Chapter 13

What is the simplest semantics imaginable?

András Kornai
Budapest University of Technology and Economics

We claim that three binary relations, 0, 1, and 2, are both necessary and sufficient for formal semantics: 1 and 2 are the well-known "subject of" and "object of" relations, and 0 corresponds to the subsumption or "is a" relationship well known from knowledge representation. We describe how these can be used to compositionally assign a semantic representation built from primitives (morphemes, semantic atoms) and how the system can be related to the computational "word vector" semantics which is surprisingly effective even though it appears to employ no grammatical rules or constraints.

1 Introduction

There is no evidence that in English the vestigial system of object marking can be extended beyond personal pronouns, yet we have little doubt that English speakers can fully grasp transitive constructions involving inanimate objects. Since most linguists assume that coordination and subordination will be present in every language, Everett's discovery of a language lacking syntactic facilities for these is seen as some grave error akin to a hypothetical discovery of a language lacking subjects and objects. But when viewed from the perspective of semantics, impoverished syntax is no more surprising than impoverished morphology, so the question should be: what is the absolute minimum we require for semantics?

In this paper, we start from the simplest imaginable cases, subject-predicate and modifier-head constructions, and gradually build up a system of semantic

András Kornai. 2024. What is the simplest semantics imaginable? In Edward Gibson & Moshe Poliak (eds.), *From fieldwork to linguistic theory: A tribute to Dan Everett*, 321–338. Berlin: Language Science Press. DOI: 10.5281/zenodo.12665929

representation both in the tradition of Knowledge Representation (KR; Brachman & Levesque 2004) and in the contemporary "thought vector" approach (LeCun et al. 2015, Kornai 2023). These are not competing but complementary views of the same subject matter, both true at the same time like the algebraic and the function-theoretic views of polynomials. While the KR view does not significantly depart from the common linguistic view that structures are to be represented by some kind of graphs (an idea common to transformational and dependency grammar formalisms), the vector approach is very unfamiliar: if the representations are *n*-tuples of numbers, what are the rules?

This question is especially vexing in light of the observation that the main vector operation, vector addition, plays only a marginal role in the computational system: it is used for solving analogical puzzles like *France is to Germany as Paris is to X* (Mikolov et al. 2013) and little besides. Using the KR side to explore the issue we find that three binary operations, 0, 1, and 2, are both necessary and sufficient for formal semantics. 1 and 2 are the well-known "subject of" and "object of" relations, and 0 corresponds to the subsumption relationship known as "is a" in KR and as hyponymy in lexicography. (The vector equivalents of these operations are somewhat more technical, and are not required for making our main point that the minimum is three – see Kornai (2023) for details.)

The sufficiency of these operations is not trivial – students of Relational Grammar and many similar systems will no doubt wonder about "3" and perhaps different kinds of linkers such as thematic (proto)roles or kārakas. For indirect objects, the reader is referred to Kornai (2012), and for deep cases, thematic roles and kārakas see Chapter 2.4 of Kornai (2023). The main line of attack in reduction to "1" and "2" is that "3" can itself be considered (together with other conceptual relations typically expressed by case markers and adpositions) to have their own subjects and (prepositional) objects. This will of course complicate the graphs (in ways that will be familiar from generative semantics) but ensure that we will never need hyper*edges* just hyper*nodes*. The resulting system is rather similar to the Resource Description Framework used in the Semantic Web where binary relations are encoded in a (subject verb object) triple. Since such triples can be substituted for one another, for *give* we obtain an analysis "cause to have" so that *x gives y to z* becomes `(x cause (z has y))`. This method is immune to the standard criticisms (Fodor 1970) leveled against generative semantics-style meaning decomposition that were based on the pronominalization possibilities of the 'to + inf' natural language paraphrase, since the formulas explicitly contain this information. Kornai (2010) discusses how the other criticism, that such a decomposition (cause to die → cause not to have life functions → cause not

to metabolize, respond, ...) may never terminate, is actually irrelevant in an algebraic setup that enables circularity, and Kornai (2012) describes how higher arity verbs, such as *promise* can be handled in the same manner.

But the necessity of three different operations is even less trivial: after all, natural language semantics is often viewed as translation to First Order Predicate Calculus (FOPC; Blackburn & Bos 2015) and via combinators (Curry & Feys 1958) FOPC can be reduced to strings of a single symbol J with the appropriate parenthetization (Schönfinkel 1924, English transl. van Heijenoort 1967). We can take the no-frills approach further, since the parens can be eliminated in favor of Reverse Polish Notation (RPN), leaving us with binary strings. As the first symbol is always J, which we denote by '1', we can use '0' for the binary operator symbol of RPN, and we are guaranteed that each well-formed predicate formula corresponds to a unique integer written in base 2. Furthermore, the translation between the original formula and the binary number is computable mechanistically in either direction by a rather simple Turing machine. Taking this to the extreme, binary integers can be written in base 1, and again translation between the formats by a Turing machine is available in both directions, so that all we need is a single symbol which can be repeated as many times as we need. If we are happy with integers, base unspecified, Gödel numbering would work just as well.

