic Interpretation in Generative Grammar*. MIT Press.
- (1977). *X-bar Syntax: A Study of Phrase Structure*. MIT Press.
- (1990). *Semantic Structures*. MIT Press.
- (2008). "Construction after construction and its theoretical challenges". In: *Language* 84, pp. 8–28.
- Jackendoff, Roman (1969). "An Interpretive Theory of Negation". In: *Foundations of Language* 5, pp. 218–241.
- Jacobson, Pauline (2014). *Compositional Semantics*. Oxford University Press.
- Jakobson, Roman (1936). *Beitrag zur allgemeinen Kasuslehre*. Travaux du Cercle linguistique de Prague.
- (1984). "Contribution to the General Theory of Case: General Meanings of the Russian Cases". In: *Roman Jakobson. Russian and Slavic Grammar: Studies 1931-1981*. Ed. by Linda R. Waughandmorris Halle. Berlin: Mouton de Gruyter, pp. 59–103.
- Janssen, T.M.V. (2001). "Frege, contextuality and compositionality". In: *Journal of Logic, Language and Information* 10.1, pp. 115–136. ISSN: 0925-8531.
- Johnson, Ch. Douglas (1970). *Formal aspects of phonological representation*. UC Berkeley: PhD thesis.

- Joyce, Richard (2009). “Moral Anti-Realism”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2009. <http://plato.stanford.edu/archives/sum2009/entries/moral-anti-realism>.
- Judson, Thomas W and Thomas W Judson (2009). *Abstract algebra: theory and applications*. Virginia Commonwealth University Mathematics.
- Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing*. 2nd edition. Pearson.
- Kahneman, Daniel and Amos Tversky (1979). “Prospect Theory: An Analysis of Decision under Risk”. In: *Econometrica* 47.2, pp. 263–291.
- Kálmán, László and András Kornai (1985). *Pattern matching: a finite state approach to generation and parsing*.
- Kann, C. and R. Kirchoff, eds. (2012). *William of Sherwood. Syncategoremata*. Meiner.
- Kant, I. (1960). “Religion within the Limits of Reason Alone (1793)”. In: ed. by TM Greene and HH Hudson.
- Kaplan, Ronald M. and Martin Kay (1994). “Regular Models of Phonological Rule Systems”. In: *Computational Linguistics* 20.3, pp. 331–378.
- Karpathy, Andrej, Armand Joulin, and Fei Fei F Li (2014). “Deep Fragment Embeddings for Bidirectional Image Sentence Mapping”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., pp. 1889–1897.
- Karttunen, L. and K.R. Beesley (2003). “Finite State Morphology”. In: *CSLI Studies in Computational Linguistics. CSLI Publication, Stanford*.
- Karttunen, L. and S. Peters (1979). “Conventional implicature”. In: *Syntax and semantics* 11, pp. 1–56.
- Karttunen, Lauri (1989). “Radical lexicalism”. In: *Alternative Conceptions of Phrase Structure*. Ed. by Mark Baltin and Anthony Kroch. University of Chicago Press, pp. 43–65.
- (1998). “The proper treatment of optimality in computational phonology: plenary talk”. In: *Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1–12.
- Katz, J. and Jerry A. Fodor (1963). “The structure of a semantic theory”. In: *Language* 39, pp. 170–210.
- Kauppinen, Antti (2014). “Moral Sentimentalism”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2014.
- Kay, Paul (2002). “An informal sketch of a formal architecture for construction grammar”. In: *Grammars* 5, pp. 1–19.
- Keenan, E.L. and B. Comrie (1977). “Noun phrase accessibility and universal grammar”. In: *Linguistic inquiry* 8.1, pp. 63–99.
- Kiparsky, Paul (1982). “From cyclic phonology to lexical phonology”. In: *The structure of phonological representations, I*. Ed. by H. van der Hulst and N. Smith. Dordrecht: Foris, pp. 131–175.

- (1998). “Aspect and Event Structure in Vedic”. In: *Yearbook of South Asian Languages and Linguistics*. Ed. by Rajendra Singh. Sage Publications, pp. 29–61.
- Kipper, Karin et al. (2008). “A large-scale classification of English verbs”. In: *Language Resources and Evaluation* 42.1, pp. 21–40.
- Kirsner, R.S. (1993). “From meaning to message in two theories: Cognitive and Saussurean views of the Modern Dutch demonstratives”. In: *Conceptualizations and mental processing in language*. Ed. by Richard A Geiger and Brygida Rudzka-Ostyn, pp. 80–114.
- Kleene, Stephen C. (1956). “Representation of events in nerve nets and finite automata”. In: *Automata Studies*. Ed. by C. Shannon and J. McCarthy. Princeton University Press, pp. 3–41.
- (2002). *Mathematical Logic*. Dover.
- Klima, Gyula (2009). *John Buridan*. Oxford University Press.
- Knuth, Donald E. (1969). *The Art of Computer Programming. Vol. II: Seminumerical Algorithms*. Addison-Wesley.
- (1971). *The Art of Computer Programming*. Addison-Wesley.
- Kolmogorov, Andrei N. (1933). *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer.
- (1953). “O ponyatii algoritma”. In: *Uspehi matematicheskikh nauk* 8.4, pp. 175–176.
- Kornai, András (2002). “How many words are there?” In: *Glottometrics* 2.4, pp. 61–86.
- (2007). *Mathematical linguistics*. Springer.
- (2008). *Mathematical Linguistics*. Advanced Information and Knowledge Processing. Springer. ISBN: 9781846289859.
- (2009). “The complexity of phonology”. In: *Linguistic Inquiry* 40.4, pp. 701–712.
- (2010a). “The algebra of lexical semantics”. In: *Proceedings of the 11th Mathematics of Language Workshop*. Ed. by Christian Ebert, Gerhard Jäger, and Jens Michaelis. LNAI 6149. Springer, pp. 174–199.
- (2010b). “The treatment of ordinary quantification in English proper”. In: *Hungarian Review of Philosophy* 54.4, pp. 150–162.
- (2012). “Eliminating ditransitives”. In: *Revised and Selected Papers from the 15th and 16th Formal Grammar Conferences*. Ed. by Ph. de Groote and M-J Nederhof. LNCS 7395. Springer, pp. 243–261.
- (2014a). “Bounding the impact of AGI”. In: *Journal of Experimental and Theoretical Artificial Intelligence* 26.3, pp. 417–438.
- (2014b). “Euclidean Automata”. In: *Implementing Selves with Safe Motivational Systems and Self-Improvement*. Ed. by Mark Waser. Proc. AAAI Spring Symposium. AAAI Press, pp. 25–30.
- (2014c). “Finite automata with continuous input”. In: *Short Papers from the Sixth Workshop on Non-Classical Models of Automata and Applications*. Ed. by S. Bensch, R. Freund, and F. Otto.
- (2015). “Realizing monads”. In: *Hungarian Review of Philosophy* 59.2, pp. 153–162.
- (2018). “Truth or dare”. In: *Karttunen Festschrift*. Ed. by Cleo Condoravdi. URL: <http://kornai.com/Drafts/dare.pdf>.

