1

Foundations of non-compositionality

Contents

For the past half century, linguistic semantics was dominated by issues of compositionality to such an extent that the meaning of the atomic units (which were generally assumed to be words or their stems) received scant attention. Here we will put word meaning front and center, and base the entire plan of the book on beginning with the lowest meaningful units, morphemes, and building upward. In 1.1 we set the stage by considering the three major approaches to semantics that can be distinguished by their formal apparatus: formulaic, geometric, and algebraic. In 1.2 we summarize some of the lexicographic principles that we will apply throughout: universality, reductivity, and keeping the lexicon free of encyclopedic knowledge. In 1.3 we describe the formulaic theory of lexical meaning. This is linked to the geometric theory in 1.4, and to the algebraic theory in 1.5. The links between the algebraic and the geometric theory are discussed in 1.6, where we investigate the possibility of a meta-formalism that could link all three approaches together.

1.1 Background

The **formulaic** (logic-based) theory of semantics (S19:3.7), Montague Grammar (MG) and its lineal descendants such as Discourse Representation Theory and Dynamic Semantics reigned supreme in linguistic semantics until the 21st century in spite of its well known failings because it was, and in some respects still is, the only game in town: the alternative 'cognitive' theory went largely unformalized, and was deemed 'markerese' (Lewis, 1970) by the logic-based school. Here we will attempt to formalize many, though

A. Kornai, *Vector Semantics*, Cognitive Technologies, https://doi.org/10.1007/978-981-19-5607-2_1

by no means all, insights of the cognitive theory, an undertaking made all the more necessary by the fact that MG has little to offer on the nature of atomic units (Zimmermann, 1999).

Starting perhaps with (Schütze, 1993; Schütze, 1998) and propelled to universal success by (Collobert and Weston, 2008; Collobert et al., 2011) an entirely new, **geometric** theory, mapping meanings to vectors in low-dimensional Euclidean space, became standard in computational linguistics (S19:2.7 Example 2.3 et seqq). Subjects central to semantics such as compositionality, or the relation of syntactic to semantic representations, hitherto discussed entirely in a logic-based framework, became the focus of attention (Allauzen et al., 2013) for the geometric theory, but there is still no widely accepted solution to these problems. One unforeseen development of the geometric theory was that morphology, syntax, and semantics are to some extent located in different layers of the multilayer models that take word vectors as input (Belinkov et al., 2017b; Belinkov et al., 2017a) but 'probing' the models is still an art, see (Karpathy, Johnson, and Fei-Fei, 2015; Greff et al., 2015) for some of the early work in this direction, and (Clark et al., 2019; Hewitt and Manning, 2019) for more recent work on contextual embeddings.

At the same time, the **algebraic** theory of semantics (S19:Def 4.5 et seqq) explored in Artificial Intelligence since the 1960s (Quillian, 1969; Minsky, 1975; Sondheimer, Weischedel, and Bobrow, 1984), which used (hyper)graphs for representing the meaning of sentences and larger units, was given new impetus by Google's efforts to build a large repository of real-world knowledge by finding named entities in text and anchoring these to a large external knowledge base, the KnowledgeGraph, which currently has over 500m entities linked by 170b relations or 'facts' (Pereira, 2012). More linguistically motivated algebraic theories (Kornai, 2010a; Abend and Rappoport, 2013; Banarescu et al., 2013), coupled with a renewed interest in dependency parsing (Nivre et al., 2016), are contributing to a larger reappraisal of the role of background knowledge and the use of hypergraphs in semantics (Koller and Kuhlmann, 2011).

Through this book, we will try to link these three approaches, giving mathematical form to the belief that they are just the trunk, leg, and tail of the same elephant. This is not to say that these are 'notational variants' (Johnson, 2015), to the contrary, each of them make predictions that the others lack. A better analogy would be the algebraic (matrix) and the geometrical (transformation) view of linear algebra: both are equally valid, but they are not equally useful in every situation.

One word of caution is in order: the formulas we will study in 1.3 are not the formulas of higher order intensional logic familiar to students of MG, but rather the basic building blocks of a much simpler proto-logic, well below first order language in complexity. The graphs that we will start studying in 1.5 are hypergraphs, very similar to the notational devices of cognitive linguistics, DG, LFG, HPSG and those of AI, but not letter-identical to any of the broad variety of earlier proposals. Only the geometry is the same n-dimensional Euclidean geometry that everyone else is using, but even here there will be some twists, see 1.4.

1.2 Lexicographic principles

Universality `4lang` is a concept dictionary, intended to be universal in a sense made more precise below. To take the first tentative steps towards language-independence, the system was set up with bindings in four languages, representative samples of the major language families spoken in Europe: Germanic (English), Slavic (Polish), Romance (Latin), and Finno-Ugric (Hungarian). In Version 1, automatically created bindings exist in over 40 languages (Ács, Pajkossy, and Kornai, 2013), but the user should keep in mind that these bindings provide only rough semantic correspondence to the intended concept. In the current Version 2 (see 9.5) two Oriental languages, Japanese and Chinese, were added manually by László Cseresnyési and Huba Bartos respectively, and further automatic binding were created (Hamerlik, 2022).

The experience of parallel development of `4lang` in four languages reinforces a simple point that lexicographers have always considered self-evident: words or word senses don't match up across languages, not even in the case of these four languages that share a common European cultural/civilizational background. It's not just that some concepts are simply missing in some languages (a frequent cause of borrowing), but the whole conceptual space (see 1.4) can be partitioned differently.

For example, English tends to make no distinction between verbs that describe actions that affect their subjects and their objects the same way: compare *John turns, John bends* to *John turns the lever, John bends the pipe*. In Polish, we need a reflexive object pronoun *się* 'self' to express the fact that it is John who is turning/bending in the first case. The semantics is identical, yet in English *??John turns/bends himself* would sound strange. In Hungarian, we must use different verbs derived from the same root: 'turn self' is *ford-ul* whereas 'turn something' is *ford-ít*, and similarly for *haj-ol* 'bend self' and *hajl-ít* 'bend something', akin to Latin *versor/verso, flector/flecto*, but Latin also offers the option of using a pronoun *me flecto/verso*.

Where does this leave us in regards to the lofty goal of universality? At one extreme, we find the strong Sapir-Whorf hypothesis that language determines thought. This would mean that a speaker of English cannot share the concept of bending with a speaker of Hungarian, being restricted to one word for two different kinds of situations that Hungarian has two different words for. At the other extreme, we find the methodology followed here: we resort to highly abstract units (core lexemes) which we assume to be shared across languages, but permit larger units to be built from these in ways that differ from language to language. Here the key notions we must countenance include *self*, which is defined as `=pat[=agt], =agt[=pat]` (see also 3.3), and *bend*, which we take to be basic in the intransitive form, see 2.4. We turn to the issue of how in general transitives can be defined by their objectless counterparts in 3.1.

`self`

How formulas such as these are to be created, manipulated, and understood will be discussed in 1.3, here we begin with high-level formatting. The main `4lang` file is divided into 11 tab-separated fields, of which the last is reserved for comments (these

begin with a percent sign). A typical entry, written as one line in the file but here in the text generally broken up in two for legibility, would be

```
water víz aqua woda mizu 水 shui3 水 2622 u N
      liquid, lack colour, lack taste, lack smell, life need
```

As can be seen, the first four columns are the 4 original language bindings given in EN HU LA PL order. In Version 1, all extended Latin characters were replaced by their base plus a number, e.g. o3 for ő, o2 for ö, and o1 for ó. This was to keep the behavior of standard unix utilities like `grep` constant across platforms (scripts for conversion to/from utf8 were available). In Version 2, two new columns are added after the fourth for JA ZH (see 9.5), and utf8-encoded accented characters are used throughout. The seventh column (in V1, the fifth) is a unique number per concept, most important when the English bindings coincide:

```
cook főz     coquo  gotować 825  V
     =agt make <food>, ins_ heat
cook szakács coquus kucharz 2152 N
     person, <profession>, make food
```

The eighth (in V1, sixth) column is an estimate of reducibility status and can take only four values: `p` means primitive, an entry that seems impossible to reduce to other entries.

wh An example would be the question morpheme *wh,* here given as `wh ki/mi/hogy quo kto/co/jak 3636 p G wh`. Note that the definiendum (column 1) appears in the definiens (column 10), making the irreducibility of this entry evident. At the other end we find entries marked by `e`, which means eliminable. An example would be

three *three* `three három tres trzy 2970 e A number, follow two`. In between we find entries marked by `c`, which are candidates for core vocabulary: and exam-

see ple would be *see* `see lát video widzieć 1476 c V perceive, ins_ eye`; and `u`, unknown reducibility status.

The ninth (in V1, seventh) column is a rough lexical category symbol, see 2.1 for further discussion. Our main subject here is the 10th (in V1, eighth) column, which gives the `4lang` definition. We defer the formal syntax of definitions to 1.3, after we discussed some further lexicographic principles, and use the opportunity to introduce some of the notation informally first. Many technical devices such as `=agt, =pat, wh, gen, ...` make their first appearance here, but will be fully explained only in subsequent chapters. Very often, we will have reason to present lexical entries in an abbreviated form, showing only the headword and the definition (with the index, reducibility, and lexical category shown or suppressed as needed):

```
bend 975 e V has form[change], after(lack straight/563)
```

drunk Where such abbreviated entries appear in running text, as *drunk* here, `drunk ittas potus pijany 1165 c A quality, person has quality, alcohol cause_, lack control` the headword is highlighted on the margin. For human readability, the concept number is omitted whenever the English binding is unique, so we have `person` in the above definition rather than `person/2185`, but we would spell

out man/659 'homo' to disambiguate from man férfi vir mężczyzna 744 man e N person, male. In running text we generally omit the Japanese and Chinese equivalents for ease of typesetting.

Generally, we take examples from V2/700.tsv, but on occasion we find it necessary to go outside the 700.tsv set to illustrate a point, and (very rarely) even outside the V1 file.

Reductivity In many ways, 4lang is a logical outgrowth of modern, computationally oriented lexicographic work beginning with Collins-COBUILD (Sinclair, 1987), the Longman Dictionary of Contemporary English (LDOCE) (Boguraev and Briscoe, 1989), WordNet (Miller, 1995), FrameNet (Fillmore and Atkins, 1998), and VerbNet (Kipper, Dang, and Palmer, 2000). The main motivation for systematic reductivity was spelled out in (Kornai, 2010a) as follows:

"In creating a formal model of the lexicon the key difficulty is the circularity of traditional dictionary definitions – the first English dictionary, Cawdrey, 1604 already defines *heathen* as gentile and *gentile* as heathen. The problem has already been 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 Maevius; 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."

The key step in minimizing circularity was taken in LDOCE, where a small (about 2,200 words) defining vocabulary called LDV, Longman Defining Vocabulary was created, and strictly adhered to in the definitions with one trivial exception: words that often appear in definitions (e.g. the word *planet* is common to the definition of Mercury, Mars, Venus, ...) can be used as long as their definition is strictly in terms of the LDV. Since *planet* is defined 'a large body in space that moves around a star' and *Jupiter* is defined as 'the largest planet of the Sun' it is easy to substitute one definition in the other to obtain for Jupiter the definition 'the largest body in space that moves around the Sun'.

4lang generalizes this process, starting with a core list of defining elements, defining a larger set in terms of these, a yet larger set in terms of these, and so on until the entire vocabulary is in scope. As a practical matter we started from the opposite direction, with a seed list of approximately 3,500 entries composed of the LDV (2,200 entries), the most frequent 2,000 words according to the Google unigram count (Brants and Franz, 2006) and the BNC (Burnard and Aston, 1998), as well as the most frequent 2,000 words from Polish (Halácsy et al., 2008) and Hungarian (Kornai et al., 2006). Since Latin is one of the four languages supported by 4lang, we added the classic Diederich, 1939 list and Whitney, 1885.