This is not just a walk through some rarely visited pages of the mathematical logic bestiary. There are sophisticated attempts at using combinatory logic in semantics since the 1980s (Szabolcsi 1987, Steedman 1987, Jacobson 1999, Baldridge 2002), with important links to mild context sensitivity/polynomial parsability (Joshi et al. 1991). Clearly, neither FOPC nor higher order intensional calculi such as those employed in Montague Grammar have a privileged status as the One True Formalism (OTF) for semantics, and the search for OTF is not a trivial one. Our argument will rely on a stricter understanding of compositionality than the one generally assumed: while the mapping from Gödel numbers (or binary strings) back to logic formulas is unique, and Turing-computable, not every such mapping is compositional in the accepted sense of taking some string X, decomposing it *by simple means* as AB, and computing the meaning of the whole from the meanings of the parts A and B.

In Section 2 we set expectations by discussing some important desiderata for OTF. We also introduce some less commonly taught desiderata students of linguistic semantics may not even have heard of, such as *smooth transition from morphology to syntax* and *embeddability*, and argue that these are actually part of the same cluster of desiderata. Our own proposal, the `4lang` system (see https:

//github.com/kornai/4lang/tree/master/V2), is discussed in Section 3, where we return to the issue whether there is, or should be, a minimal system among the proposals meeting the desiderata.

2 What do we expect of semantics?

Let us begin with some standard desiderata:

D1 Comes with reasonable model theory

D2 Reasonably simple (compositional) mapping from natural language to OTF

D3 Mapping in the reverse direction into passable natural language so that OTF can serve as a translation pivot

D4 Usable for disambiguation

D5 Usable for characterizing synonymy

D6 Extends smoothly to verbal description of non-verbal material (music, scientific models, functional description of algorithms, ...)

D1 is taken very seriously by proponents of logical semantics, who treat all other approaches (by natural language paraphrase, by diagrams, and by KR in general) as *markerese* since Lewis (1970). To satisfy this, OTF must contain three well-defined parts: a language of formulas L, a collection of models $\mathcal{M}$, and an interpretation relation $i : L \to \mathcal{M}$ between the two (Tarski 1956). By well-defined we mean the existence of effective procedures to decide whether something is a (well-formed) formula and to decide whether something amounts to a model. The mapping itself needs to be not just effective (Turing-computable), but computable in a particularly simple manner we will discuss at D8 below.

For linguistic semantics to follow the same architecture one would expect L to contain all well-formed (grammatical) strings, and only these, and would use $\mathcal{M}$, the collection of models, to capture the world that is being talked about, with i mapping elements of the language onto their meanings. In reality, Montague Grammar (MG; Montague 1970, 1973) represents a considerable departure from this architecture. On the left side, we do not find 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 (see discussion of D4 below).

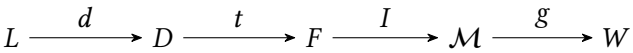


Figure 1: Information objects associated with MG

On the right side, we do not find real-world objects or even formal objects (models), but formulas F of a particular logic calculus. The full picture of MG is composed of the first two or three arrows in Figure 1, with the primary attention focused on the translation homomorphism t. The models $\mathcal{M}$ are 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 is completely left out – Montague was no doubt familiar with Quine's and others' criticism of direct reference.

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. In this key respect, MG actually fails D4.

D2 is also taken very seriously, so much so that important ranges of phenomena where it obviously fails, such as noun-noun compounding, are simply declared out of scope for semantics. Fodor (1998) is typical in treating all word meanings as atomic, i.e. ignoring all productive morphological phenomena. This of course requires the memorization of all word meanings and brings back the psychological problem (Partee 1979, 2013) of accounting for infinite data sets in a finite brain.

Clearly, expressions like *ropeladder* 'ladder made of rope,' *testtube* 'tube used for testing,' and *manslaughter* 'slaughter undergone by man' (Kiparsky 1982) are not entirely compositional. Equally clearly, the meaning of novel compounds is largely predictable, as are the meanings conferred by productive derivational processes. The Lexicalist Hypothesis (Chomsky 1970) segregating morphology from syntax is clearly untenable (Bruening 2018), and in its place we offer our own desideratum:

D7 Compositional (syntactic) and non-compositional (morphological) processes must be part of the same continuum

In other words, there cannot be a different semantics for morphology and for syntax, especially as the border between the two is not uniform across languages.