- Kracht, Marcus (2003). *The Mathematics of Language*. Berlin: Mouton de Gruyter.
- Kripke, Saul A (1959). “Semantical Analysis of Modal Logic”. In: *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 24.4, pp. 323–324.
- (1963). “Semantical Analysis of Modal Logic I: Normal Modal Propositional Calculi”. In: *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 9, pp. 67–96.
- Kuich, W. and A. Salomaa (1985). *Semirings, Automata and Languages*. Springer-Verlag New York, Inc. Secaucus, NJ, USA. ISBN: 0387137165.
- Kurzweil, Ray (2012). *How to create a mind*. Viking Press.
- Ladusaw, William A. (1980). *Polarity Sensitivity as Inherent Scope Relations*. New York: Garland Press.
- Lakoff, George (1987). *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press. ISBN: 978-0-226-46803-7.
- Lambalgen, Michiel van and Fritz Hamm (2005). *The proper treatment of events*. Oxford: Blackwell.
- Landis, D. and T. Saral (1978). “Atlas of American Black English”. In: *Project on Cross-Cultural Affective Meanings: Tables for 17 cultures*. Ed. by Charles Osgood. Urbana, IL.
- Landsbergen, Jan (1982). “Machine translation based on logically isomorphic Montague grammars”. In: *Proceedings of the 9th conference on Computational linguistics-Volume I*. Academia Praha, pp. 175–181.
- Langacker, Ronald (1987). *Foundations of Cognitive Grammar*. Vol. 1. Stanford University Press.
- Lapierre, Serge (1994). “Montague-Gallin’s Intensional Logic, Structured Meanings and Scott’s Domains”. In: *Logic and Philosophy of Science in Uppsala*. Ed. by Dag Westerstål Dag Prawitz. Vol. 236. Synthese Library. Reidel, pp. 29–48.
- Lemmon, Edward John, Dana S Scott, and Krister Segerberg (1977). *An Introduction to Modal Logic: The “Lemmon Notes”*. Blackwell.
- Lenat, Douglas B. and R.V. Guha (1990). *Building Large Knowledge-Based Systems*. Addison-Wesley.
- Levesque, Hector, Ernest Davis, and Leora Morgenstein (2012). “The Winograd Schema Challenge”. In: *Proc. 13th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 8–15.
- Levin, Beth (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.
- Levinson, Stephen C. (1983). *Pragmatics*.
- Levy, Omer and Yoav Goldberg (2014). “Neural Word Embedding as Implicit Matrix Factorization”. In: *Advances in Neural Information Processing Systems* 27. Ed. by Z. Ghahramani et al., pp. 2177–2185.
- Levy, Omer, Yoav Goldberg, and Ido Dagan (2015). “Improving Distributional Similarity with Lessons Learned from Word Embeddings”. In: *Transactions of the Association for Computational Linguistics* 3, pp. 211–225.

- Lewis, D. (1970). “General semantics”. In: *Synthese* 22.1, pp. 18–67.
- Locke, John, Ed. with critical apparatus, and Peter H Nidditch (1970). *An essay concerning human understanding, 1690*. Scolar Press.
- Lyons, John (1995). *Linguistic semantics: An introduction*. Cambridge University Press.
- Mackie, J.L. (1977). *Ethics: Inventing Right and Wrong*. Penguin.
- Makrai, Márton (in preparation). *Inter-lingual meaning representations*. PhD thesis, Loránd Eötvös University.
- Manning, Christopher D and Bill MacCartney (2009). “An extended model of natural logic”. In: *Proceedings of the 8th International Conference on Computational Semantics*, pp. 140–156.
- Marcus, Mitchell P. (1980). *A theory of syntactic recognition for natural language*. MIT Press.
- Marcus, Mitchell, Beatrice Santorini, and Mary Ann Marcinkiewicz (1993). “Building a Large Annotated Corpus of English: The Penn Treebank”. In: *Computational Linguistics* 19, pp. 313–330.
- Matthews, P. H. (1991). *Morphology, 2nd edition*. Cambridge University Press.
- McCarthy, John (1979 (1990)). “Ascribing mental qualities to machines”. In: *Formalizing common sense*. Ed. by V. Lifschitz. Ablex, pp. 93–118.
- (1976). “An example for natural language understanding and the AI problems it raises”. In: *Formalizing Common Sense: Papers by John McCarthy*. Ablex Publishing Corporation 355.
- (1979). “First order theories of individual concepts and propositions”. In: *Machine Intelligence* 9.
- McCawley, James D. (1970). “Where do noun phrases come from?” In: *Semantics*. Ed. by D. Steinberg and L. Jakobovits. Cambridge University Press, pp. 217–231.
- McCulloch, W.S. (1945). “A heterarchy of values determined by the topology of nervous nets”. In: *Bulletin of Mathematical Biophysics* 7, pp. 89–93.
- McCulloch, W.S. and W. Pitts (1943). “A logical calculus of the ideas immanent in nervous activity”. In: *Bulletin of mathematical biophysics* 5, pp. 115–133.
- McKeown, Margaret G. and Mary E. Curtis (1987). *The nature of vocabulary acquisition*. Lawrence Erlbaum Associates.
- Mencius (1970). Trans. by D.C. Lau. Harmondsworth: Penguin.
- Merchant, Jason (2001). *The Syntax of Silence: Sluicing, Islands, and the Theory of Ellipsis*. Oxford University Press.
- Mikolov, Tomas, Quoc V Le, and Ilya Sutskever (2013). “Exploiting similarities among languages for machine translation”. Xiv preprint arXiv:1309.4168.
- Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig (2013). “Linguistic Regularities in Continuous Space Word Representations”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*. Atlanta, Georgia: Association for Computational Linguistics, pp. 746–751.