Based on these 3,500 words, we reduced the defining vocabulary by means of a heuristic graph search algorithm (Ács, Pajkossy, and Kornai, 2013) that eliminated all

words that were definable in terms of the remaining ones. The end-stage is a vocabulary with the *uroboros property*, i.e. one that is minimal wrt this elimination process. This list (1,200 words, not counting different senses with multiplicity) was published as Appendix 4.8 of S19 and was used in several subsequent studies including (Nemeskey and Kornai, 2018). (The last remnant of the fact that we started with over 3k words is that numbers in the 5th column are still in the 1-3,999 range, as we decided against renumbering the set.) This '1200' list is part of Release V1 of `4lang` on `github`, and has bindings to Release 2.5 of Concepticon (List, Cysouw, and Forkel, 2016).

By now (Release V2), this list has shrunk considerably, because improvements in the heuristic search algorithm (see Ács, Nemeskey, and Recski (2019) and uroboros.py) and a systematic tightening of `4lang` definitions by means of def_ply_parser.py made further reductions possible. The name of the '700' list is somewhat aspirational (the Version 2 file has 739 words in 776 senses) but we believe the majority of the 359 senses marked `e` are indeed eliminable, and the eventual uroboros core (`p` and `c` entries) will be below 200 senses. With every substitution, we decrease the sparseness of the system. In the limiting case, with a truly uroboros set of maybe 120 elements, we expect the definitions to become much longer and more convoluted. This phenomenon is very observable in the Natural Semantic Metalanguage (NSM) of (Wierzbicka, 1992; Wierzbicka, 1996; Goddard, 2002), which in many ways served as an inspiration for `4lang`.

The two theories, while clearly motivated by the same goal of searching for a common universal semantic core, differ in two main respects. First, by using English definitions rather than a formal language, NSM brings many subtle syntactic problems in tow (see Kornai (2021) for a discussion of some of these). Second, NSM is missing the reduction algorithm that `4lang` provides. In brief, for any sense of any word we can look up the definition in a dictionary, convert this definition to a `4lang` graph that contains only words from the LDV, and for any LDV word we can follow its reduction to V1, and further, to V2 terms. Preliminary work on V3 suggests that it will still have about twice as many primitives than the 63 primes currently used in NSM.

Indeed, just by looking at an ordinary English word such as *random* (see S19:Ex.° 4.21) we are at a complete loss how to define it in terms of the NSM system beyond the vague sense that the prime MAYBE may be involved. With `4lang` , we start with 'aimlessly, without any plan' (LDOCE). We know (see 6.4) that *-ly* is semantically empty, and that *-less* is to be translated as `lack stem_`. Further, from 4.5 we know that *any* is defined as `<one>, =agt is_a`, so that *any plan* is defined as `<one> plan`. Since here neither the presence of *one* not its absence (see Rule 6 of 1.6 that the 〈〉 signify optionality) adds information, we have `lack aim, lack plan`. At this point, all defining terms are there in the (V2) core vocabulary, we are done.

Perhaps someone with deeper familiarity with NSM could concoct a definition using only the primes, though it appears that none of the 63 primes except WANT seem related to aims, goals, plans, or any notion of purposive action. To the extent that Gewirth, 1978 includes 'capability for voluntary purposive action' as part of the definition of what defines a human as a 'prospective purposive agent', this lack of defining NSM terms is

highly problematic, placing the people whose language is describable in purely NSM terms on the level of infants with clear wants but no agency to plan. But our issue is a more general one: it is not this particular example that throws down the gauntlet, it is the lack of a general reduction algorithm.

In contrast, since at any stage the uroboros vocabulary is obtained by systematic reduction of a superset of the LDV, it is still guaranteed that every sense of every word listed in LDOCE (over 82k entries) are definable in terms of these. Since the defining vocabularies of even larger dictionaries such as Webster's 3rd (Gove, 1961) are generally included in LDOCE, we have every reason to believe that the entire vocabulary of English, indeed the entire vocabulary of any language, is still definable in terms of the uroboros concepts.

Redefinition generally requires more than string substitution. Take again PLANET, a word LDOCE uses in the same manner as NSM uses semantic molecules, and defines as 'a large body in space that moves around a star'. If we mechaically substitute this in the definition of *Jupiter*, 'the largest __ of the Sun' we obtain 'the largest a large body in space that moves around a star of the Sun'. It takes a great deal of sophistication for the substitution algorithm to realize that *a large* is subsumed by *the largest* or that *a star* is instantiated by *the Sun*. People perform these operations with ease, without conscious effort, but for now we lack parsers of the requisite syntactic and semantic sophistication to do this automatically. Part of our goal with the strict definition syntax that replaces English syntax on the right-hand side (rhs) of definitions is to study the mechanisms required by an automated parser for doing this, see Chapter 2.

Encyclopedic knowledge In light of the foregoing, the overall principle of keeping linguistic (lexicographic) knowledge separate from real-world (encyclopedic) knowledge is already well motivated. First, universality demands a common lexical base, whereas it is evident that real-world knowledge differs from culture to culture, and thus from language to language – in the limiting case, it differs within the same culture and the same language from period to period. Since the completion of the Human Genome Project in 2003, our knowledge of genes and genomes have exploded: at the time of this writing the Cancer Genome Atlas holds over 2.5 petabytes of data, yet the English language is pretty much the same as it was 20 years ago. The need to keep two so differently growing sources of knowledge separate is obvious.

Second, reductivity demands that knowledge be expressed in words. This may have made sense for biology two hundred years ago (indeed, biological taxa are traditionally defined by means of the same Aristotelian technology of *genus* and *differentia specifica* (S19:2.7) that we rely on), but clearly makes vanishingly little sense in chemistry, physics, and elsewhere in the sciences where knowledge is often expressed by a completely different language, that of mathematics. As we shall see in Chapter 8, trivia like *Who won the World Series in 1967?* are within scope for the `4lang` Knowledge Representation (KR) system. But core scientific statements, from the Peano Axioms (see 3.4) to Gauss' Law of Magnetism, $\nabla \cdot \mathbf{B} = 0$, are out of scope.

How are the lines to be drawn between lexical and encyclopedic, verbally expressible and mathematics-intense knowledge? This is a much debated isse (see Peeters, 2000 for a broad range of views) and `4lang` clearly falls at the Aristotelian end of the dualist/monist spectrum introduced in Cabrera, 2001. We begin our discussion with a simple item. The first edition of LDOCE (Procter, 1978) defines *caramel* as 'burnt sugar used for giving food a special taste and colour'. In `4lang` this could be recast as

```
caramel sugar[burnt], cause_ {food has {taste[special],
colour[specal], <taste[sweet]>, <colour[brown]>}}
```

where quite a bit of the syntax is implicit, such as the fact that `caramel` is the subject of `cause_`, see Section 1.3, and we sneaked in some real world knowledge that the special taste is (in the default case) sweet, and the special color is brown.

`special` As the preceding make clear, we could track further *special* (defined in `4lang` as `lack common`), or *food*, or *burnt*, or any term, but here we will concentrate on *sugar* 'a sweet white or brown substance that is obtained from plants and used to make food and drinks sweet'. Remarkably, this definition would also cover xylitol ($CH_2OH(CHOH)_3CH_2OH$) or stevia ($C_{20}H_{30}O_3$) which are used increasingly as replacements for common household sugar ($C_6H_{12}O_6$).

This is not to say that the editors should have been aware in 1978 that a few decades later their definition will no longer be specific enough to distinguish sugar from other sweeteners. Yet the clause 'obtained from plants' is indicative of awareness about saccharine ($C_7H_5NO_3S$) which is also sweet, but is not obtained from plants.

`4lang` takes the line that encyclopedic knowledge has no place in the lexicon. Instead of worrying about how to write clever definitions that will distinguish sugar not just from saccharine but also from xylitol, stevia, and whatever new sweeteners the future may bring, it embraces simplicity and provides definitions like the following:

```
rottweiler dog
greyhound dog
```

This means that we fail to fully characterize the competent adult speaker's ability to use the word *rottweiler* or *greyhound*, but this does not seem to be a critical point of language use, especially as many adult speakers seem to get along just fine without a detailed knowledge of dog breeds. To quote Kornai, 2010a:

> So far we 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 the 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 in a set **E**

Today, we use Wikipedia for our encyclopedia, and denote pointers to it by a prefixed @ sign, see Section 1.3. Our definitions are

```
sugar cukor saccharum cukier  440 N
      material, sweet, <white>, in food, in drink
sweet e1des dulcis    sl1odki 495 A
      taste, good, pleasant, sugar has taste, honey has taste
```

Instead of sophisticated scientific taxonomies, `4lang` supports a naive world-view (Hayes, 1979; Dahlgren, 1988; Gordon and Hobbs, 2017). We learn that *sugar* is sweet, and *sweet* is_a taste – the system actually makes no distinction between predicative (is) and attributive (is_a) usage. We learn that sugar is to be found in food and drink, but not where exactly. In general, the lexicon is restricted to the core premisses of the naive theory. When in doubt about a particular piece of knowledge, the overriding principle is not whether it is true. In fact the lexicon preserves many factually untrue propositions, see e.g. the discussion in 3.1 of how the heart is the seat of love. The key issue is whether a meaning component is learnable by the methods we suggest in 5.3 and, since these methods rely on embodiment, a good methodological guideline is 'when in doubt, assign it to the encyclopedia'.

One place where the naive view is very evident is the treatment of high-level abstractions. For example, the definition of *color* has nothing to do with photons, frequency ranges in the electromagnetic spectrum, or anything of the sort – what we have instead is `sensation, light/739, red is_a, green is_a, blue is_a` and `colour` when we turn to e.g. *red* we find `colour, warm, fire has colour, blood` `red has colour`. Another field where we support only a naive theory is grammar, see 2.5.

As with *sugar* and *sweet*, we posit something approaching a mutual defining relation between *red* and *blood*, but this is not entirely like Titius and Caius sending you further on: actually *blood* gets eliminated early in the uroboros search as we iteratively narrow the defining set, while *red* stays on. Eventually, we have to have some primitives, and we consider *red*, a Stage II color in the (Berlin and Kay, 1969) hierarchy, a very reasonable candidate for a cross-linguistic primitive. In fact, `uroboros.py` is of the same opinion (in no run does *red* get eliminated, hence the marking `c` (core) in column 7).

So far, we have discussed the fact that separating the encyclopedia from the lexicon leaves us with a clear class of lexical entries, exemplified so far by colors and flavors, where the commonly understood meaning is anchored entirely outside the lexicon. There are also cases where this anchoring is partial, such as the suffix *-shaped*. The meaning of *guitar-shaped, C-shaped, U-shaped, ...* is clearly compositional, and relies on cultural primitives such as *guitar, C, U, ...* that will remain at least partially outside the lexicon. According to Rosch (1975), lexical entries may contain pointers to non-verbal material, not just primary perceptions like color or taste, but also prototypical images. We can say that *guitar* is a stringed musical instrument, or that C and U are letters of the alphabet,

and this is certainly part of the meaning of these words, but it is precisely for the image aspect highlighted by *-shaped* that words fail us. Again anticipating notation that we will fully define only in 2.2, we can define *guitar-shaped* as `has shape, guitar has shape` and in general

-shaped `has shape, stem_ has shape, "_-shaped" mark_ stem_`

and leave it to the general unification mechanism we will discuss in 1.5 and 8.3 to guarantee that it is the same shape that the stem and the denotation of the compound adjective will share.