It must be one and the same interpretation mechanism that takes you "from morpheme to utterance". This is not to say that there is no *word* unit that syntax can refer to (the classical psycholinguistic evidence in favor of memorized units with lexicalized meanings cited in Müller (2018) is hardly controvertible), but simply to insist on deriving as much of this meaning by compositional means as possible. In Section 3 we offer a mechanism that deals with the non-compositional aspects by means of subdirect products, which contain the fully compositional direct products as a limiting case.

D3, while in principle compatible with many theories, is seriously underresearched. Using a natural language (typically English) as pivot (intermediary) between two languages is common both in manual and in machine translation. The use of a formal language is almost unheard of: the only proposal with actual translations is Universal Networking Language (Cardeñosa et al. 2005), and the use of logic formulas is unattested. Given how common it is to consider semantics "the language of thought," the single-minded focus on translation to, but never from, mentalese is rather surprising.

This onesidedness cannot be entirely attributed to the fact that systems of translation to logic formulas (including descendants of MG such as Dynamic Predicate Logic) have very little coverage to begin with. It appears the real issue is lack of transparency, a phenomenon well observable on the Schönfinkel-style reduction step of replacing the standard S and K combinators by a single combinator J. This J is defined by cases:

$$Jx = \begin{cases} K & \text{if } x = S \\ S & \text{otherwise} \end{cases}$$

Therefore, we have $JJ = S; J(JJ) = JS = K$ eliminating the original S and K entirely in favor of a single entity. Notice that the method would be just as applicable if we didn't have 2 things to reduce but 52. We would only need to stretch the case-by-case definition accordingly (see Curry & Feys 1958: Chapter 1E4).

For a concrete example, consider the translation of the English reflexive pronoun *himself* which Szabolcsi (1987) argues to be the combinator W, defined as $Wxy = xyy$. In the standard S, K basis W is expressible as $((SS)(SK))$ so W is $(((JJ)(JJ))((JJ)(J(JJ))))$. Continuing with the no-frills approach, the order of applications encoded in the parenthetization can be just as well encoded by RPN, using the operator symbol $\circ$. This will make the formula into $JJ \circ JJ \circ \circ JJ \circ JJJ \circ \circ \circ \circ$ which, by transliterating J as 1 and $\circ$ as 0 becomes the binary number

11011001101110000, better known to us as decimal 111472, which could be written in unary base as a string of 111472 1s (see Fokker (1989) on how to obtain one-combinator bases).

It is worth emphasizing that the tricks of converting to combinatory logic, using the Schönfinkel reduction, converting the parenthesized *J* strings to binary numbers (and finally converting the binaries to unaries) are not essential for this undertaking. As is well known to students of logic, every formula (e.g. the kinds of formulas used in Montague's intensional logic) can be converted to a number by Gödel numbering[1], and a Turing-computable and invertible mapping of natural language meanings to numbers is not hard to define.

But when we see decimal 69720375229712477164533808935312303556800 what is it exactly that we see? Well, we see $2^6 \cdot 3^4 \cdot 5^2 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 \cdot 53 \cdot 59 \cdot 61 \cdot 67 \cdot 71 \cdot 73 \cdot 79 \cdot 83 \cdot 89 \cdot 97$ which would be the Gödel code for $[6, 4, 2, 2, 1, ..., 1]$ (a total of 21 1s). The problem is not that the translation back from the Gödel code to the *n*-tuple is not computable, but rather that it is not at all transparent, requiring a relatively powerful Turing machine to compute. For a translation, we would want compositionality, D2, which in turn requires a transparent machine, one that finds the boundary in the expression to make the first split into substrings A and B, and can recursively repeat the process for A and B. The real problem is that there is no boundary in the decimal number that the Gödel coding yields. Even if there were compositional boundaries in the original, these are washed out in the encoding process. Therefore, we replace the original desiderata D2 and D3 with D8 (mnemonic: $8 = 2^3$):

D8 The form ↔ meaning mapping should be maximally transparently compositional in both directions

D4 and D5 are part of the Katz & Fodor (1963) criteria that for many years were (and in many ways remain) the standard statement regarding the adequacy of any semantic theory:

> A semantic theory describes and explains the interpretative ability of speakers by accounting for their performance in determining the number and content of the readings of a sentence, by detecting semantic anomalies, by deciding on paraphrase relations between sentences, and by marking every other semantic property or relation that plays a role in this ability.

[1] https://en.wikipedia.org/wiki/Gödel_numbering

Over the years, as emphasis gradually shifted from lexical to compositional semantics, it became clear that these criteria are exceedingly hard to meet: D4 required some one-to-many mapping from form to "disambiguated language", a technical device that (somewhat akin to universal phonetic realization) was never worked out in sufficient detail.