- Mikolov, Tomas et al. (2013). “Efficient Estimation of Word Representations in Vector Space”. In: *Proceedings of the ICLR 2013*. Ed. by Y. Bengio and Y. LeCun.
- Miller, George A. (1995). “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11, pp. 39–41.
- Miller, George A. and Noam Chomsky (1963). “Finitary models of language users”. In: *Handbook of Mathematical Psychology*. Ed. by R.D. Luce, R.R. Bush, and E. Galanter. Wiley, pp. 419–491.
- Miller, Philip (1986). “String analysis”. In: *Prague Bulletin of Mathematical Linguistics*. — (1987). “String analysis II”. In: *Prague Bulletin of Mathematical Linguistics*.
- Minsky, Marvin (1986). *The Society of Mind*. Simon and Schuster.
- Mitchell, Tom M. (1997). *Machine learning*. McGraw Hill.
- Mitzenmacher, Michael (2004). “A Brief History of Generative Models for Power Law and Lognormal Distributions”. In: *Internet Mathematics* 1.2, pp. 226–251.
- Modrak, Deborah K. W. (2009). *Aristotle’s Theory of Language and Meaning*. Cambridge University Press.
- Montague, Richard (1970). “Universal Grammar”. In: *Theoria* 36, pp. 373–398.
- (1973). “The proper treatment of quantification in ordinary English”. In: *Formal Philosophy*. Ed. by R. Thomason. Yale University Press, pp. 247–270.
- Moore, G.E. (1903). *Principia ethica*. Cambridge University Press.
- Morrill, Glynn (2011). “CatLog: A Categorical Parser/Theorem-Prover”. In: *Type Dependency, Type Theory with Records, and Natural-Language Flexibility*.
- Nalisnick, Eric T. and Sachin Ravi (2015). “Infinite dimensional word embeddings”. In: *arXiv preprint arXiv:1511.05392v2*.
- Nelson, R.J. (1982). *The logic of mind*. Dordrecht: Reidel.
- Nemeskey, Dávid et al. (2013). “Spreading activation in language understanding”. In: *Proceedings of the 9th International Conference on Computer Science and Information Technologies (CSIT 2013)*. Yerevan, Armenia: Springer, pp. 140–143.
- Ng, Andrew Y., Michael I. Jordan, and Yair Weiss (2001). “On Spectral Clustering: Analysis and an algorithm”. In: *Advances in neural information processing systems*. MIT Press, pp. 849–856.
- Nivre, Joakim et al. (2016). “Universal Dependencies v1: A Multilingual Treebank Collection”. In: *Proc. LREC 2016*, pp. 1659–1666.
- Nussbaum, Martha C. (1986). *The fragility of goodness*. Cambridge University Press.
- Ogden, C.K. (1944). *Basic English: a general introduction with rules and grammar*. K. Paul, Trench, Trubner.
- Osgood, Charles E., William S. May, and Murray S. Miron (1975). *Cross Cultural Universals of Affective Meaning*. University of Illinois Press.
- Ostler, Nicholas (1979). *Case-Linking: a Theory of Case and Verb Diathesis Applied to Classical Sanskrit*. MIT: PhD thesis.
- Parsons, Terence (1970). “Some problems concerning the logic of grammatical modifiers”. In: *Synthese* 21.3–4, pp. 320–334.