1.3 The syntax of definitions

Here we discuss, somewhat informally, the major steps in the formal analysis of `4lang` definitions. A standard lex-yacc parser, `def_ply_parser.py` is available on github. The syntax is geared towards *human* readability, so that plaintext lexical entries where the definiens (usually a complex formula) is given after the definiendum (usually an atomic formula) are reasonably understandable to those working with `4lang`. In 1.5 we will discuss in more detail the omission of overt subjects and objects, an *anuvṛtti*-like device, that greatly enhances readability. Here we present a simple example:

```
April   month, follow march/1563, may/1560 follow
bank    institution, money in
```

The intended graph for April will have a 0 link from the definiendum to month, a 1 link to march/1563 and a 2 link to may/1560. Strictly speaking, anuvṛtti removes redundancies across stanzas (sūtras) whereas our method operates within the same stanza across the left- and right-hand sides, but the functional goal of compression is the same.

Often, what is at the other side of the binary is unspecified, in which case we use the `gen` symbol "plugged up". Examples:

```
vegetable  plant, gen eat
sign gen perceive, information, show, has meaning
```

Thus, *vegetable* is a plant that someone (not specified who) can eat (it is the object of eating, subject unspecified), and *sign* is_a information, is the object of perception, is_a show (nominal, something that is or can be shown) and has meaning.

Starting with 'disambiguated language' (S19:3.7), semanticists generally give themselves the freedom to depart from many syntactic details of natural language. For example Cresswell, 1976 uses

> λ-deep structures that look as though they could become English sentences with a bit of tinkering. In this particular work I am concerned more with the underlying semantic structure than with the tinkering.

By aiming at a universal semantic representation we are practically forced to follow the same method, since the details of the 'tinkering' change from language to language, but we try to be very explicit about this, using the `mark_` primitive that connects words to their meanings (see 2.5). One particular piece of tinkering both Cresswell and I are guilty of is permitting semantics to cross-cut syntax and morphology, such as by reliance on a comparative morpheme `er_` (called **er than** in Cresswell, 1976) but really, what can we do? The comparative *-er* is a morpheme used in about 5% of the definitions, and there is no reason to assume it means different things following different adjectival stems.

`mark_`

`er_`

Coordination A `4lang` definition always contains one or more clauses (hypergraph nodes, see 1.5) in a comma-separated list. The first of these is distinguished as the *head* (related to, but not exactly the same as the *root* in dependency graphs). In 1.5 the top-level nodes will be interpreted so as to include graph edges with label 0 running from the definiendum to the definiens. The simplest definitions are therefore of the form x, where x is a single atomic clause. Example

```
aim ce1l finis cel 363 N
    purpose
```

that is, the word *aim* is defined as *purpose*. Somewhat more complex definitions are given by a comma-separated list:

```
board lap   tabula tablica 456 N
      artefact, long, flat
boat  hajo1 navis  l1o1dz1 976 N
      ship, small, open/1814
```

(The number following the '/', if present, serves to disambiguate among various definitions, in this case adjectival *open* 'apertus' from verbal *open* 'aperio'. These numbers are in column 7 of the `4lang` file.) In 1.4 we will discuss the appropriate vector space semantics for coordination of defining properties in more detail, but as a first approximation it is best to think of these as strictly intersective.

Subordination Deefinitions can have dependent clauses e.g. *protect* `=agt cause_ {=pat[safe]}` 'what *X protects Y* means is that *X* causes *Y* to be safe'. Of particular interes are relative clauses, which are handled by unification, without an overt *that* morpheme, e.g. 'red is the color that blood has' is expressed by a conjunction `red is_a color, blood has color` where the two tokens of `color` are automatically unified, see 8.3.

`protect`

External pointers Sometimes (42 cases in the 1,200 concepts published in S19:4.8) a concept doesn't fully belong in the lexicon, but rather in the encyclopedia. In the formal language defined here, such *external pointers* are marked by a prefixed @. Examples:

```
Africa land, @Africa
London city, @London
Muhammad  man/744, @Muhammad
U letter/278, @U
```

These examples, typically less than 5% of any dictionary, are but a tiny sample from millions of person names, geographic locations, and various other proper names. We will discuss such 'named entities' in greater detail in Chapter 8.

Subjects and objects In earlier work, staring with Kornai, 2010a, we linked `4lang` to the kind of graphical knowledge representation schemas commonly used in AI. Such (hyper)graphs have (hyper)edges roughly corresponding to concepts, and *links* connecting the concepts. `4lang` has only three kinds of links marked 0,1, and 2.

0 links cover both predicative *is*, cf. the definition of *sugar* as `sweet, in food, in drink` above, and subsumptive `is_a` which obtains both between hyponyms and hypernyms and between instances and classes. 1 links cover subjects, and 2 links cover objects. We will discuss hypergraphs further in 1.5 and the link inventory in 2.3.

In addition to 0 links, definitions often explain the definiendum in terms of it being the subject or object of some binary relation. In some cases, these relations are highly grammatical, as `for_`, known as "the dative of purpose":

```
handle 834 u N part_of object, for_ hold(object in hand)
```

while in other cases the relation has a meaning that is sufficiently close to the ordinary English meaning that we make no distinction. An example of the latter would be `for` used to mark the price in an exchange as in *He sold the book for $10*, or `has` used to mark possession as in *John has a new dog*. When we use a word in the sense of grammar, we mark this with an underscore, as in `for 2824` versus `for_ 2782`. We defer discussing the distinction between "ordinary" and "grammatical" terms to 2.5, but note here that the English syntax of such terms can be very different from their `4lang` syntax. Compare `-er 14` which is a suffix attaching to a single argument, the stem (which makes it a unary relation), to `er_ 3272` which has two obligatory arguments (making it a binary relation).

Direct predication In a formula `A[B]` means that there is a 0-link from A to B. This is used only to make the notation more compact. The notation B(A) means the same thing, it is also just syntactic sugar. Both brackets and parens can contain full subgraphs.

```
tree plant, has material[wood], has trunk/2759, has crown
```

That trees also have roots is not part of the definition, not because it is inessential, but because trees are defined as plants, and plants all have roots, so the property of having roots will be inherited.

Defaults In principle, all definitional elements are strict (can be defeased only under exceptional circumstances) but time and again we find it expedient to collapse strongly related entries by means of defaults that appear in angled brackets.

```
ride travel, =agt on <horse>, ins_ <horse>
```

These days, a more generalized *ride* is common (*riding the bus, catching a ride, ...* so the definition `travel` should be sufficient as is. The historically prevalent mode of traveling, on horseback, is kept as a default. Note that these two entries often get translated by different words: for example Hungarian distinguishes *utazik* 'travel' and *lovagol* 'rides a horse', a verb that cannot appear with an object or instrument the same way as English *ride a bike* can. Defaults are further discussed in 6.4.

Agents, patients The relationship between horseback riding (which is, as exemplified above, just a form of traveling) and its defining element, the horse, is indirect. The horse is neither the subject, not the object of travel. Rather, it is the rider who is the subject of the definiendum and the definiens alike, corresponding to a graph node that has a 1 arrow leading to it from both. This node is labeled by `=agt`, so when we wish to express the semantic fact that Hungarian *lovagol* means 'travel on a horse' we write

```
lovagol travel, =agt on horse
```

Note that the horse is not optional for this verb in Hungarian: it is syntactically forbidden (*lovagol* is intransitive) and semantically obligatory. (Morphologically it is already expressed, as the verb is derived from the stem *ló* 'horse' though this derivation is not by productive suffixation.) Remarkably, when the object is_a horse (e.g. a colt is a young horse, or a specific horse like Kincsem) we can still use *lovagol* as in *János a csikót lovagolta meg* or *Elijah Madden Kincsemet lovagolta*.

For the patient role, consider the word *know*, defined as 'has information about'. For this to work, the expression `x know y` has to be equivalent to `x has information about y` i.e. we need to express the fact that the subject of has is the same as the subject of *know* (this is done by the `=agt` placeholder) and that the object of `about` is the same as the object of knowing – this will be done by the `=pat` placeholder.

As discussed in Kornai, 2012 in greater detail, these two placeholders (or *thematic roles*, as they are often called) will be sufficient, but given the extraordinary importance of these notions in grammatical theory, we will discuss the strongly related notions of thematic relations, deep cases, and *kārakas* in 2.4 further.

More complex notation When using [] or (), both can contain not just single nodes but entire subgraphs. For subgraphs we also use { }, see 1.6.

```
stock re1szve1ny syngrapha papier_wartos1ciowy 3626 N
      document, company has, {person has stock} prove
      {person has part_of company}
```

'stocks are documents that companies have, if a person has stock it proves that a person owns a part of the company'.

1.4 The geometry of definitions

Computational linguistics increasingly relies on *word embeddings* which assign to each word in the lexicon a vector in n-dimensional Euclidean space $\mathbb{R}^n$, generally with $150 \leqslant$

$n \leqslant 800$ (typically, 300). These embeddings come in two main varieties: *static*, where the same vector $\mathbf{v}(w)$ is used for each occurrence of a string w, and *dynamic* (also called *context-sensitive*) where the output depends on the context x_y in which w appears in text. On the whole, dynamic embeddings such as BERT (Devlin et al., 2019) work much better, but here we will concentrate on the static case, with an important caveat: we permit *multi-sense* embeddings where a single string such as *free* may correspond to multiple vectors such as for 'gratis' and 'liber'. Our working hypothesis is that dynamic embeddings just select the appropriate sense based on the context.

Embeddings, both static and dynamic, are typically obtained from large text corpora (billions of words) by various training methods we shall return to in Chapter 8, though other sources (such as dictionaries or paraphrase databases) have also been used (Wieting et al., 2015; Ács, Nemeskey, and Recski, 2019). Most of the action in a word embedding takes place on the unit sphere: the length of the vector roughly corresponds to the log frequency of the word in the data (Arora et al., 2015), and similarity between two word vectors is measured by cosine distance. Words of a similar nature, e.g. first names *John, Peter,...* tend to be close to one another. Remarkably, analogies tend to translate to simple vector addition: $\mathbf{v}$(king) $-\mathbf{v}$(man) $+\mathbf{v}$(woman) $\approx$ $\mathbf{v}$(queen) (Mikolov, Yih, and Zweig, 2013), a matter we shall return to in 2.3.

For cleaner notation, we reverse the multi-sense embeddings and speak of vectors (in the unit ball) of $\mathbb{R}^n$ that can carry *labels* from a finitely generated set D^* and consider the one-to-many mapping $l : \mathbb{R}^n \to D^*$. We note that the degree of non-uniqueness (e.g. a vector getting labeled both *faucet* and *tap*) is much lower on the average than in the other direction, and we feel comfortable treating l, at least as a first approximation, as a function.

Definition 1. *A* voronoid $V = \langle \mathcal{P}, P \rangle$ *is a pairwise disjoint set of polytopes* $\mathcal{P} = \{Y_i\}$ *in* $\mathbb{R}^n$ *together with exactly one point* p_i *in the inside of each* Y_i.

In contrast to standard Voronoi diagrams, which are already in use psychological classification (see in particular Gärdenfors, 2000 3.9), here there is no requirement for the p_i to be at the center of the Y_i, and we don't require facets of the polytopes to lie equidistant from to labeled points. Further, there is no requirement for the union of the Y_i to cover the space almost everywhere, there can be entire regions missing (not containing a distinguished point as required by the definition). Given a label function l, if $p_i \in Y_i$ carries the label $w_i \in D^*$ we can say that the entire Y_i is labeled by w_i, written $l(Y_i) = w_i$.

Now we turn to learning. As in PAC learning (Valiant, 1984), we assume that each concept c corresponds to a probability distribution π_c over $\mathbb{R}^n$, and we assume that level sets for increasingly high probabilities bound the prototypical instance increasingly tightly, as happens with the Gaussians often used to model the π_c. An equally valid view is to consider the polytopes themselves as already defining a probability distribution, with sharp contours only if the softmax temperature is low.