D4, together with D5, which is generally conceived of as a many-to-one mapping from different forms to the same meaning, jointly amount to assuming a form-to-meaning relation that is not functional in either direction. But the branching factors are very different: ambiguity is everywhere, synonymy is rare, in fact it is often claimed that no two natural language expressions are perfectly synonymous. This, if true, is highly problematic for Boolean connectives, where the logic creates synonymy: if something is translated as $p \wedge q$ it is perforce translated as $q \wedge p$ which then translates back to a non-synonymous natural language expression. This in fact happens: *I went home and had dinner* is not synonymous to *I had dinner and went home*.

This particular problem instance can be eliminated by insisting that the logic translation must also include an update of the temporal index that tracks event time, but the overall problem is much harder, since now all natural language tautologies must mean the same thing $\top$, and all natural language falsities must mean the same thing $\bot$. For this reason in Section 3 we will considerably relax D4 and D5: whatever is OTF, translation from it should not be more difficult than translation to it, and a full capture of ambiguity and paraphrase is impossible.

D6 is very ambitious, and is not shared widely among linguists, except those with a more semiotic bent. Clearly, there is such a thing as "the language of music." It even has a written form, scores. But it is not clear that when we say that "music speaks to us" we mean the sequence of notes as traditionally depicted in scores: everyday experience shows that mechanical rendering of a score often fails to elicit the kind of emotional response that is triggered, according to many artists, precisely by those minute departures from the score that are the essence of human interpretation.

The same can be said for scientific theories: it is hard not to be touched by a deep sense of awe when understanding the Maxwell equations. But the awe is not a constitutive factor of the equations, and it is not clear how it is communicated to us, it just is there: we see the truth, and we marvel. And it's not the truth, in and of itself, that triggers this response: we also see the truth of $3 = 3$ but we don't particularly marvel.

This is not to say that music or science are somehow ineffable, impossible to explain, but without some notion of what is it that needs explication it is very hard to make progress on their semantics. With D1, as commonly understood,

this is much easier, because one of the several functions of natural languages is the interpretative function, to tell us things about the world, and model theory is an attempt to explicate how things are (or at least how things can be) in the world. If we had a substantive theory of being awestruck, "feeling great respect for the importance, difficulty, or seriousness of someone or something" (LDOCE, Procter 1978), we could make some progress on the semantics of these non-linguistic domains by leveraging the lexical semantics of words like *awe*, a matter we shall return to in Section 3.

Until now we have discussed a set of desiderata that any semantic theory should meet, selecting D1, D7, and D8 as our central desiderata. D2 and D3 are subsumed under D8, while D4, D5, and D6 are seen as *good to have*s, criteria that must be subordinated to the central ones. That failure to meet these three is not generally considered fatal is best seen from the widespread acceptance of MG and similar theories.

Perhaps the most important takeaway so far concerns D8, compositionality. The point of our "logic bestiary" examples is that semantics requires more than any old Turing-computable algorithm, it requires a specific mechanism of *decomposing* expressions into constituent parts, and computing the results based on the parts. Decomposition itself must be a simple operation, ideally expressed by a low-power Turing machine such as a finite state transducer that detects the constituent boundary. The overall semantics is obtained by (i) successive decomposition steps that together yield a parse tree of the input, and (ii) rolling back these steps by merging constituents. Proposals for these two steps go back as far as Wells (1947) and Knuth (1968) respectively. Whether the parse tree is strictly binary or not, whether it can contain discontinuous (gapped, interleaved) constituents are questions of great technical importance, but compositionality can be achieved either way.

This leaves us with one central desideratum we have not touched upon so far, *learnability*. In theory, the interpretation mechanism can be given externally (e.g. as a lex/yacc parser), but in practice we would prefer the entire algorithm to be learnable, ideally from positive data alone. Whether this is just good to have, or a non-negotiable desideratum as urged by Chomsky (1965) is hard to say, but one thing is clear: so far, all successful learners are *supervised*, requiring labeled data. These include *self-supervised* techniques where the labels are generated by simple automated methods from initially unsupervised data (raw text). At the price of demanding orders of magnitude more data than encountered by human language learners during language acquisition, such self-supervision is used to great effect in Large Language Models.

The difference between the purely symbolic algorithms, such as lex/yacc parsers commonly developed for computer languages by their creators on the one hand, and the machine learned algorithms on the other, generally boils down to a difference between the use of symbolic debugging versus optimization. The learning algorithm closest to the former is "principles and parameters" learning as proposed in Chomsky & Lasnik (1993), which has many precursors in formal language theory (for a survey, see Angluin 1980).