- (1974). “A Prolegomenon to Meinongian Semantics”. In: *The Journal of Philosophy* 71.16, pp. 561–580.
- Partee, Barbara (1980). “Montague grammar, mental representation, and reality”. In: *Philosophy and Grammar*. Ed. by S. Ohman and S. Kanger. Dordrecht: D. Reidel, pp. 59–78.
- (1984). “Nominal and temporal anaphora”. In: *Linguistics and Philosophy* 7, pp. 243–286.
- Pearl, Judea (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.
- Pearson, K. (1901). “LIII. On lines and planes of closest fit to systems of points in space”. In: *Philosophical Magazine Series 6* 2.11, pp. 559–572.
- Pearson, Karl (1894). “Contributions to the Mathematical Theory of Evolution”. In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 185, pp. 71–110. ISSN: 0264-3820. DOI: [10.1098/rsta.1894.0003](https://doi.org/10.1098/rsta.1894.0003). URL: <http://rsta.royalsocietypublishing.org/content/185/71>.
- Penrose, Roger (1989). *The emperor’s new mind: concerning computers, minds, and the laws of physics*. Oxford University Press.
- Perlmutter, David M. (1983). *Studies in Relational Grammar*. University of Chicago Press.
- Piattelli-Palmarini, M., J. Piaget, and N. Chomsky (1980). *Language and learning: the debate between Jean Piaget and Noam Chomsky*. Routledge. ISBN: 0710004389.
- Pin, Jean-Eric (1997). “Syntactic semigroups”. In: *Handbook of Formal Language Theory*. Ed. by G. Rozenberg and A. Salomaa. Vol. 1. Springer, pp. 679–746.
- Pinker, S. and A. Prince (1994). “Regular and irregular morphology and the psychological status of rules of grammar”. In: *The reality of linguistic rules* 32151.
- Pinker, Steven (1994). *The language instinct*. William Morrow and Co.
- Plank, Frans, ed. (1984). *Objects*. London: Academic Press.
- Plate, Tony A (1995). “Holographic reduced representations”. In: *Neural networks, IEEE transactions on* 6.3, pp. 623–641.
- Plotinus (0250). “Enneads”. In: <http://classics.mit.edu/Plotinus/enneads.5.fifth.html>.
- Pollack, Jordan B. (1990). “Recursive Distributed Representations”. In: *Artificial Intelligence* 46.1, pp. 77–105.
- Pollard, Carl (2008). “Hyperintensions”. In: *Journal of Logic and Computation* 18.2, pp. 257–282.
- Poltoratski, Sonia and Frank Tong (2013). “Hysteresis in the Perception of Objects and Scenes”. In: *Journal of Vision* 13.9, p. 672.
- Postal, Paul M. (1969). “Anaphoric Islands”. In: Papers from the fifth regional meeting of the Chicago Linguistic Society, pp. 205–239.
- Potts, C. (2005). *The logic of conventional implicatures*. Oxford University Press, USA.

- Priest, Graham (1979). "The Logic of Paradox". In: *Journal of Philosophical Logic* 8, pp. 219–241.
- Priest, Graham, Richard Routley, and J. Norman (1989). *Paraconsistent Logic: Essays on the Inconsistent*. Munich: Philosophia-Verlag.
- Pullum, Geoffrey K. (1989). "Formal linguistics meets the Boojum". In: *Natural Language and Linguistic Theory* 7.1, pp. 137–143.
- Pustejovsky, James (1995). *The Generative Lexicon*. MIT Press.
- Putnam, H. (1976). "Two Dogmas Revisited". In: *printed in his (1983) Realism and Reason, Philosophical Papers* 3.
- Quillian, M. Ross (1967). "Semantic memory". In: *Semantic information processing*. Ed. by Minsky. Cambridge: MIT Press, pp. 227–270.
- (1968). "Word concepts: A theory and simulation of some basic semantic capabilities". In: *Behavioral Science* 12, pp. 410–430.
- (1969). "The teachable language comprehender". In: *Communications of the ACM* 12, pp. 459–476.
- Quine, Willard van Orman (1951). "Two dogmas of empiricism". In: *The Philosophical Review* 60, pp. 20–43.
- Rabin, M.O. and D. Scott (1959). "Finite automata and their decision problems". In: *IBM journal of research and development* 3.2, pp. 114–125. ISSN: 0018-8646.
- Reisinger, Joseph and Raymond J Mooney (2010). "Multi-prototype vector-space models of word meaning". In: *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 109–117.
- Restall, Greg (2007). *Modal models for Bradwardine's truth*. University of Melbourne: ms.
- Reynolds, C. J. (2005). "On the computational complexity of action evaluations". In: *Presented at Computer Ethics: Philosophical Enquiry*. <http://affect.media.mit.edu/pdfs/05.reynolds-cepe.pdf>.
- Rosch, Eleanor (1975). "Cognitive Representations of Semantic Categories". In: *Journal of Experimental Psychology* 104.3, pp. 192–233.
- Ruhl, C. (1989). *On monosemy: a study in linguistic semantics*. State University of New York Press.
- Russell, Bertrand (1905). "On denoting". In: *Mind* 14, pp. 441–478.
- Ryle, Gilbert (1949). *The concept of mind*. University of Chicago Press, p. 334.
- Sadock, Jerrold M. (1999). "The Nominalist Theory of Eskimo: A Case Study in Scientific Self-Deception". In: *International Journal of American Linguistics* 65.4, pp. 383–406.
- Sainsbury, M. and T. Williamson (1995). "Sorites". In: *Blackwell Companion to the Philosophy of Language*. Ed. by B. Hale and C. Wright. Blackwell.
- Sainsbury, Mark (1992). "Sorites Paradoxes and the Transition Question". In: *Philosophical Papers* 21.3, pp. 77–190.