It is often assumed in cognitive psychology that concepts such as *candle* are associated not just to other verbal descriptors (e.g. that it is roughly cylindrical, has a wick at

the axis, is made of wax, is used on festive occasions, etc.) but also to nonverbal ones, such as a picture of 'the candle' or even the characteristic smell of burning candles. In fact, image labeling algorithms such as YOLO9000 (Redmon et al., 2016) have considerable success in finding things in pictures and naming them, but generating prototypical images remains a research goal even for human faces, where the state of the art is most developed.

Definition 2. *A* linear voronoid *is a voronoid defined by hyperplanes h_j such that every facet of every polytope lies in one of these.*

By adding a hyperplane for each facet of every polytope, every voronoid can be made into a linear one, but our interest is with the sparse case, when many facets, not just those for adjacent polytopes, are on the same hyperplane. Thus we have two objectives: first, to enclose the bulk of each concept set c in some Y_i so that $\pi_c(Y_i)$ is sufficiently close to 1, and second, to reduce the cardinality of the hyperplane set. Each half-space is defined by a normal vector $\mathbf{f}$ and an offset (called the *bias*), and we call these *features* (rather than half-spaces) in keeping with standard terminology in machine learning.

Definition 3. *A vector* $\mathbf{v}$ satisfies *a feature* $\mathbf{f}$ *iff* $\langle \mathbf{v}, \mathbf{f} \rangle > b$

Since our central interest is with just one half-space to the exclusion of the other (see Chapter 4), we orient the normal vector so that a feature takes positive value in this affine half-space. Note that a normed vector has $n-1$ free parameters and the bias adds the nth, so feature vectors are not qualitatively different from word vectors. So that we don't have to move to a dual space we will also call the positive half-spaces features, and denote them by F_j.

Now we can restate our sparsity goal as finding features $F_1, \ldots, F_k$ so that all polytopes can be defined by the intersection of a few of these. We leave open the possibility $k > n$, i.e. that the system of features is *overcomplete*. As a practical matter, models with $n = 300$ work reasonably well, while we expect k to be in the 500–1200 range. What we are looking for is a finite system $\mathcal{F} = \{F_1, \ldots, F_k\}$ such that each of the Y_i is expressible as a sparse vector with nonzero (positive) elements only on a few (in practice, less than 10) coordinates.

Remarkably, these simple (and in case of Def. 3, completely standard) definitions are already sufficient for a rudimentary theory of communication. Assume two parties, a speaker and a hearer. They both have *mental spaces*, a place where they store not just words and other linguistic expressions, but also concepts, sensory memories, things that philosophers of language would generally treat as sortally different. The term is chosen to express our indebtedness to (Fauconnier, 1985; Talmy, 2000) and the entire loosely connected school of Cognitive Linguistics, but we don't use 'mental space' in exactly the same way as Fauconnier, especially as we are modeling it by ordinary n-dimensional Euclidean space.

Ideally, the speaker and the hearer share the same voronoids, and simple ideas or sensations can simply be communicated by uttering the label of the polytope where it

falls: I see a candle, and say *candle*. This is sufficient for the hearer to know which polytope was meant, and thereby gain some rough understanding of my mental activity. In reality, both speakers and hearers are aware that their mental spaces are not identical: my notion of a candle can differ from yours in ways that may be significant. But day-to-day communication is seldom hindered by this, by asking for a fork I'm unlikely to be handed a spoon. This is not because our $Y_{\texttt{fork}}$ polytopes have identical boundaries, but rather because the boundaries cover so much of the $\pi(\texttt{fork})$ probability mass that the symmetric difference between the polytopes of speaker and hearer is negligible.

The same logic extends to the vexing cases of hyperintensionals (Cresswell, 1975), phrases that describe contents that are not instantiated at all. I can speak of a pink elephant, and anybody who understands English understands what I mean with the same degree of (im)precision as they understand 'pink' and 'elephant'. Putting these two polytopes together just gives us their intersection, which works quite well even though in the real world this intersection happens to be empty. Note that the intersection can be empty even where there is no counterfactuality involved: a *former president* is by definition not a president, and at any rate it seems hard to maintain a subset of the space that contains former things. Since *former x* means 'was x, no longer x' i.e. a change of its x-ness, the point under discussion is one that has left Y_x.

In logical semantics it is a standard assumption that *extensions* of words, here modeled by polytope volumes, are changing with time. If I decide to paint a formerly black wall white, the meanings of *black* and *white* (standardly modeled by an indexed set of extensions e_λ, with the indexes running over the class of 'possible worlds' and called the *intension* of a word) remain constant, it is just their extensions that change with λ. We will assume a discrete time index t and require only three values 'before', 'now', and 'after'. We will discuss temporal semantics in greater detain in 3.2 – here we will simply assume three voronoids V_b, V_n, V_a and consider *former* an operator that effects a change from the identically labeled polytope, say Y for 'president' that somehow moves a point corresponding to the subject, say *Obama*, from the interior of Y in V_b to the exterior in V_n.

We have in both of these models a vector $\mathbf{p}$ corresponding to *president* and a vector $\mathbf{O}$ corresponding to *Obama*. The key insight is that not only do these vectors remain static, but the polytope Y that surrounds $\mathbf{p}$ also remains unchanged. What changes is the scalar product: in V_b we had $\langle \mathbf{O}, \mathbf{p} \rangle > b$ and in V_n we have $\langle \mathbf{O}, \mathbf{p} \rangle < b$. It is not that the threshold for presidency b has changed: what changed is the definition of the scalar product. We will assume the standard basis for V_n, but some B (before) basis in V_b, some A (after) basis in V_a, and use $\langle \mathbf{O}B, \mathbf{p}B \rangle > b$ conjoined with $\langle \mathbf{O}, \mathbf{p} \rangle < b$ to express the meaning of *former*. We return to scalar products in Chapter 2.3, but note in advance that we follow the literature in being a bit more loose in terminology than is common in mathematics: we will use *basis* also for generating systems that are not necessarily linearly independent, and *scalar product* also for bilinear forms that are not necessarily symmetrical.

The geometric model offers its own sortal types: vectors, half-spaces, polytopes, matrices, and so on. We will link these up to the lexical categories of `4lang` in 2.1, but to build intuition we list some of the key correspondences here. Proper names are points, a matter we will discuss in greater detail in Chapter 8. This doesn't mean that all points p in concept space receive a label l that is a proper name, but by and large, all things can be named (have a proper name assigned to them), not just people, pets, or boats. Adjectives are typically half-spaces, with gradient effects modeled by the bias term, whereas common nouns are often polytopes (finite intersections of half-spaces) or projections thereof. Verbs, including the copula, carry time information, and their description often involves not just V_n, but also V_a and/or V_b as well.

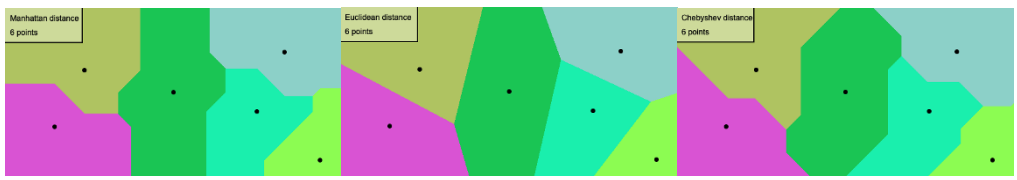


Fig. 1.1: Dependence of voronoids on metric chosen

Note that any set of vectors defines its own voronoid, but the boundaries of the cells depend on the metric chosen. This is illustrated in Fig. 1.1, which was generated using http://yunzhishi.github.io/voronoi.html. Since the probability mass is near the center, exact placement of the boundaries is of little interest.

We will use voronoids to represent the nominal aspects of *conceptual schemas*, compact configurations of knowledge pertinent to some domain. With the addition of verbal information (in particular, timing, see 3.2) these schemas become a linear algebraic version of Schankian scripts. As an example, consider the `exchange_` schema, roughly depicted in Fig. 1.2.

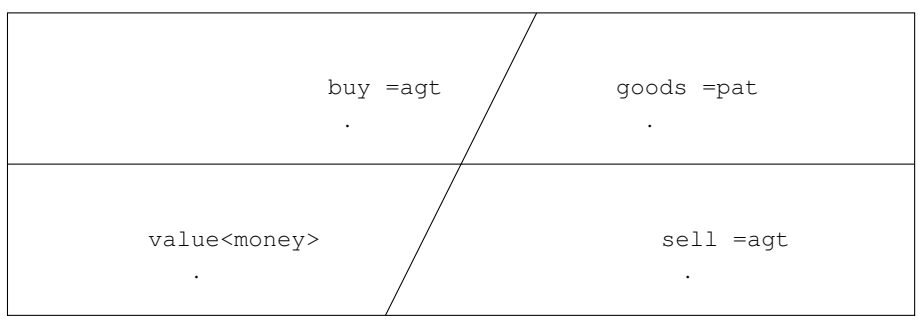


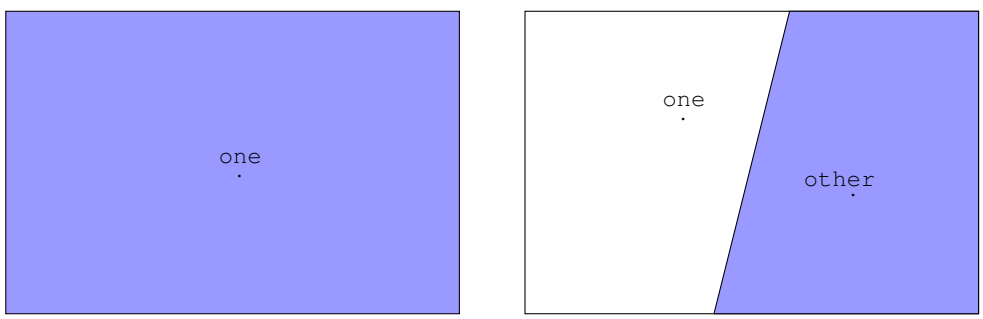
Fig. 1.2: `exchange_`

The words used are highly evocative: if we hear *sell* we automatically typecast the subject in the seller role, and the object in the 'goods' role. If we choose *buy*, the subject is

bound to the buyer role, and again the object of buying is treated as the goods. Before the exchange, the seller has the goods, and the buyer the money, afterwards the buyer has the object and the seller the money. This analysis (similar to the one proposed in Hovav and Levin, 2008) is easily implemented as hypergraph unification (see 1.5), and also in the vector calculus we are using, but we defer the details to 3.2, where we discuss handling the temporal aspects `before` and `after`.

While unification proceeding from the keywords *buy* or *sell* proceeds naturally, the word *goods*, rarely used outside the context of shipping/insurance contracts, is quite a bit less evocative in English, and is really used just for want of a better term. The same can be said for the word *money*, even though the association is strong, buying is what money is for, and selling is what earns money. (Also, in the full lexical entries for *buy/sell*, `money` is merely a default: clearly goods can exchanged for services and other things of value.) Typically, we invoke the schema from the perspective of the controlling participants, the potential agents, though alternatives like promoting the money to the agent role, *In this village, ten thousand will buy you a beautiful house*, are often feasible.

In Fig. 1.3 we depict the two simplest schemas. The left panel shows a voronoid with a single region labeled, for want of a better name, `one`. Since this encompasses everything, we could have called it `all` or `whole` just as well. The ambiguity between *one* and *all*, reminiscent of the first basic principle of Plotinus "the One" (or "the Good"), will not play the same generative role here as with Plotinus, and we will also refrain from entertaining analogies between the right panel and Gnostic thought.



(a) `one/all` (b) `other`

Fig. 1.3: `one` and `other`

The type difference between our first quantifier, `gen`, defined simply as a vector with the same value $1/d$ on each component, and `all`, is very clear. `gen` is simply a nominal, whereas `all` is a schema that requires implicit or explicit typecasting: as in *all books are for sale*, where *all* is already limited to *in this store* (Kornai, 2010b).