Since Large Language Models (LLMs) are far more successful in acquiring syntax than any symbolic approach, the hopes of acquiring semantics by symbolic means are rather dim, especially as compositionality requires the acquisition of a system that creates the parse tree, i.e. the acquisition of at least rudimentary syntax capabilities. Therefore, making the system optimization-friendly appears as a central desideratum. Since optimization is performed by gradient descent, this requires a system, any system, that states the problem in a framework where gradient descent is feasible, i.e. a smooth system where derivatives can be computed. Whether derived from a learnability desideratum or seen as a practical necessity, we have

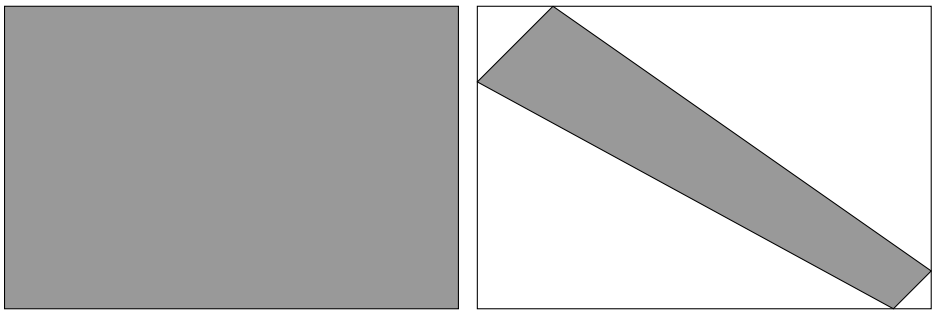
D9 The problem statement must be embedded in a differentiable setup

One of the key inventions that powered the LLM revolution was enabling gradient learning by means of a new semantic structure, *word vectors* (Schütze 1993, Collobert et al. 2011). This is by no means the only relevant invention: we already mentioned *self-supervision*; and we should mention at least *byte pair encoding* (Gage 1994); *sequence to sequence transformation* (Sutskever et al. 2014); and *attention* (Vaswani et al. 2017). By replacing the discrete tree structures used since Katz & Fodor (1963) for encoding the meaning of lexical items by vectors in n-dimensional space where partial derivatives can be taken, learning based on optimization became possible. It is worth emphasizing that the resulting continuity/differentiability fully applies to the terminal nodes in the representation of lexical meanings, which were conceptualized as discrete (typically, binary) features by Katz and Fodor, and rightly objected to as "atomization of meaning" by Bolinger (1965).

In the next Section we turn to the vector-based, and thus optimization-friendly `4lang` system, with special emphasis on meeting the desiderata by a minimal system from this class of models. In fact, the system is so skeletal that the vectors can be computed just by solving a system of equations, a goal that makes particular sense for "low density" languages where training data is in short supply.

3 Hypergraphs and their linearization

In what follows, we take the system of polytopes[2] induced by word vectors as our starting point (Kornai 2023), and begin with the trivial observation that the *thought vectors* of LeCun et al. (2015), which are intended as semantic representations of the (already spoken part of) sentences and larger discourses, appear in the same space. This takes care of D7, which asks for a style of representation that is common to subword units (morphemes, or the bytepair-like units used in the WordPiece algorithm of Wu et al. 2016), phrases, sentences, and even larger units. In this system, non-compositionality corresponds to *subdirect* products, and compositionality appears as a special case, *direct* products (Kornai 2010) – the difference is illustrated in Figure 2.



(a) Direct product (b) Subdirect product

Figure 2: Direct and subdirect products of the same two intervals $[0, 12]$ and $[0, 8]$

The subdirect product[3], standardly defined as a *sub*set of the *direct product* satisfying projection requirements is not unique: there can be many subsets of the direct product that project onto both components. This means that the semantics itself is underdetermined, but this is only to be expected in cases like noun-noun compounding. Whatever portion of the semantics is rule-governed is captured, e.g. that in N-N compounding we have 'N_2 that is V-ed by N_1' with the V indeterminate: ladder *made of* rope, slaughter *undergone by* man, tube *used for* test (Kiparsky 1982), the non-compositional part is admitted as suChapter This seems to be the right approach not just for morphology, but also for the grey zone of *constructions* between the purely morphological and the purely syntactic such as *NP of NP* studied in Berkeley Construction Grammar[4], taking care of our desideratum D7.

[2]https://en.wikipedia.org/wiki/Polytope

[3]https://en.wikipedia.org/wiki/Subdirect_product

[4]https://www1.icsi.berkeley.edu/~kay/bcg/ConGram.html (Kornai 1988)

What are, then, the non-negotiable elements of vector semantics? One, perhaps the most important one, is the notion of containment, IsA, which we see as essential for the reconstruction of Aristotelian *genus*. Whatever definition we provide for *dachshund* or *labrador*, the first thing in the definiens will be *dog*. Given that we use polytopes (polyhedra-line *n*-dimensional regions) around the word vectors, IsA comes for free as the set-theoretical inclusion "⊂" relation. This works well for ordinary (intersective) adjectival modifiers as well: a *brown dog* is in the intersection of the *brown* and the *dog* polytopes. (For non-intersective adjectives like *former*, see Chapter 3.2 of Kornai 2023).