- Salzmann, Martin David (2004). “Theoretical Approaches to Locative Inversion”. PhD thesis. MA Thesis, University of Zurich.
- Saussure, Ferdinand de (1966). *Course in General Linguistics*. McGraw-Hill.
- Scanlon, J. (1988). “Husserl’s Ideas and the Natural Concept of the World”. In: *Edmund Husserl and the Phenomenological Tradition*, pp. 217–233.
- Schank, Roger C. (1972). “Conceptual dependency: A theory of natural language understanding”. In: *Cognitive Psychology* 3.4, pp. 552–631.
- Schmitt, Otto H (1938). “A thermionic trigger”. In: *Journal of Scientific Instruments* 15.1, p. 24.
- Schöne, H. and S. Lechner-Steinleitner (1978). “The effect of preceding tilt on the perceived vertical. Hysteresis in perception of the vertical”. In: *Acta Otolaryngol.* 85.1-2, pp. 68–73.
- Schütze, Hinrich (1993). “Word Space”. In: *Advances in Neural Information Processing Systems 5*. Ed. by SJ Hanson, JD Cowan, and CL Giles. Morgan Kaufmann, pp. 895–902.
- Sewell, Abigail and David Heise (2010). “Racial differences in sentiments: Exploring variant cultures”. In: *International Journal of Intercultural Relations* 34, pp. 400–412.
- Shieber, Stuart M. (1994). “Lessons from a Restricted Turing Test”. In: *Communications of the ACM* 37.6, pp. 70–78.
- (2007). “The Turing test as interactive proof”. In: *Noûs* 41.4, pp. 686–713.
- Simon, Herbert (1969). *The sciences of the artificial*. MIT Press.
- Sinnott-Armstrong, Walter (1992). “An argument for consequentialism”. In: *Philosophical Perspectives* 6, pp. 399–421.
- Smith, Barry and Roberto Casati (1994). “Naive Physics: An Essay in Ontology”. In: *Philosophical Psychology* 7.2, pp. 225–244. URL: <http://ontology.buffalo.edu/smith/articles/naivephysics.html>.
- Smith, Henry (1996). *Restrictiveness in Case Theory*. Cambridge University Press.
- Smith, J. Maynard (1974). “Theory of games and the evolution of animal conflicts”. In: *Journal of Theoretical Biology* 47, pp. 209–221.
- Smolensky, Paul (1990). “Tensor product variable binding and the representation of symbolic structures in connectionist systems”. In: *Artificial intelligence* 46.1, pp. 159–216.
- Socher, Richard et al. (2012). “Semantic Compositionality through Recursive Matrix-Vector Spaces”. In: *Proc. EMNLP’12*.
- Socher, R. et al. (2013). “Zero-shot learning through cross-modal transfer”. In: *International Conference on Learning Representations (ICLR)*.
- Somers, Harold L (1987). *Valency and case in computational linguistics*. Edinburgh University Press.
- Sperber, Dan and Deirdre Wilson (1996). *Relevance*. Cambridge MA: Blackwell Publishing.
- Strang, Gilbert (2009). *Introduction to Linear Algebra*. Wellesley Cambridge Press.

- (2010). “Linear Algebra Lectures”. In: <http://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010>.
- Strawson, P.F. (1950). “On Referring”. In: *Mind* 59, pp. 320–344.
- Suddendorf, T. and E. Collier-Baker (2009). “The evolution of primate visual self-recognition: evidence of absence in lesser apes”. In: *Proceedings of the Royal Society B: Biological Sciences* 276, pp. 1662–1671.
- Swinburne, RG (1968). “Grue”. In: *Analysis* 28.4, pp. 123–128.
- Szabolcsi, Anna (2004). “Positive Polarity — Negative Polarity”. In: *Natural Language and Linguistic Theory* 22.2, pp. 409–452.
- Talmy, L. (1983). “How Language Structures Space”. In: *Spatial Orientation: Theory, Research, and Application*. Ed. by H. Pick and L. Acredolo. Plenum Press, pp. 225–282.
- (1988). “Force dynamics in language and cognition”. In: *Cognitive science* 12.1, pp. 49–100.
- Tarski, A. and S.R. Givant (1987). *A formalization of set theory without variables*. American Mathematical Society.
- Tarski, Alfred (1956). “The concept of truth in formalized languages”. In: *Logic, Semantics, Metamathematics*. Ed. by A. Tarski. Oxford: Clarendon Press, pp. 152–278.
- Tesnière, Lucien (1959). *Éléments de syntaxe structurale*. Paris: Klincksieck.
- Thalbitzer, William (1911). “Eskimo: an illustrative sketch”. In: *Handbook of the American languages*. Vol. 1. US Government Printing Office.
- Thomason, Richmond H., ed. (1974). *Formal Philosophy: Selected papers of Richard Montague*. Yale University Press.
- Thorndike, Edward L (1917). “Reading as reasoning: A study of mistakes in paragraph reading”. In: *Journal of Educational Psychology* 8.6.
- Titchmarsh, Edward C. (1939). *The theory of functions*. Oxford University Press.
- Titterton, D.M., A.F.M. Smith, and U.E. Makov (1985). *Statistical analysis of finite mixture distributions*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley. ISBN: 9780471907633.
- Trier, J. (1931). *Der Deutsche Wortschatz im Sinnbezirk des Verstandes die Geschichte eines sprachlichen Feldes Band I: Die Geschichte eines sprachlichen Feldes*. C. Winter.
- Turner, R. (1983). “Montague semantics, nominalisations and Scott’s domains”. In: *Linguistics and Philosophy* 6, pp. 259–288.
- (1985). “Three theories of nominalized predicates”. In: *Studia Logica* 44.2, pp. 165–186.
- Turney, Peter D. and Patrick Pantel (2010). “From Frequency to Meaning: Vector Space Models of Semantics”. In: *Journal of Artificial Intelligence Research* 37, pp. 141–188.
- Urmson, J.O. (1953). “The Interpretation of the Moral Philosophy of J. S. Mill”. In: *The Philosophical Quarterly* 3.10, pp. 33–39.