The same difficulty of naming certain regions of the voronoid, a problem we already encountered with `goods`, is manifest on the white side of the right panel of Fig. 1.3. The blue side directly defines `other`, but whatever is on the white side is typecast to

one. (Numerical *one* is defined in opposition to *more*, and has little to do with the one in either part of the figure.) To keep such technical names distinct from the vocabulary, we will suffix them with _. The names of schemas (voronoids defined by unordered sets of vectors) will be enclosed in the same curly braces { } as more ad hoc statements like {person has stock} in 1.3 above. The graph-theoretic view, where schemas are simply hypernodes, does not do full justice to schemas as information objects – we will discuss this problem in 1.6.

To summarize the key geometric ideas, most words (proper names in particular, but also common nouns, adjectives, and verbs) correspond to vectors or polytopes with distinguished vectors. We can also compute vectors for adpositions and other function words like *be* that we will shortly turn to, but we actually consider these to be matrices (relations between vectors). Consider *bark*. For us, this is a pair of vectors $bark_1$ 'cortex' and $bark_2$ 'latrat(us)' indistinguishable without context, be it morphological (*barked, barking* can only refer to dog bark) or larger (*birch bark* can only refer to tree bark). When we assign just one vector to this, this is just the log frequency weighted sum of the two vectors corresponding to the two senses, sitting in polytopes Y_1 and Y_2 that are not even adjacent in concept space.

In terms of the distinguished points it is easy to tell them apart: Y_2 falls in the sound subset since $bark_2$ is defined as sound[short, loud], <dog> make, whereas Y_1 is not a sound. The separating surface is not unique, $bark_1$ is some kind of covering that trees have, and as such, it is an *object* defined by the cluster of properties that physical objects have: thing, <has colour>, has shape, has weight, <has surface>, has position, <lack life>, and clearly dog barks have none of these, so any of these surfaces can be used to separate the two polytopes.

bark/2517

object

The one vector for *bark* that we obtain from running GloVe (Pennington, Socher, and Manning, 2014), word2vec (Mikolov et al., 2013), or any of the other algorithms must be related to the $bark_1$ and $bark_2$ vectors by addition, weighted by log frequency (Arora et al., 2015). How is this differentiated from cases like boat being defined as ship, small, open? In other words, how do we know that *bark* contains two vectors corresponding to two distinct senses, while *boat* contains only one, corresponding to a single unified sense? The answer is that this fact can't be read off of the vectors themselves, but can be read off the polytopes: in the *bark* case we have two, but in the *boat* case only one polytope. This is actually a key distinguisher between the more common variety of vector semantics that relies on word vectors directly and the variety that is presented here, since without polytopes the 'raw' vectors for homonymous and polysemous cases are indistinguishable.

A related question is how to distinguish the head from the subordinate elements in a definition: how would the definition ship, small, open differ from open, small, ship? Here we could rely on the fact that addition, after softmax, is not associative: $\sigma(\mathbf{a} + \sigma(\mathbf{b} + \mathbf{c})) \neq \sigma(\sigma(\mathbf{a} + \mathbf{b}) + \mathbf{c})$, in fact it is the term added last that would receive the greatest weight. More important is the observation that in this definition, open really means it lacks a deck (while an *open bottle* lacks a cork, and an *open*

letter lacks the privacy protection offered by an envelope) so we have 'open in the way ships can be open' so a more pedantic definition would be 'ship, open as ship' and of course 'small as ship' for a boat is quite large on a default (human) scale. This gives `ship` a weight of 3, `open` and `small` a weight of 1 each. After softmax ($\beta = 1$) this becomes (0.787,0.106,0.106).

London

In general, we will assume that the head carries larger weight than the modifiers. This is especially clear in definitions such as *London* as `city, @London`. People, not just readers of the Wikipedia article that `@London` points to, but all competent speakers of English, have a wealth of information about London. Much of this information (e.g. images of Tower Bridge, Beefeaters, Parliament, ...) is non-linguistic (not pertinent to grammar), and the projection on the subspace L is dominated by one component (one-hot) on `city`.

The key link type in the algebraic (hypergraph) description we now turn to is the type 0 (is, is_a) link, which simply corresponds to set-theoretic containment: if A, as subset of $\mathbb{R}^n$ is contained in B, we say that

$$l(A) \text{ is\_a } l(B) \tag{1.1}$$

In the algebraic representation lexemes, and larger sentence representations, are hypergraphs, hypergraph unification is a well-defined symbol-manipulation operation, and such symbol manipulation can be performed by neural nets (Smolensky, 1990). In 2.3 we will present a more direct, geometric description in terms of a simple eigenspace model, keeping in effect only the linear and the quadratic terms from the full generality of the tensor model. This will answer a whole set of vexing problems, such as defining the meaning of *be*, where even the magnificent LDOCE resorts to circularity, offering the following senses:

1. used with a present participle to form the tenses of verbs
2. used with past participles to form the passive
3. used in sentences about an imagined situation
4. used in sentences to introduce an aim when you are saying what must be done in order to achieve it
5. used instead of 'have' to form the tense of some verbs
6. used to say that someone or something is the same as the subject of the sentence
7. used to say where something or someone is
8. used to say when something happens
9. used to describe someone or something, or say what group or type they belong to
10. to behave in a particular way
11. used to say how old someone is
12. used to say who something belongs to
13. used to talk about the price of something
14. to be equal to a particular number or amount
15. to exist

We emphasize that we are not singling out LDOCE for unfair treatment here. The online Cambridge Dictionary has a very similar assortment of 'used to' definitions:

1. used to say something about a person, thing, or state, to show a permanent or temporary quality, state, job, etc. *He is rich. It's cold today. I'm Andy. That's all for now. What do you want to be (= what job do you want to do) when you grow up? These books are (= cost) $3 each. Being afraid of the dark, she always slept with the light on. Never having been sick himself, he wasn't a sympathetic listener. Be quiet! The problem is deciding what to do. The hardest part will be to find a replacement. The general feeling is that she should be asked to leave. It's not that I don't like her - it's just that we rarely agree on anything!*
2. used to show the position of a person or thing in space or time *The food was already on the table. Is anyone there? The meeting is now (= will happen) next Tuesday. There's a hair in my soup.*
3. used to show what something is made of *Is this plate pure gold? Don't be so cheeky! Our lawyers have advised that the costs could be enormous. You have to go to college for a lot of years if you want to be a doctor. Come along - we don't want to be late! Oranges, lemons, limes and grapefruit are types of citrus fruit.*
4. used to say that someone should or must do something *You're to sit in the corner and keep quiet. Their mother said they were not to (= not allowed to) play near the river. There's no money left - what are we to do?*
5. used to show that something will happen in the future *We are to (= we are going to) visit Australia in the spring. She was never to see (= she never saw) her brother again.*
6. used in conditional sentences to say what might happen *If I were to refuse they'd be very annoyed. (formal) Were I to refuse they'd be very annoyed.*
7. used to say what can happen *The exhibition of modern prints is currently to be seen at the City Gallery.*
8. to exist or live *(formal) Such terrible suffering should never be. (old use or literary) By the time the letter reached them their sister had ceased to be (= had died).*

More traditional dictionaries, such as *Webster's New World* (Guralnik, 1958), use even more vague terms in the definition, such as 'used to express futurity, possibility, obligation, intention, etc'; *The Concise Oxford* (McIntosh, 1951) has, distributed among several senses, 'exist, occur, live, remain, continue, occupy such a position, experience such a condition, have gone to such a place, busy oneself so, hold such a view, be bound for such a place, belong under such a description, coincide in identity with, amount to, cost, signify'. A more unified treatment seems warranted, and will in fact be provided in 2.3.

1.5 The algebra of definitions

The method of capturing meaning by definitions is at the heart of our undertaking: each definition (line in the dictionary) corresponds to an equation or inequality in the overall system that determines the meaning of each part. Of the three methods discussed here, compositional semantics has long been dominated by the formulaic approach. This approach would have to be coupled to a theory of *grounding* and a theory of *meaning postulates* to fulfill its promise (see S19:3.7-8 for details) and we will not spend any time trying to turn our algebraic formulas into formulas of logic.

The use of (hyper)graphs is an algebraic method on its own, one that can be matched to the compositional manner in which we build the formulas by means of parallel synchronous rewriting. When it comes to detaching meaning representations from linear ordering, graphs are particularly useful, but to take full advantage of them we will need a workable definition of a 'well-formed hypergraph'. To this end, let us first recapitulate the syntax of the definitions we surveyed 1.3 in context-free rules.

1. Definition → Definiendum Definiens (% Comment)
2. Definiendum → Atom
3. Definiens → MarkedClause (',' MarkedClause)*
4. Comment → (ArbitraryString)
5. MarkedClause →DefaultClause|PositionClause|ComplexClause|Clause
6. DefaultClause →'⟨'Clause'⟩'|λ
7. PositionClause → PositionMarker mark_ UnaryAtom
8. ComplexClause → {Definiens}
9. PositionMarker → '"'SuffixMarker|PrefixMarker|InfixMarker""
10. Atom → PlainAtom|NumberedAtom|ExternalAtom|PositionMarker
11. NumberedAtom → PlainAtom'/'Number
12. ExternalAtom → '@'WikipediaPointer
13. PlainAtom → UnaryAtom|BinaryAtom
14. UnaryAtom → `Asia|acid|...|yellow|young|=agt|=pat`
15. BinaryAtom → `at|between|cause_|er_|follow|for_|from|has|in|ins_|is_a|lack|mark_|on|part_of|under`
16. Clause → 0Clause|1Clause|2Clause|FullClause
17. 0Clause → Atom'['Definiens']'|Atom'('Definiens')'|Atom
18. 1Clause → BinaryAtom Clause
19. 2Clause → Clause BinaryAtom
20. FullClause → ComplexClause BinaryAtom ComplexClause

As usual in syntax definitions, | in a rule indicates choice and () indicates optionality. (This is the metalanguage: in the language itself we use angled brackets to denote optional parts of definitions, see Rule 6.) This way, 1. abbreviates two rules, one containing no comment and the other containing a Comment after the % sign, which can be expanded to an arbitrary string by Rule 4. Needless to say, comments are irrelevant for

the emerging representations, and in the system of parallel synchronized rewriting that we will turn to in 1.6 rules governing the comments will be discarded.

In regards to Rule 2, it should be noted that Atom is intended in the sense of 'dictionary entry' and may include expressions such as *I beg your pardon* which have a unitary meaning 'please repeat what you just said' quite disinct from their compositional sense. The intuition is the same as with lexemes (cf. S19:3.8,4.5) both in linguistics and lexicographic practice: different senses e.g. for *chrome*$_1$ 'hard and shiny metal' and *chrome*$_2$ 'eye-catching but ultimately useless ornamentation, especially for cars and software' often correspond to different words in another language (and when they systematically fail to, we begin to suspect that the purported senses are not distinct after all).

The right-hand side of a definition, the Definiens, is given as one or more marked clauses. The marking can be for defaults, marked by $\langle\rangle$, see Rule 6; for position (within word, or more rarely, among words), marked by doublequoted material, see Rule 7 and 2.2; for complexity, set-theoretical comprehension of several elements, marked by {}; or it may not be marked for any of these, yielding a clause. (As a practical matter, less than 20% of `4lang` defining clauses are marked.)