The method of assigning semantics to *Kim is a donkey* by leveraging set-theoretical containment cannot be directly generalized. Clearly, there is nothing in set theory that would directly work for *Kim has a donkey*, but the underlying idea of taking a relation, in this example the possessive relation Has, and using that for assigning meaning, is solid. (Has can be further subdivided into inalienable and ordinary possession, but we will not pursue this matter here.) There remains one technical difficulty: however the language signals the distinction, *John ate the fish* and *The fish ate John* should not be treated as synonymous. We use SubjectOf and ObjectOf for the disambiguation. These are good candidates for universality, even in languages where the distinction is made in absolutive/ergative terms.

With this, we are done – we don't need further disambiguators (deep cases, thematic roles or proto-roles, etc.) to get to ditransitive or even higher arity predicates, since these can be obtained by classic techniques of meaning decomposition that go back to generative semantics (Kornai 2012). (The `4lang` system writes `=agt` and `=pat`, but we could have written "1" and "2" as well – the only theoretical claim here is that there is no "3" required.) The representation structures we obtain are best depicted as hypernode graphs that can contain other such graphs as nodes (but not as edges). These should be familiar from the Resource Description Framework[5] that is standard on the WorldWideWeb.

It is easy to check that the system presented here meets our desiderata D1 and D8 as well, so our work is done. Readers interested in how the system can be extended, without adding further operators, to issues of temporal and spatial semantics, indexicals, negation, quantification, probability, modality, gradience, implicature, and other issues generally considered relevant for semantics are advised to look at Kornai (2023). But one word of caution is in order: not having further operations is not the same as not having further primitives.

[5]https://en.wikipedia.org/wiki/Resource_Description_Framework

The 4lang system actually treats a handful of binary relations AT, BETWEEN, CAUSE, ER, FOLLOW, FOR, FROM, HAS, IN, INS, ISA, LACK, MARK, ON, PARTOF, UNDER as primitives (and makes the claim that all others are derivable). These correspond to matrices, rather than vectors. Remarkably, what traditional syntax treats as higher order operators, quantifiers in particular, will require only vectors, rather than full matrices: the central example is the generic quantifier gen, which simply corresponds to the n-dimensional vector $(1/n, 1/n, ..., 1/n)$ (for details see Kornai (2023) Chapter 4.5). The bulk of the primitives are unaries (vectors) appearing in a system of mutually constraining definitions, and this includes most verbs that can have an optional object like *eat* as well.

With *eat* it is reasonably easy to see how one can define it in terms of the Longman Defining Vocabulary 'to put food in your mouth and chew and swallow it' and the process of turning this into a 4lang clause can be automated (Recski 2016) to yield `=agt cause_ {=pat in mouth}, swallow, <=pat[food]>, <bite/1001>, <chew>, =agt has mouth`, which uses an even smaller defining vocabulary of 739 elements (including the 16 binaries).

Arguably *eat*, if not a universal semantic primitive, is at least very close to being one, and clearly it is a "simple" word (Kornai 2021) that comes very early in language acquisition. Our earlier example, *awe*, is clearly far from the simple/basic layer of the vocabulary, but the same method remains applicable: take the LDOCE definition, in this case 'a feeling of great respect and liking for someone or something', normalize the syntax, and reduce further until only the 4lang primitives remain. We begin with *for someone or something* and replace it by `=pat`. *great* and *liking* are defined, *great* as `big` and *like* as `feel {=pat[good], good for_ =agt}`. For *respect*, we have to go back to LDOCE to obtain "admire" , for which we obtain 'to look at something and think how beautiful or impressive it is'. The process goes on, but for *beautiful* we obtain "extremely attractive" and with `attract` we terminate at `=agt cause_ {=pat want {=pat near =agt}}`.

This may appear tedious, but eventually all non-4lang words are eliminated, since the system was constructed from the Longman Defining Vocabulary by systematic elimination (Ács et al. 2019) until a feedback vertex set[6] is obtained. The price of the termination guarantee is that the resulting set is considerably larger than the system of Natural Semantic Metalanguage (NSM; Wierzbicka 1992, 1996, Goddard 2002), which in many ways served as an inspiration. But 4lang both has a formal syntax and guarantees that all words not defined in the core are definable by it via LDOCE, whereas NSM uses an informal (English) syntax, and has no guarantees that words outside the core are actually definable as NSM stanzas.

[6] https://en.wikipedia.org/wiki/Feedback_vertex_set

As for minimality, we make no claim that the set of `4lang` primitives is truly minimal, just that by systematic reduction of the entire English vocabulary we arrived at a stage where we see no further reduction possibilities. This does not mean that for other languages no further reductions would be possible, and it would be an interesting research program to (i) harden NSM syntax until it becomes machine-parsable and (ii) define the `4lang` primitives in terms of the NSM primitives. Whether this is possible remains to be seen, but our system already provides an upper bound on the dimension of the vector space we use for modeling semantics.