- (1958). “Saints and heroes”. In: *Essays in moral philosophy*. Ed. by A. Melden. University of Washington Press, pp. 198–216.
- Valiant, Leslie G. (1984). “A theory of the learnable”. In: *Communications of the ACM* 27.11, pp. 1134–1142.
- Van Der Waerden, B.L. (1930). *Moderne Algebra. Erster Teil*. Verlag von Julius Springer.
- Vincze, Veronika (2011). “Semi-Compositional Noun + Verb Constructions: Theoretical Questions and Computational Linguistic Analyses”. PhD thesis. University of Szeged.
- Voegtlin, Thomas and Peter Ford Dominey (2005). “Linear Recursive Distributed Representations”. In: *Neural Networks* 18.7, pp. 875–895.
- von Neumann, John and Oskar Morgenstern (1947). *Theory of games and economic behavior*. Princeton University Press.
- Wallach, W. and C. Allen (2009). *Moral machines: Teaching robots right from wrong*. Oxford University Press.
- Watkins, Calvert, ed. (1985). *The American Heritage Dictionary of Indo-European Roots*. Boston: Houghton Mifflin.
- Wells, Roulon S. (1947). “Immediate constituents”. In: *Language* 23, pp. 321–343.
- Whitney, William Dwight (1885). “The roots of the Sanskrit language”. In: *Transactions of the American Philological Association (1869-1896)* 16, pp. 5–29.
- Wierzbicka, Anna (1985). *Lexicography and conceptual analysis*. Ann Arbor: Karoma.
- Wilks, Yorick A. (1978). “Making preferences more active”. In: *Artificial Intelligence* 11, pp. 197–223.
- Williams, Edwin (1981). “On the notions ‘lexically related’ and ‘head of a word’”. In: *Linguistic Inquiry* 12, pp. 245–274.
- Wilson, Dierdre and Dan Sperber (2012). *Meaning and relevance*. Cambridge University Press.
- Winograd, T. (1972). “Understanding natural language”. In: *Cognitive Psychology* 3.1, pp. 1–191. ISSN: 0010-0285.
- Woods, William A. (1975). “What’s in a link: Foundations for semantic networks”. In: *Representation and Understanding: Studies in Cognitive Science*, pp. 35–82.
- Zimmermann, Thomas E. (1999). “Meaning Postulates and the Model-Theoretic Approach to Natural Language Semantics”. In: *Linguistics and Philosophy* 22, pp. 529–561.
- Zipf, George K. (1935). *The Psycho-Biology of Language; an Introduction to Dynamic Philology*. Boston: Houghton Mifflin.

Index

- accidental gap, 130
- accusative, 113
- action, 58
- algebra, 17
- alphabet, 26
- ambiguity, 2
- anaphor resolution, 165
- antitone, 217
- Aristotle, 162
- arity, 18, 100
- Artificial General Intelligence, AGI, 221
- Artificial General Intelligence, AGI, 13
- AT, 110
- atheism, 243
- attribute-value matrix, AVM, 163
- automated speech recognition, ASR, 222
- autonomy of syntax, 150
- axiom, 29
- Axiom of Choice, AC, 23

- bag of words, 80
- Basic English, BE, 109
- behavior, 102
- being, 60
- Bernoulli process, 141
- black bun, 207
- bleen, 77
- bound morpheme, 127

- cancer, 25
- categoricity, 32
- CAUSE, 64
- common ground, 154
- compositionality, 1
- compound value type, CVT, 197
- concept, 162, 175
- concept, 64

- concept, individual, 46
- conclusion, 33
- congruence, 21
- conjugation, 22
- connotative meaning, 6
- consistency, 32
- constituent structure, 169
- constructivism, 47
- context-free grammar, CFG, 106
- context-free grammar, CFG, 125
- contextuality, 3, 134
- cooccurrence restriction, 159
- coordination reduction, 142
- coreference resolution, 165
- cure, 25

- definition graph, 166
- deism, 243
- derivation, 127
- deterministic (semi)automaton, 88
- direct product, 20
- distinguisher, 80
- distribution, 95
- divine command, 248
- double negation, 33

- elementary equivalence, 31
- embedding, 43, 80
- emotive meaning, 6
- encyclopedic knowledge, 161
- entelechy, 13
- equality, 18
- equivalence, 21
- error-reject tradeoff, 2
- essentialism, 55
- ethical rationalism, 249
- Euclidean machines, 225

- eudaimonia, 241
- exist, 216
- extension, 72
- external model, 204

- feature, 38
- feature space, 37
- filler, 82
- filter, 22
- finite index property, 137
- finite intersection property, 22
- finite state transducer, FST, 55
- finite state automaton, FSA, 88
- finite state transducer, FST, 105
- fluent, 58
- free algebra, 26
- free monoid, 26
- free morpheme, 127

- gapping, 142
- Gaussian mixture model, GMM, 38
- grue, 77

- happiness, 59
- HAS, 62
- head, 110
- hedonism, 61
- homomorphism, 21
- homonymy, 175
- homunculus, 61, 230
- hyperedge replacement, 106
- hypothesis, 33

- ice, 24
- idea, 155, 162
- idea, 64
- identity, 18
- idiom, 99
- indeclinabilia, 178
- indexical, 214
- infelicity, #, 113
- inflection, 127
- information object, 97
- instrumental, 113
- intension, 72
- interpretation relation, 2
- intersectivity, 72
- intuitionistic logic, 33
- isomorphism, 21

- kernel, 22
- know, 75
- knowledge discovery, 202

- knowledge selection, 202

- latent semantic analysis, LSA, 43
- law of nature, 64
- law of the excluded middle, 33
- lexeme, 99
- lexeme, binary, 110
- lexeme, unary, 110
- lexical category, 99
- lexical category, 99
- lexicon, 99, 107
- lie, 205
- linear space, 17
- linguistic sign, 107
- literal, 29
- locality, 65
- locative inversion, 115
- Locke, 155, 162
- long short-term memory, LSTM, 141
- Longman Defining Vocabulary, LDV, 109
- Longman Defining Vocabulary, LDV, 193

- machine, 90
- matter, 58
- Mealy machine, 57
- meaning postulate, 77
- measurement, 38
- mereology, 58
- metaphor, 45, 101
- Mexican standoff, 231
- mind, 60, 64
- modal logic, 73
- mode, 89
- model structure, 29
- modus ponens, 34
- monad, 70
- monosemy, 2, 187
- monotone, 217
- Montague Grammar, MG, 71
- mood, grammatical, 208
- Moore machine, 57
- morpheme, 107
- morphotactics, 126
- must, 75

- natural kinds, 35
- necessary, 75
- necessity, 73
- negation, 208
- never give up, 230
- nominative, 113
- noun phrase, NP, 124, 142