In a similar manner, we differentiate between ordinary (plain) atoms, and those that are numbered for disambiguation using Rule 11. NumberedAtoms are there simply to provide the same kind of sense disambiguation that lexicographers generally do by subscript numbering, except that we find it expedient to keep the index set 1–3,999 fixed, rather than restarting indexing for each (English) word. For example, we define *set/2746* somewhat similarly to mathematical sets as `group, has many(item), together, unit, item has common(characteristic)` but *set/2375* as `=agt cause_ {=pat at position[<stable>,<pro-per>]}`. `set/2746` `set/2375`

In a more hardcore system we could keep only the numbers: the words are there only to help with human readability. It is a historical accident that English uses the same syllable for both `2746` and `2375`, but from the Hungarian-Latin-Polish bindings it is evident that `kollekcio1 classis kolekcja` and `tesz pono kl1as1c1` are not the same thing. (In this particular case this would also follow from their lexical categories, see 2.1, but these are never used for disambiguation.) Generally we suppress the disambiguation indexes, but note that the ambiguity of English *set* cannot be expressed by making these optional: whenever there is more than one lexical entry with the same English printname, disambiguation numbers are obligatory (as the true heads of the NumberedAtom construction, they are the only obligatory part).

Another kind of specially marked atom is provided in Rule 12 by pointers to the encyclopedia. These are given in abbreviated style: for example the *Asia* in `@Asia` corresponds to `https://en.wikipedia.org/wiki/Asia`. Finally, the use of position markers, doublequoted strings with an explicit insertion locus marker __ that shows whether the definiendum is prefixed, suffixed, or infixed (Rule 9) is no more than a simple workaround to make sure semantics doesn't get entangled in all the technical issues of morphophonology (see 2.2 and 2.5 for further discussion).

Rule 3 is the one we started out with: a definition is the comma-separated conjunction of one or more (marked or unmarked) Clauses. Only a quarter of the definitions have four or more conjunct clauses, and over a quarter have only one, the average number of clauses is 2.68. To understand the internal structure of a clause, we need to look more closely at the alternatives in Rule 16. 0Clauses are elementary predicates, and ComplexClauses can be pretty much anything a definiens can be. The 1Clause and 2Clause constructs serve to help make the syntax human-readable, at least for those humans who are comfortable with SVO word order. Take something like

```
blood ve1r sanguis krew 2599 N
      liquid, in body, red
```

The first clause simply says that blood is a `liquid`, and the third one says it is (or is_a, `4lang` makes no distinction) `red`. In the middle we find a 1Clause (subject clause, Rule 18) that puts `blood` to the left, and `body` to the right of a relational predicate `in`, guaranteeing that `blood in body` is part of the definition of `blood`, without making it appear in the definiens. 2Clauses (object clauses, Rule 19) behave similarly:

```
mud sa1r lutum bl1oto 2056 N
    substance, wet, earth, soft, sticky, water in
```

abbreviating `water in mud` which makes clear that it is mud that contains water, not the other way around. Relational elements are discussed further in 2.3, but Rule 15 makes clear that they come from a small, closed list containing only 16 elements. Similarly, unary atoms come from the closed list given in the Appendix, which represents considerable reduction compared to the list of 1,200 elements in S19:4.8.

The method of having implicit elements in a rule harkens back to the Pāṇinian device of *anuvṛtti* (see Kornai, 2007 7.3.1 for a brief description, and Joshi and Bhate, 1984 for a full treatment). For Pāṇini the goal of anuvṛtti is to enhance brevity in order to lessen the effort to memorize (improve human recallability), while here the shortening of definitions enhances human readability. For simplicity, Release V2 of `4lang` provides both a more machine-readable expanded version, and a more human-readable compacted one, with software to create each from the other, see 9.5.

Definition 4 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 att:$E \rightarrow V^*$ that assigns a sequence of pairwise distinct attachment nodes att(e) to each $e \in E$ and a mapping lab:$E \rightarrow \Sigma$ that labels each hyperedge. The size of the sequence att(e) is called the *type* or *arity* of the label lab(e). As Eilenberg machines (S19:Def.4.4) 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.

While the definition of hypergraphs stated above is reasonably standard, and it enables hooking up our machinery with that of s-graph grammars (Courcelle and Engelfriet, 2012; Koller, 2015) by means of synchronized string and hypergraph rewriting in 1.6, in `4lang` we concentrate on a simpler class of (hyper)graphs we will call *hypernode graphs* or *RDF graphs* or just `4lang` graphs.

Definition 5 A `4lang` graph or *hypernode graph* contains only ordinary directed edges (arrows) between a starting and an endpoint, these can be labeled 0,1, or 2, no other edge labels (colors) are countenanced. The hypernodes are ordered triples `(x,y,z)` where `x` or `z` may remain empty, As in the Resource Description Framework, members of the triple are called the 'subject', 'predicate', and 'object' of the triple. Subjects and objects (but not predicates) can themselves be `4lang` graphs.

This definition is again supported by a series of syntactic conventions to support human readability. Edge type 0 is used both for attribution *John is brave* and for IS_A indiscriminately. In larger graphs, we will write dashed arrows, $- \rightarrow$ instead of $\overset{0}{\rightarrow}$. Edge type 1 has the type number suppressed, we write $\rightarrow$ rather than $\overset{1}{\rightarrow}$. Finally edge type 2 will be depicted by a dotted arrow $\cdots >$ rather than $\overset{2}{\rightarrow}$.

In triple notation, $x \leftarrow y$ can be written as `[x,y,]`, and $y \cdots > z$ can be written as `[,y,z]`. A full triple `[x,y,z]` could be depicted as $x \leftarrow y \cdots > z$. For ease of presentation, we introduce a special symbol @ (not to be confused with the external pointer delimiter of Rule 12 above) that will be placed in the middle of edges that should *in their entirety* be the terminal point of some other edge. Consider the sentence *video patrem venire* traditionally analyzed in Latin grammar with an infinitival object, meaning that the object of seeing is neither the father, nor his coming, but rather the entire 'coming of father'. An English translation could be *I see father's coming* or even *I see father coming*.

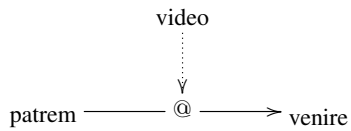


Fig. 1.4: *Video patrem venire*

In Chapter 5 this kind of graph structure will be further enriched by mappings from graph (hyper)nodes and (hyper)edges to small discrete partially or fully ordered sets.

Definition 6 A *valuation* is a partial mapping from some elements (both nodes and edges) of a hypergraph to a finite poset.

We will see in Chapter 2 in far greater detail how morphology and syntax are handled by the same mechanism, and here we omit the details of how syntax and morphological analysis of Latin sentences ordinarily proceeds hand in hand (S19:5.3).

Semantically, we have two units `father`, and `come`, the former being the subject of the latter. This is expressed in `4lang` syntax by `father[come]` or `come(father)`, keeping alive both function-argument alternatives explored in early Montague Grammar. Since this entire clause is the object of seeing, the whole sentence can be written

as `(1stsg) see father[come]`. We parenthesized the 1st singular pronoun, not overt in the Latin original, but inferable for the conjugated form of the verb, in anticipation of a fuller discussion of pronouns in 3.3. In RDF-style triple notation we have `[I,see,[father,come,]]`.

In terms of hypergraphs, we can consider `father` a single (atomic) vertex, but in light of the `4lang` definition

```
father apa pater ojciec 173 N
       parent, male
```

we are equally free to consider it a small hyperedge containing two vertices `parent` and `male`. We do not fully explore the hypergraph connection here (see S19:4.1, Nemeskey et al., 2013, Ács and Recski, 2018, and 7.4 for further discussion) but we note that our concept of "doing grammar by spreading activation" is almost identical to that of Jackendoff and Audring, 2020 7.2.3. This is not at all surprising, as they both go back to the same ideas (Quillian, 1969; Collins and Loftus, 1975), but it is worth emphasizing that this view comes hand in hand with obliterating the usual distinction between rules and representations. In effect, all the work is done by the representations and there are only a few generic rules that apply to all representations, primitive and derived, intermediary or final, the same way. This uniformity, characteristic of early combinatorial system like the untyped lambda calculus (Church, 1936) and categorial grammar (Ajdukiewicz, 1935) is maintained in all implementations of `4lang`, be they by Eilenberg machines (which directly formalize spreading activation), by (hyper)graph kernel methods (Ghosh et al., 2018), or by direct linear algebraic manipulation.

1.6 Parallel description

So far, we have three main approaches to endowing natural language expressions with semantics: the formulaic, the geometric, and the algebraic approaches discussed in 1.3, 1.4, and 1.5 respectively. All three have a long tradition going back to the 1960s, with many current variants. No doubt other approaches, such as the (now deprecated) automata-theoretic work, are feasible. The view we take here is that all these approaches are algebras of their own, and as such they can be connected by a parallel hyperedge rewriting system with as many branches as there are contenders for the notion 'semantic representation'. For example, the Abstract Meaning Representation (AMR) theory Banarescu et al. (2013) could be added as another branch, and for those content with the rough semantics encoded in explicit marking of head-dependent relations, Universal Dependencies could be added as yet another branch. In fact, some of the applied work discussed in 9.1 already transduces UD to `4lang` .

The idea of *syntax-directed translation*, going back to Aho and Ullman, 1971, is standard both in compiler design and in semantics, where it is considered to implement the Fregean principle of compositionality (see S19:1.1) by two systems operating in parallel: a syntax that, proceeding from the bottom (leaf) nodes gradually collects these together, and a semantics that computes at each step a formula based on the formulas associated

to the leaves and associates it to the parent node, using only *synthesized* attributes in the sense of Knuth, 1968. The basic idea has been fruitfully generalized for more powerful rewriting methods (Rambow and Satta, 1994; Shieber, 2004), and here we suggest, with implementation planned for Release V3, a hyperedge replacement framework (see Drewes, Kreowski, and Habel (1997) for a detailed overview) for two reasons: first, because it offers great clarity in regards to separating the metalanguage from the language, the tools from the objects themselves, and second, because it has an efficient implementation, the Algebraic Language Toolkit (Alto).

Alto (Gontrum et al., 2017) is an open-source parser which implements a variety of algebras for use with Interpreted Regular Tree Grammars (Koller and Kuhlmann, 2011; Koller, 2015) to simultaneously encode transformations between strings, trees, and `4lang` graphs. Alto has been used for semantic parsing both in Groschwitz, Koller, and Teichmann, 2015 and in the applied work we will discuss in Chapter 9, but a full Alto implementation of `4lang` is still in the planning stage. While this is hard to guarantee in advance, early experience suggests Alto will work well as the computational substratum, the kind of abstract machine the calculus is implemented on. S19:Def 5.8 used Eilenberg machines for implementing spreading activation, an approach we still consider viable for theoretical clarity, but one that has not gained traction beyond a small group of devotees. As Maler and Pnueli, 1994 already warned

> Another sociological problem associated with Eilenberg's construction is the elegant, concise, and motivationless algebraic style in which it is written, which makes it virtually inaccessible to many contemporary theoretical computer scientists.

This time we go with the flow, and take to heart William Stein's maxim: *Mathematics is the art of reducing any problem to linear algebra*. But much of the linear algebraic development has to wait until Chapter 6 and beyond, and in the meantime we assume a different, still algebraic but perhaps better motivated, system built on the hypernode graphs of Definition 5. To prevent any confusion, we emphasize that the machinery we propose, *hyperedge* replacement, uses a metalanguage that relies on a different notion of hypergraphs (Definition 4) than the object language. That the metalanguage is not the same as the object language should come as no surprise to students of logic or computer science: a well known example is regular expressions which describe finite state object languages but use a context-free metalanguage.

One particular semantic representation that we shall pay attention to is the *translational* approach whereby the semantics of one natural language is explicated in terms of another natural language. For this to work, we need to consider each natural language a kind of string algebra, operating on semantic atoms, morphemes. For the sake of simplicity, we will consider only one string operation, concatenation, even though more complex nonconcatenative operations are present in many languages. To the extent syntactic structure explicates semantic relations (e.g. the head-dependent relation that

plays a central role in dependency grammar), we may even decorate the nodes with the appropriate graph structure links (see Chapter 9).