4 Conclusions

Minimality requires thrift both in the number of operations and in the number of primitives manipulated by these. To maintain compositionality in both directions, the "bestiary-style" minimalism of (Gödel) numbering has to be sacrificed for more transparent operations. Of particular interest is the case when the objects manipulated are vectors and matrices in finite-dimensional Euclidean space, since these can be acquired gradually, by optimization techniques that change the vectors only a little bit as new learning data becomes available, rather than by huge and unpredictable discrete steps that require a complex system of inborn directives.

As for the primitives, our current system is likely overcomplete[7], at least as far as the vectors (unaries) are concerned, though we seem to approach the limits of reducibility for the matrices (binary relations) used. Remarkably, it is not the verbs, transitive, ditransitive, or even higher arity, that require departure from unary relations, but the prepositions expressing spatial relations, AT, BETWEEN, FOLLOW, FROM, IN, ON, UNDER, for which we must assume a prepositional subject and a prepositional object, the comparative ER, the negative LACK, and a few conceptual relation markers, quite often expressed by cases, such as CAUSE, FOR, HAS, INS, and PARTOF. Pride of place goes to ISA, essential for taxonomic organization, and MARK, denoting the relation between the two parts of the Saussurian sign.

Acknowledgments

I'm grateful to Geoff Pullum and Bob Levine for penetrating comments on an earlier draft.

[7] https://en.wikipedia.org/wiki/Overcompleteness

References

Ács, Judit, Dávid Márk Nemeskey & Gábor Recski. 2019. Building word embeddings from dictionary definitions. In Beáta Gyuris, Katalin Mády & Gábor Recski (eds.), *K + K = 120: Papers dedicated to László Kálmán and András Kornai on the occasion of their 60th birthdays*, 1–18. Budapest, Hungary: Research Institute for Linguistics, Hungarian Academy of Sciences (RIL HAS).

Angluin, Dana. 1980. Inductive inference of formal languages from positive data. *Information and Control* 21. 46–62.

Baldridge, Jason. 2002. *Lexically specified derivational control in combinatory categorial grammar*. Univ. of Edinburgh. (Doctoral dissertation).

Blackburn, Patrick & Johan Bos. 2015. *Representation and inference for natural language*. Chicago, IL: University of Chicago Press.

Bolinger, Dwight. 1965. The atomization of meaning. *Language* 41(4). 555–573.

Brachman, Ronald J. & Hector Levesque. 2004. *Knowledge Representation and reasoning*. Los Altos, CA: Morgan Kaufmann Elsevier.

Bruening, Benjamin. 2018. The lexicalist hypothesis: Both wrong and superfluous. *Language* 94(1). 1–42. DOI: 10.1353/lan.2018.0000.

Cardeñosa, Jesús, Alexander Gelbukh & Edmundo Tovar (eds.). 2005. *Universal networking language: Advances in theory and applications* (Research on Computing Science 12). Mexico City, Mexico: Centre for Computing Research of IPN.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, Noam. 1970. Remarks on nominalization. In Roderick Jacobs & Peter Rosenbaum (eds.), *Readings in English transformational grammar*, 184–221. Waltham, MA: Blaisdell.

Chomsky, Noam & Howard Lasnik. 1993. Principles and parameters theory. In Joachim Jacobs (ed.), *Syntax: An international handbook of contemporary research*, vol. 1, 505–569. Berlin: de Gruyter.

Collobert, Ronan, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu & Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research (JMLR)* 12. 2493–2537.

Curry, Haskell B. & Robert Feys. 1958. *Combinatory logic I*. Amsterdam: North-Holland.

Fodor, Jerry. 1970. Three reasons for not deriving "kill" from "cause to die". *Linguistic Inquiry* 1(4). 429–438.

Fodor, Jerry A. 1998. *Concepts*. Oxford, UK: Clarendon Press.

Fokker, Jeroen. 1989. *The systematic construction of a one-combinator basis*. Tech. rep. RUU-CS-89-14. Department of Computer Science, University of Utrecht.

Gage, Philip. 1994. A new algorithm for data compression. *The C. Users Journal* 12(2). 23–38.

Goddard, Cliff. 2002. The search for the shared semantic core of all languages. In Cliff Goddard & Anna Wierzbicka (eds.), *Meaning and universal grammar: Theory and empirical findings*, vol. 1, 5–40. Amsterdam: Benjamins.

Jacobson, Pauline. 1999. Towards a variable-free semantics. *Linguistics and Philosophy* 22. 117–184.