- object (thing), 162
- obligation, 61
- opacity, 53, 72, 81
- operation, 18
- opposition, 44, 177
- optical character recognition, OCR, 36
- optical character recognition, OCR, 222
- ought, 61

- pain, 60
- panda, 150
- paraconsistent logic, 46, 210
- paradigm, 128
- parse tree, 169
- partition, 110
- pattern recognition, 52
- perplexity, 141
- phenogrammar, 113
- phrasal verb, 218
- phrase, 124
- Plato, 155, 209
- pleasure, 60
- Plotinus, 54
- polar adjectives, 200
- polar adjectives, 39
- polarity, 217
- polysemy, 175
- possibility, 73
- possible, 212
- pragmatics, 7, 84
- premiss, 33
- prepositional attachment problem, 15
- principal component analysis, PCA, 39
- printname, 163, 172
- probably approximately correct, PAC, 37
- prolepsis, 47
- pronoun, 214
- pronoun resolution, 165
- proposition, 45
- propositional calculus, 27

- quotient structure, 21

- rational prejudice, 207
- reading, 2
- Recognizing Textual Entailment, RTE, 81
- Recognizing Textual Entailment, RTE, 200
- recursive auto-associative memory, RAAM, 83
- relation, 18
- resource description framework, RDF, 197
- reversible language, 98
- Russell's paradox, 92

- salva veritate, 125, 252
- satisfaction, 30
- satisfiability, 32
- selectional restriction, 101, 111, 159
- self, 61
- self, 61, 208
- semantic differential, SD, 38
- semantic field, 185
- semantic generation, 2
- semantic interpretation, 2
- semantic web, 162
- semantics, v
- semiautomaton, 69, 88
- sense, 7
- sentence, 99
- sentimentalism, 247
- signature, 18
- slot, 82
- slot, paradigmatic, 128
- sort, 17
- soundness, 34
- Stanford Encyclopedia of Philosophy, SEP, ix
- state machine, 69
- state of affairs, 58
- state space, 56
- string, 26
- structure, algebraic, 17
- subcategorization, 100
- subcategory, 99, 145
- subdirect product, 20
- substructure, 20
- surface representation, 173
- synonymy, 2
- syntactic monoid, 95
- syntax, 94, 126
- system of sets, 22

- tautology, 33
- tectogrammar, 113
- term, 26
- tertium non datur, 33
- theism, 244
- theory, 29
- time tick, 57, 85
- transformation, 88
- transformation monoid, 88
- transition, 57
- truth, 205
- truth condition, 134
- Turing test, 13
- type/token count, 144

- ultraproduct, 23

unattested form, 130
universal grammar, 143
universal grammar, UG, 250
universe, 17
upward closed set, 22
utilitarianism, 245
utterance, 124
valid formula, 30

valid rule, 33
varimax, 40
vector, 17
vector space, 17

WANT, 53
word, 126
word sense disambiguation, 169

External index

λ -calculus
 4lang, main repository
 4lang
 A to D converter
 Abstract Meaning Representation, AMR
 ACT-R
 AI
 Allen Newell
 ambiguity
 anaphora
 Anti-Foundation Axiom, AFA
 Apollonius
 arete
 Argument from Disagreement
 argumentum ad populum
 Aristotle
 Arrow's Impossibility Theorem
 Artemis
 Artificial General Intelligence, AGI
 Artificial Intelligence, AI
 AI box
 artificial neuron
 attractive nuisance
 attribute grammar
 attribute-value notation
 attribute-value pair
 automorphism
 autosegmental representation
 Axelrod Tournament
 Axiom of Foundation
 Bacon, Roger
 bag of words
 Begriffsschrift
 Berkeley Construction Grammar
 Bernoulli scheme
 bignum arithmetic
 bigram
 binary relation
 Birkhoff's Theorem
 bit
 black bun
 Blish, James
 Bourbaki
 Brasilia
 Buridan
 cancellation property
 Cartesian product
 case
 case grammar
 Catalan number
 categorical perception
 category
 Christo
 circuit theory
 click
 clitic
 Codd
 codomain
 cognitive linguistics
 coherence
 combinator calculus
 Combinatory Categorial Grammar, CCG
 Commentary on the Gallic war
 communicative dynamics
 comprehension
 computability
 computable function
 concept
 Condorcet voting paradox
 cone cell
 cone
 congenital analg
 conjunctive normal form
 consensus theory of truth
 consequentialism
 construction grammar
 content word
 context-free grammar, CFG
 Cook-Levin theorem
 coreference resolution
 correspondence theory of truth
 covariance matrix
 Cratylus
 Culturally Authentic Pictorial Lexicon

Curry
 cyclic group
 D to A converter
 data structure
 dative shift
 DBpedia
 De Medicina
 deep learning
 deism
 deixis
 dependency grammar
 dependency graph
 design plan
 determinism
 diacritic feature
 dict_to_4lang
 dictionary.com
 dihedral group
 dimension reduction
 direct reference
 distributional semantics
 divine command
 domain
 don't do that then
 double negation
 dummy
 Eilenberg machine
 elementary classes
 embedding
 embodied cognition
 empty string
 endomorphism
 engrams
 entelechy
 enthymeme
 epistemic modality
 ergativity
 eschatology
 essentialism
 essive
 etymology
 Euclidean Algorithm
 euphemism
 Event Calculus
 evolutionary algorithm
 ex falso quodlibet
 exegesis
 factor analysis
 feature, binary
 feedback vertex set
 Fermat's Last Theorem
 Fermat
 filled pause
 final cause
 finite state automaton, FSA
 Finite State Transducer, FST
 Firth
 flip-flop
 flowchart
 fluent
 formal grammar
 formal language
 free Boolean algebra
 Free Will Theorem
 free will
 Freebase
 freight elevator
 friendliness
 Frobenius norm
 function word
 functional MRI
 function
 Galois connection
 gantry crane
 gapping
 general systems theory
 generative semantics
 generic
 GF(2)
 GitHub
 Goal-Oriented Action Planning, GOAP
 Goodstein's Theorem
 Google 1T corpus
 grammaticality