The atomic components of all algebras are the morphemes and words (including multi-word expressions that contain orthographic word boundaries (whitespaces). These are conceptualized as small, and individually rather limited nodes loosely connected by an is_a network. This network is a DAG but not necessarily a tree: undirected cycles are common, as in the classic Nixon diamond (Reiter and Criscuolo, 1983). Edges of this network are labeled 0. There are two other networks, with edges labeled 1 and 2. In these, no undirected or directed cycles have been found, but confluences (directed edges originating in different nodes but terminating in the same node) are not rare. Rough translational equivalents are provided across the 4 languages of `4lang` and in principle pivot-based translation across these using the synchronous rewrite mechanism is possible.

This is not to say that the elementary components (nodes) are devoid of non-linguistic content: they may contain pointers pointing to all kinds of encyclopedic (verbal) knowledge as well as non-verbal memory: sounds, images, smell. Further, activation of such may bring activation of the nodes, so these pointers (associative links) are often bidirectional, or better yet, directionless. The entire set of nodes is viewed as adiabatically changing: new nodes are added as the individual, whose linguistic capabilities are being modeled, is acquiring new words/morphemes.

In addition to these static node-like structures, we permit the building of more dynamic structures, hypernodes, by a process of *grouping*. In the simplest case, this is just coordinating a few elementary nodes: instead of *Tom, Dick, and Harry* we can refer to the collective entity they form as *the boys*. Typically, hypernodes are nonce elements: *boys* may very well refer to other groups, say *Bill and Dave*, depending on context. Such temporary configurations, best thought of as the meanings of constituents, are denoted in the syntax by curly brackets. On rare but important occasions we will also encounter strongly lexicalized groupings we call *schemas*. For example, we will distin-

`place` guish `place`, defined as `point, gen at` from `{place}`, a complex schema we will discuss in great detail in 3.1.

One conceptual difficulty we already touched upon in 1.5 is that nodes and hy-

`fight` peredges are not that different. In fact, when we define *fight* as `person want {harm at other(person)}, ins_ weapon` this means that we can at any time replace the node `fight` by the hypernode `{person want {harm at other (person), ins_ weapon}` *salva veritate*. This kind of substitution plays a major role in the low-level deduction process that takes place synchronous with text comprehension: when we hear *John fought the coyote with his bare hands* we automatically put *bare hands* in the ins_ slot and typecast it as a weapon.

Complex deduction like this will have to be built from more elementary operations. The nodes (in what follows, we well refer to hypernodes also as nodes, unless there is a specific reason to distinguish the two) are capable of (i) activating themselves and adjacent edges to various degrees; (ii) copying themselves (triggered by the keyword

other); (iii) unifying subnodes. This unification, which is automatic for nodes named identically (or for the element gen, which is capable of unification with anything), is not to be confused with coercion (see 3.3), though the effects are somewhat similar.

1. Definition → Definiendum Definiens (% Comment)

Unlike in generative theories of the lexicon (Pustejovsky, 1995), where the process of enumerating senses is assumed to start from some start symbol S, we see our system of definitions as a network (hypergraph). This is a large structure with tens, if not hundreds of thousands of hypernodes characterizing the lexical component of adult linguistic competence, and there is no starting point as such. Even developmentally, the first words learned will often correspond to rather complex sensory units (*mama* is a great deal more complex than *light*) as long as they are motivationally salient. As diary studies of early vocabularies clearly demonstrate, new words are often completely unattached to the existing inventory: before a child learns *peepee* or *doodoo* (apparently equally applicable for toilets, people on the toilet, or hearing the toilet flush) there is not one word related to excretion that could be used to describe the meaning (Rescorla, 1980).

For us, this rule plays a key role in *expansion*, the operation whereby we substitute the definiendum by the definiens. We emphasize that this is not a generative operation, but a deductive one that replaces one hypergraph, in which the definiendum appears as a node, by another one, where this node is replaced by the entire definiens, typically resulting in a more complex hypergraph. For example, in *John appears drunk* we may replace *appear* by its definition gen think {=agt is_a =pat} to obtain gen think John is_a drunk. As we shall see in Chapter 9, expansion, now implemented using the GraphMatcher class of the NetworkX library, plays a key role in analyzing lexical entailment (Kovács et al., 2022a). We return to this operation, our model of *spreading activation*, in 7.4.

appear

In terms of vector representations, substitution doesn't change the actual system of vector space objects described, but may bring to light a view of these objects from another basis. Consider for example *crime*, defined as action, illegal and trace *illegal* through the system by expanding it as bad for_ law to obtain action, bad for_ law. By tracing further *bad* as cause_ hurt we end up with an even more compact definition of crime: action, hurt law – this has the advantage that we don't have to get sidetracked with the issues of experiencer subjects (see 2.4) that the use of for_ would bring in tow. At the same time, by highlighting the fact that crimes are actions, this definition makes evident that crime has a temporal dimension (and an agent, given that *action* is defined as person do). A noun like *tree* which is defined by plant, has material[wood], has trunk/2759, has many(branch) will have neither of these implications.

crime

illegal

bad

action

tree

2. Definiendum → Atom

Definienda are always numbered atoms. (The numbering is generally omitted for ease of presentation.) Semicompositional definienda, where a great deal (but not all) of the

meaning can be inferred from the parts will have to be adjoined as atoms. We discuss the key technique, *subdirect decomposition*, in 2.2, but offer a simple, and from the lexicographic standpoint easy to defend, example here.

Consider *preferred stock* 'stock that entitles the holder to a fixed dividend, whose payment takes priority over that of ordinary share dividends' (Oxford) 'has a higher claim on assets and earnings than common stock has' (Investopedia). The defintion `stock, preferred` captures most of the meaning, both that preferred stock is a kind of stock,
`preferred` and that it is in some sense *preferred*, a notion defined in `4lang` as `{gen like/3382 =pat} er_ {gen like/3382 other}, =agt choose =pat`. However, this does not say under what circumstances will this preference be manifest. Clearly it not the preference of the buyer that is relevant here, for if it were, nobody would ever buy common stock. The technical definition makes clear that it is for dividends, and in case of the division of assets, that preferred stock has an advantage, and this fact is external to (cannot be inferred from) the meaning of *prefer, preferred,* or *preference*.

Semicompositional expressions are spread over a continuum with fully compositional expressions at one end, and entirely non-compositional ones at the other. For a multi-word example consider *go Dutch* 'split the bill after a meal' and for a single word consider *went* which will mean, under any analysis, the past thense of *go, *go-ed*. If we assign meaning representation f to expression F and g to G, no case where the meaning of FG involves some extra element h beyond f and g can be considered fully compositional. A great deal depends on the lexicographic purpose: the same FG will be considered compositional if for some reason we consider the h element negligible, and non-compositional if we must make substantive use of it. For example the difference between *hold* and *give* is generally quite clear, yet in the expressions *hold/give a lecture* they are fully interchangeable, acting as *light verbs* (Jespersen, 1965) that contribute little beyond adding a verbal aspect to *lecture* which, in isolation, is ambiguous between noun and verb.

3. Definiens → MarkedClause (',' MarkedClause)*

In terms of graphs, each of the defining clauses are linked to the definiendum by type 0 links. In terms of the vectorial representation, the polytopes corresponding to the clauses are intersected. Noncompositionality arises precisely in those cases where the intersection of the clause polytopes is a superset of the definiendum polytope.

4. Comment → (ArbitraryString)

Comments are restricted to a separate column of the file. Since the comments themselves only benefit the human reader of the file, the rule is a no-op as far as its effect on meaning is concerned. Most of the comments list potentially interesting cross-linguistic tidbits, e.g. that the hand of an English person has four fingers and a thumb, while the hand of a Hungarian has five fingers, as the thumb is called *nagyujj* 'big finger'. Phenomena such as this are common (indeed, typical) and they served as motivating examples for taking the abstract, algebraic view.

5. MarkedClause →DefaultClause|PositionClause|ComplexClause|Clause

Unless overriden, default clauses are carried (credulous inference). This hides a great deal of complexity, both in terms of the deontical status of default existents (see 6.2) and the default logic overall (see 6.4). When the default fails, we use rewrite rule 6.

Position clauses, just as `mark_`, are language-specific. They are used in a rudimentary fashion throughout the book, mostly to indicate whether a form is free-standing or affixal, and if an affix, is it a prefix or a suffix, and sometimes to describe slightly more complex situations (infixes, circonfixes, tripartite constructions like `er_`). Of necessity, we abstract away from a great deal of micro-syntax, since most of the 'tinkering' is both highly syntactic and highly language-specific, while our focus is with the semantic and the universal.

Complex clauses are typically used in subordinate position. As an example, take *attract* `=agt cause_ {=pat want {=pat near =agt}}`. What is being caused is itself a complex state of affairs, the patient wanting something, and that something again is a complex state, the patient being near the agent. `attract`

Rule 5 groups all these together with simple clauses, but this is only for the convenience of the formula parser. There are no deep similarities between default clauses and complex clauses, but one is surrounded by $\langle\rangle$ and the other by $\{\}$ so the notation brings them close.

6. DefaultClause →'⟨'Clause'⟩'|λ

In expansion, the second alternative means we do override i.e. we omit the default for some reason. Consider *sugar* defined as `material, sweet, <white>, in food, in drink`. We still have to deal with *brown sugar* and not get entangled in some sophistry about how brown is really a kind of white, or how brown sugar is both brown *and* white, etc., see 6.4. `sugar`

7. PositionClause → PositionMarker mark_ UnaryAtom

`mark_`, as opposed to the non-technical *mark* `sign, visible`, is a semi-technical term, the closest we will get to the Sausserean sign: its agent is a sign, its patient is a meaning, and it itself means 'represent': *mark_* `=agt[sign], =pat[meaning], represent`. A typical example would be in the last clause defining the English word *buy* we discussed in 1.4: `=agt receive =pat, =agt pay seller, "from _" mark_ seller`. Whatever follows the string "from" is the seller in English – in Hungarian it would be whatever precedes the ablative case marker. `mark` `mark_` `buy`

8. ComplexClause → {Definiens}

The key distinction between simplex and complex clauses is that the former appear in intersective situations, while the latter are unions, both in graphs and in vectors. Consider

defend `=agt cause_ {=pat[safe]}`. The agent doesn't cause the patient, or the safety, what the agent causes is the safety of the patient, a complex situation with two components. In our example of *attract* above, what the agent causes is also a complex situation, one that has another complex situation as one of its components. `defend`

Here it is perhaps worth emphasizing that there cannot be two agents, or two patients, or indeed, two of anything, unless this is signalled by the `other` keyword. Unification is an automatic low-level process that we have not incorporated in these rewrite rules in order to keep them simple, but are used in the IRTG/Alto system under development.

9. PositionMarker → '"'SuffixMarker|PrefixMarker|InfixMarker'"'

Position clauses are language-dependent, and `4lang` only gives them for English. They are primarily used in morphology, where the underscore _ is written together with the stem, and in the rare cases where English uses positional marking (e.g. subjects in preverbal, objects in postverbal position) they are separated by whitespace. The reader should not take this simple notation as some profound statement about proto-syntax – position markers appear in less than 5% of the dictionary, and English syntax offers many constructions that are inconvenient to describe by this mechanism (see 2.1 on the autonomy of syntax).

The system gives a good indication of what is what, e.g. that in *buy* `"from _" mark_ seller`, but without more developed morphophonological machinery this is generally insufficient to drive a parser. This is because the quoted strings rarely stay invariant: there can be all kinds of changes both to the stem and to the affix (e.g. in the Hungarian ablative, *-tOl* the choice of realizing *O* as *ó* or *ő* depends on the vowel harmonic properties of the stem), linking vowels or consonants may appear, material may get truncated, there are suppletive forms, etc etc.