Joshi, Aravind K., K. Vijay-Shanker & David J. Weir. 1991. The convergence of mildly context-sensitive grammar formalisms. In Stuart M. Shieber, Peter Sells & Thomas Wasow (eds.), *Foundational issues in Natural Language Processing*, 31–81. Bolton, MA: MIT Press.

Katz, Jerrold & Jerry A. Fodor. 1963. The structure of a semantic theory. *Language* 39. 170–210.

Kiparsky, Paul. 1982. Word-formation and the lexicon. In Frances Ingemann (ed.), *Proceedings of the Mid-America Linguistics Conference*, 3–29. Lawrence, Kansas.

Knuth, Donald E. 1968. Semantics of context-free languages. *Mathematical Systems Theory* 2. 127–145.

Kornai, András. 1988. Compositionality, of, word-formation. *Acta Linguistica* 38. 118–130.

Kornai, András. 2010. The algebra of lexical semantics. In Christian Ebert, Gerhard Jäger & Jens Michaelis (eds.), *Proceedings of the 11th Mathematics of Language Workshop* (LNAI 6149), 174–199. Springer. DOI: 10.5555/1886644.1886658.

Kornai, András. 2012. Eliminating ditransitives. In Philippe de Groote & Mark-Jan Nederhof (eds.), *Revised and Selected Papers from the 15th and 16th Formal Grammar Conferences* (LNCS 7395), 243–261. Berlin: Springer.

Kornai, András. 2021. Vocabulary: Common or basic? *Frontiers in Psychology* 12. 1–7. DOI: 10.3389/fpsyg.2021.730112.

Kornai, András. 2023. *Vector semantics*. Berlin: Springer. DOI: 10.1007/978-981-19-5607-2.

LeCun, Yann, Yoshua Bengio & Geoffrey Hinton. 2015. Deep learning. *Nature* 521. 436–444.

Lewis, David. 1970. General semantics. *Synthese* 22(1). 18–67.

Mikolov, Tomas, Wen-tau Yih & 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)*, 746–751. Atlanta, Georgia: Association for Computational Linguistics.

Montague, Richard. 1970. Universal grammar. *Theoria* 36. 373–398.

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In Richmond Thomason (ed.), *Formal philosophy*, 247–270. New Haven, CT: Yale University Press.

Müller, Stefan. 2018. The end of lexicalism as we know it? *Language* 94. 54–66.

Partee, Barbara. 1979. Semantics: Mathematics or psychology? In Rainer Bäuerl, Urs Egli & Arnim von Stechow (eds.), *Semantics from different points of view*, 1–14. Berlin: Springer.

Partee, Barbara. 2013. *Changing perspectives on the "mathematics or psychology" question.*

Procter, Paul. 1978. *Longman dictionary of contemporary English.* 1st edn. Harlow, UK: Longman.

Recski, Gábor. 2016. Building concept graphs from monolingual dictionary entries. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).* Portorož, Slovenia: European Language Resources Association (ELRA).

Schönfinkel, Moses. 1924. On the building blocks of mathematical logic. *Mathematische Annalen* 3–4. 305–316.

Schütze, Hinrich. 1993. Word space. In Stephen Jose Hanson, Jack D. Cowan & C. Lee Giles (eds.), *Advances in neural information processing systems 5*, 895–902. Burlington, MA: Morgan Kaufmann.

Steedman, Mark. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory* 5. 403–439.

Sutskever, Ilya, Oriol Vinyals & Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberger (eds.), *NIPS'14: Proceedings of the 27th international conference on Neural Information Processing Systems*, vol. 2, 3104–3112. Cambridge, MA. http://arxiv.org/abs/1409.3215 (16 March, 2024).

Szabolcsi, Anna. 1987. Bound variables in syntax: Are there any? In Jeroen Gronendijk, Martin Stokhof & Frank Veltman (eds.), *Proceedings of the 6th Amsterdam colloquium*, 331–351. Amsterdam: Institute for Language, Logic, & Information.

Tarski, Alfred. 1956. The concept of truth in formalized languages. In Alfred Tarski (ed.), *Logic, semantics, metamathematics*, 152–278. Oxford, UK: Clarendon Press.

van Heijenoort, Jean (ed.). 1967. *From Frege to Gödel.* Cambridge, MA: Harvard University Press.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna Wallach, Rob Fergus, S.V.N. Vishwanathan & Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30*, 5998–6008. Boston, MA: Curran Associates, Inc. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf (16 March, 2024).

Wells, Rulon S., III. 1947. Immediate constituents. *Language* 23. 321–343.

Wierzbicka, Anna. 1992. *Semantics, culture, and cognition: Universal human concepts in culture-specific configurations*. Oxford: Oxford University Press.

Wierzbicka, Anna. 1996. *Semantics: Primes and universals*, vol. 26. Oxford: Oxford University Press.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes & Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. http://arxiv.org/abs/1609.08144 (16 March, 2024).