Grice
 group
 Gödel incompleteness theorem
 Gödel completeness theorem
 Harris, Roy
 Harris, Zellig
 Hasegawa, Chiyono
 head directionality
 HPSG
 headword
 Herbert Simon
 higher-order logic
 Hillary Step
 Hippocratic lunes
 homonymy
 homunculus argument
 horror vacui
 Hume
 Hun*
 hypergraph
 hysteresis
 identity element
 Idiosyncrasies of distribution project
 Iliad
 impossible worlds
 indexical
 information retrieval, IR
 information extraction
 information theory
 injective
 innatism
 interlinear gloss
 International Phonetic Alphabet, IPA
 intuitionistic logic
 involution
 isolating language
 Jakobson
 JSON
 JSTOR
 Kant
 Kaposvár
 keiyōdōshi
 KISS
 Kleene's Theorem
 knowledge representation, KR
 Krohn-Rhodes theory
 Kung-sun Lung
 Kálmán-Peredy handouts
 Lagrange multiplier
 language resources
 Latent Semantic Analysis, LSA
 lattice
 LDOCE
 LDV
 leap second
 learning mechanism
 Leipzig glossing rules
 lemma
 lexeme
 Lexical Functional Grammar, LFG
 lexicography
 light verb
 Lindemann
 linear bounded automaton, LBA
 linguistic sign
 li
 Locke
 Locke Essay
 logical form
 Long Short-Term Memory
 Longman Defining Vocabulary
 LSA style sheet
 Macy conferences
 Masorettes
 McCarthy
 Mealy machine
 measure zero
 Meinong
 memristor
 mereology
 minisat
 mirror neuron
 MNIST
 modal logic

modality, deontic
 modality, epistemic
 modality, medieval
 modality
 modists
 modus ponens
 Mohism
 Monadology
 monad
 monoid
 Montague Grammar
 mood
 Moore machine
 moral anti-realism
 moral sentimentalism
 moral universalism
 morpheme
 morphology, non-concatenative
 morphology
 Mr. Hug
 Mt. Olympus
 multi-word expression, MWE
 multiplication table
 n-gram
 Nahuatl
 naïve set theory
 nativism
 natural kind
 naturalistic fallacy
 neural networks
 No true Scotsman
 non-existent objects
 NP complete
 numerosity
 NumType
 Ogden list
 Old French
 one-shot learning
 one-way function
 onomatopoeia
 opacity
 OpenBLAS
 OpenCog
 opera slipper
 ordered pair
 Organon
 orthography
 Osgood
 ouroboros
 OWL
 paralinguistic
 parallel text
 parse tree
 part of speech (POS) tagging
 partial differential equation, PDE
 passive voice
 pattern recognition
 Peano Arithmetic, PA
 Peirce
 perplexity
 Petri net
 Petrus Hispanus
 philosophy of language
 philosophy of science
 phoneme
 phonology
 phrase
 phrase structure grammar physicalism
 PlanetMath
 Plato
 pointwise mutual information, PMI
 polymorphism
 polynomial
 polysynthetic language
 Porter stemmer
 POS tagger
 possibility
 power law
 PP attachment
 pragmatics
 Prague School
 premiss
 priming
 Principal Component Analysis, PCA

Principle of Explosion
 productivity
 proof assistant
 programming language semantics
 Project MUSE
 PronType
 proofs as programs
 proposition
 prototypicality
 proxy
 psychometric g factor
 qualia
 quantum gate
 question answering
 RDF
 RDFS
 reading of Chinese
 README
 real number
 realism
 Recognizing Textual Entailment, RTE
 Reed-Kellogg diagram
 regular expression
 reism
 relationship extraction
 revealed teachings
 rewrite rule
 ring
 Robinson's Q
 robotics
 Roget's Thesaurus
 Rta
 Rule of Tautology
 Russell's paradox
 salva veritate
 scalar implicature
 Schank
 Schmitt trigger
 Schur's Lemma
 Second-order arithmetic (Z_2)
 sekki
 self
 Semantic Differential
 semantic field
 semantic network
 semantic web
 semantics, logic
 semantics, programming languages
 semigroup
 semiotics
 Semitic
 separating mechanism from policy
 set
 Shakespeare
 sigmoid function
 Simple English wikipedia
 Singular Value Decomposition, SVD
 Skolem paradox
 Skolemization
 sluicing
 SOAR
 society of mind
 software orrery
 solar shade
 sorites
 spreading activation
 Stanford Encyclopedia of Philosophy
 Stanford Parser
 stemming
 Stevin
 stoichiometry
 stoicism
 strength
 sui generis
 supererogation
 support vector machines
 SVO word order
 synthetic language
 TACIT project
 Tarski's World
 tensor product
 tertium non datur
 text to speech, TTS
 The Heap

The problem of evil
theological voluntarism
topicalization
Torricelli's experiment
transcription
transformation semigroup
transformational grammar
Tree-Adjoining Grammar, TAG
tree
Turing machine
Turing test
Universal Dependencies, UD
Universal Grammar, UG
unmarked
Urban Dictionary
urelement
uroboros
utilitarianism
Voronoi tessellation
Wantzel
Watson
web form
Webster's Third
Western philosophy
wh-question
Wikipedia disambiguation pages
Wikipedia
William of Sherwood
William of Ockham
word sense disambiguation
WordNet
Wundt
yacc
{yak}
yamato
YamCha
Yokutsan
Young Earth creationism
Zipf's law
Zorn's Lemma