10. Atom → PlainAtom|NumberedAtom|ExternalAtom|PositionMarker

Atoms, just as clauses, are grouped here together for ease of parsing. Loosely speaking, an Atom is a minimal entry in `4lang` – a PlainAtom is just a word or morpheme, signifying a unique concept. Non- and semi-compositional entries get their own atoms (see discussion of Rule 2 above). We emphasize that the presence of compositionally non-derivabable meaning is insufficient for us to declare the entry non-compositional, for example, the Battle of Jena is just that, a battle that took place at Jena. We may very well be aware that Clausewitz was captured by the French in this battle, but such knowledge belongs in the encyclopedia, not the lexicon. Such knowledge is *inessential* for understanding what this battle was, even a graduate student of history can get an A on an exam or paper that doesn't mention this fact. This is in sharp contrast to the case of preferred stock: not knowing how it is preferred amounts to not understanding the MWE.

11. NumberedAtom → PlainAtom'/'Number

The numbering of the Atoms, effected by a slash followed by a serial number below 4,000, is just the standard disambiguation device to get around homonymy. A more human-friendly dictionary would use subscripts for different word senses. At the core level we are most interested in (Kornai, 2021) the numbering carries very little load: over 95% of the English headwords has only one sense in `4lang`. An interesting counterexample would be `place/1026` 'locus' versus `place/2326` 'spatium', see 3.1 for discussion.

12. ExternalAtom → '@'WikipediaPointer

ExternalAtoms are pointers to Wikipedia. They refer to concepts about which a great deal is known, such as the Battle of Jena, where this knowledge is properly considered a part of history, or Tulip, where the knowledge is really part of biology. As we discussed in 1.2, linguistic semantics is a weak theory that cannot serve as the foundation for all this kind of knowledge amassed by the sciences over the centuries.

13. PlainAtom → UnaryAtom|BinaryAtom

Almost all our atoms are unary. Binary atoms are a small, closed subset (see Rules 14-15), and we do not permit atoms of higher arity (Kornai, 2012).

14. UnaryAtom → `Asia|acid|...|yellow|young|=agt|=pat`

There can be millions of unary atoms such as pointers to the encyclopedia (see Chapter 8). `4lang` concentrates on the defining set, where we already know that less than a thousand items are sufficient. However, these are not defined uniquely. In linear algebraic terms, it is just the dimension of the basis that is given, the basis vectors can be chosen in many ways. A handful of elements like `=agt, =pat, wh, ...` are reasonable candidates from a universal standpoint, but many others, including natural kinds, are not. In (Kornai, 2010a) we wrote

> 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 Indoeuropean language, but for our purposes neither of these terms can be truly treated as primitive.

More important than the actual selection of defining words is the method we employ in proving that the set so selected is actually capable of defining everything else. Once this is demonstrated, the issue of *which* elements are chosen is seen to be equivalent to deciding which equations to simplify by substituting the definiens for the definiendum.

How do we define words in general? Our method is akin to the use of multi-stage rockets in lifting a payload. In Stage 1, we simply look up the word in the dictionary, typically LDOCE. For example, at *intrude* we find 'interrupt someone or become involved in their private affairs in an annoying and unwanted way'. In Stage 2, those familiar with the system will translate this to `=agt cause_[pause in =pat], after(=agt part_of =pat), =agt cause_ [=pat[angry]]` manually. In the implementation we use the Stanza NLP package[1] to create a UD parse of the definition, and the `dict_to_4lang` system (Recski, 2018) to transform this to `4lang` syntax.

One can be far more faithful to the original definition than we were here: clearly *annoying/unwanted* is not exactly the same as *make angry*. If this is significant for some purpose, we may trace the LDOCE definition of *annoy* 'make someone feel slightly angry and unhappy'; that of *slightly* to 'a little'; and adjust the last clause of the above definition to `=agt cause_[=pat[angry[little]]]`. The claim here is that there is no shade of meaning that is inexpressible by these methods, not that the automatic system can already create perfectly faithful definitions for each and every word in each and every context for each and every langauge. As is typical in NLP, the automated systems are somewhat inferior to the best human-achievable performance. We return to the matter of contextual disambiguation, whether to choose `fall/2694` 'cado' or `fall/1883` 'autumnus' in 6.4.

For other languages, we need to begin (Stage 0) with a bilingual dictionary translating the word into English, and proceed from there. Let us consider a word that is often claimed to have no English equaivalent, *schadenfreude* 'pleasure derived by someone from another person's misfortune' (Oxford). In Stage 1, we consult LDOCE to find that *pleasure* can be replaced by *joy*. This is not to say that these two words are perfect synonyms, but whatever shades of meaning distinguish the two appear irrelevant in the definition of schadenfreude. In Stage 2, we can go even further, and replace *joy* by
`joy` its `4lang` definition `sensation, good` to obtain 'good sensation caused by other person's harm' which becomes in the formal language of definitions `sensation, good, {other(person) has harm} cause_`. In this step we switched from *misfortune* to *harm* manually, because the former specifically implies bad luck (and thereby absolves the experiencer of responsibility) while the latter stands neutral on whether the person is the cause of their own bad situation or not. Since schadenfreude is appropriate for both cases, we need to revise the Oxford definition a bit.

This last step of emending a definition may look at first blush as something beyond the powers of any automated dictionary builder algorithm. But keep in mind that we already have several systems that assign vectors to words purely on the basis of corpora, and we may resort to these in refining any definition. Even more important, the addition of a new definition will bring in one more unknown, the definiendum, and one more equation, the definition itself. Therefore, if the original system was solvable, the new one will also be solvable.

[1] `https://stanfordnlp.github.io/stanza`

15. BinaryAtom → `at|between|cause_|er_|follow|for_|from|has|in|ins_|is_a|lack|mark_|on|part_of|under`

Unlike unaries, which come from a large open list, binaries are restricted to a small closed set. We represent unaries by vectors, as standard, or by polytopes surronding these, a slight extension of the standard. For binaries we use matrices, which are much more expensive, n^2 parameters for n-dimensional vectors. By far the largest group are spatial (or, in the sense of Anderson (2006), 'local') cases and adpositions which we will discuss in 3.1. These are the prototypical ones, and we will see how temporal, and even more abstract cases such as the instrumental, can be brought under the same formal umbrella (see 6.2 for instruments, and 2.4 for causation).

16. Clause → 0Clause|1Clause|2Clause|FullClause

For ease of parsing we group together a variety of Clauses subject to different expansion in an anuvṛtti-like process, as explained below.

17. 0Clause → Atom'['Definiens']'|Atom'('Definiens')'|Atom

0Clauses are defining clauses linked to the definiendum by a 0 link. A typical example would be *below* defined as `under`, or *fast* defined as `quick` – these are to be understood as 'below is a (kind of) under' or 'fast is a (kind of) quick'. When there are several defining 0Clauses, as is typical, the definiens is in 0 'is/is_a' relation to each of them: *dot* `mark, small, round` means 'a dot is a mark, a dot is small, a dot is round'. The square brackets are also abbreviating is/is_a in `A[B]` constructions, as in *energy* `work[physical]` which means 'energy is work (that) is physical' or, for even better conformity with English syntax, 'energy is physical work'. (We exhort the reader not to get bogged down in high school physics where energy is *capacity* for work. Our definitions, intended to capture a naive world-view, will rarely stand up to scrutiny from the contemporary scientific standpoint.)

`below`
`fast`
`dot`
`energy`

Constructions involving parentheses, `B(A)` are strictly equivalent to `A[B]` and are used only when this order sounds more natural. Example: *powder* `substance, more(particle)`. There is nothing in the system of definitions that strictly requires this: we are catering to English syntax where adjectives are preceeding the noun but can be reversed as in *blue box, the box is blue* but numerals and similar quantifiers don't really tolerate the same reversal *four legs, ??the legs are four*.

`powder`

18. 1Clause → BinaryAtom Clause

1Clauses are used whenever the definiendum should occupy the subject (1) slot in the definiens. Example: *bee* `insect, has wing, sting, make honey`. The implicit 0Clause links are bee is_a insect, bee is_a sting (yes, and dog is_a bark, a design

`bee`

decision that makes a great deal of sense within the larger system of unaries and binaries we will discuss in 2.1) but we will not say that a bee is_a 'has wing' or a 'make honey'. Rather, `has` is a BinaryAtom, and `make` is a non-atomic binary (an obligatory transitive as its definition contains an `=agt` and a `=pat`). When a clause begins with a binary, we automatically put the definiendum in its subject slot 'bee has wing', 'bee make honey'.

19. 2Clause → Clause BinaryAtom

2Clauses are similar to 1Clauses, except the definiendum fills the object slot of the defin-
food ing clause. Example: *food* `substance, gen eat` 'food is what people eat' (see 4.5 for the treatment of the generic quantifier `gen`). In parsing, each clause needs to be inspected whether it has a binary, and if so, whether the binary has both valences filled
make in, as in *make* `=agt cause_ {=pat[exist]}`. If the pre-binary position is empty, we are dealing with a 1Clause, if the post-binary position is empty, we are dealing with a 2Clause. When both positions are empty, the definiendum and the definiens rely on the
notice same agent and patient, as in *notice* 'animadverto' `know, see`.

20. FullClause → ComplexClause BinaryAtom ComplexClause

polish Finally, FullClauses have both the subject and the object slots filled. Example: *polish* `=agt cause_ surface[smooth, shine], =pat has surface`. 'agt polishing pat means that agt is causing the surface of pat to be smooth and to shine'.

Classroom experience shows that the system is learned relatively easily, with students providing remarkably similar, often identical, definitions after a few weeks. The exception is students of linguistics and philosophy, who really need to unlearn a lot, as they are professionally trained to have a fine ear for minute distinctions. The marriage of lexicography and encyclopedia-writing is never happy. Consider the definition of *potash* as given in Webster's 3rd:

> 1a: potassium carbonate, esp. that obtained in colored impure form by leaching wood ashes, evaporating the lye usu. in an iron pot, and calcinating the residue – compare pearl ash. b: potassium hydroxide. 2a : potassium oxide K_2O in combined form as determined by analysis (as of fertilizers) ⟨ soluble ∼ ⟩ b: potassium – not used systematically ⟨ ∼ salts ⟩ ⟨ sulfate of ∼ ⟩ 3: any of several potassium salts (as potassium chloride or potassium sulfate) often occurring naturally and used esp. in agriculture and industry ⟨ ∼ deposits ⟩ ⟨ ∼ fertilizers ⟩

What are we to make of this? The COBUILD project (Moon, 1987) and the resulting Collins-COBUILD dictionary, attempted to clarify matters by distinguishing three different senses:

1. another name for {potassium carbonate}, esp. the form obtained by leaching wood ash

2. another name for {potassium hydroxide}
3. potassium chemically combined in certain compounds

But is it now carbonate or hydroxide? Or, perhaps, both could be subsumed under 'certain compounds'? LDOCE (Procter, 1978) avoids chemistry altogether:

> any of various salts of potassium, used esp. in farming to feed the soil, and in making soap, strong glass, and various chemical compounds

In `4lang` we can accomodate the chemistry only by explicit reference to the encyclopedia *potash* `@potassium_carbonate` which resolves to the WP article which in turn offers a wealth of information on the subject, and similarly for potassium hydroxide. But what to do with all this artisanal knowledge about industrial processes, that leaching wood ash produces lye, that caustic soda is used in glassmaking, that farmers feed the soil with potassium salts, and so on? We use a much simpler style of definition whereby *potash* is simply `salt, contain potassium` and consider the pain of invoking scientific theories in the midst of dictionary building to be self-inflicted.

`potash`

The key takeaway from this section is that once lexicography is freed of this burden, it is possible to formalize definitions to such a degree that we can automatically convert them into equations, in this case `potash is_a salt` and `potash contain potassium`. How a symbolic equation `A is_a B` or `A contain B` get translated to more conventional vector equations will be discussed in 2.3. The overall strategy of converting definitions to equations is made more concrete in a step by step fashion throughout the book, with a summary provided in 9.5.