Compositional Lexical Networks
A case study of the English spatial adjectives

Daniel Worthing

St Anne's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2021

Acknowledgements

My thanks are due first and foremost to Paul Chilton, for stepping in midway through this project and helping me complete it, at a time when I was tempted to give up. Without his detailed comments and general encouragement, this document would likely not exist. My gratitude also to Ash Asudeh, for crucial advice during the early stages of the project, to John Coleman, for facilitating an unconventional supervision arrangement and providing useful comments, and to all the other members of the LPP faculty who have helped me in various ways.

This research was made possible by a three-year studentship from the Arts and Humanities Research Council. I thank them for their generosity and the work they do in supporting the Humanities.

Throughout my time at Oxford, I have benefited from belonging to St Anne's College. An aspect of college life which I find extremely valuable is the opportunity to learn from students of other disciplines: informal discussions in the common room and dining hall have provided many of the ideas on which this thesis is based. The college has also been a source of support and welcome distraction when work became overwhelming. Thank you to all my friends at St Anne's, particularly: Barbar, Alex, Brittany, Virginia, Sebastian, Jae, Matthew, Guillaume, Adrian, Harry, Valeria, Nesrine and Vasillios.

I would also like to acknowledge friends outside of Oxford, whose company has sustained and rejuvenated me throughout this long project. A big thank you to all members of the 'Landan Team': Tom, Charlene, Jiameng, Alison, Peter, Laurence, Amy, Hannah, Jen and James. My gratitude also to Simon Conrad and Matt Farrugia.

Finally, my deepest gratitude to those without whom my life would be incomplete: Cassie Zhou, who maintains my enthusiasm with her constant cheerfulness and encouragement; my siblings Katy, Beccy and Bog, who are true privilege to live and grow alongside; and my parents, Sue and Rob, whose love, support, and continual moral example have made me who I am.

Abstract

Most words cannot be given a single precise definition, but instead consist of multiple senses related to each other like members of a family. In cognitive approaches to semantics, this kind of category is described by a *lexical network*, a diagram in which nodes represent senses and arrows represent sense connections. However, lexical network theory is not *compositional*: it does not explain how lexical networks are combined together to yield the meanings of phrases and sentences. The aim of this thesis is to develop lexical network theory in a formal, compositional setting. I argue that a traditional approach to formal semantics based on the simply-typed lambda calculus is not rich enough to implement lexical networks because it is unable to type the arrows which link word senses together. Instead, I propose replacing simple type theory with Martin-Löf Dependent Type Theory, and show how this allows for a fully compositional implementation of lexical networks. The resulting theory is applied to the description of the English spatial adjectives – *high*, *low*, *tall*, *long*, *short*, *deep*, *shallow*, *thick* and *thin*. These adjectives are an ideal starting point for studying the interaction between lexical and compositional semantics, since they have been studied extensively from both points of view. I illustrate how a compositional theory of lexical networks can provide an interface by which the insights of cognitive semantics can be imported into formal semantics, and vice versa.

Contents

Chapter 1

Introduction and Background

Contents

1.1 Motivation

Meaning in language can be approached from two perspectives starting from the level of individual words. The first perspective, that of lexical semantics, looks 'downwards' at the meanings of individual lexical items, how these are grounded in cognition and perception, and how they are connected together. Consider, for instance, all the situations which can be described as *running*: running a race, running for president, running a business, running a program, a running river, etc. The lexical/conceptual approach is concerned with questions such as: How many senses of *run* are there? How is the perceived similarity between different senses to be explained? Which senses are typical and which atypical? How are the different senses related to perception and bodily experience?, and so on. A common theoretical tool in lexical semantics is the *lexical network*, a way of mapping the senses of an individual word, or the relations between multiple words, using nodes and edges (e.g. Lakoff 1987, Norvig & Lakoff 1987, Brugman & Lakoff 1988, Tyler & Evans 2001).

The second perspective, that of compositional or formal semantics, looks 'upward' at how words combine together to yield the meanings of phrases and sentences, and how sentences contribute to an ongoing discourse. To illustrate, consider the two sentences: *Running a marathon takes 4 to 5 hours* and *As he ran, John enjoyed the evening sunset*. Both depend on the meaning of the verb *run*, but in very different ways: in the second sentence, *John* is understood as the agent of *ran*, whereas in the first sentence there is no particular person who does the running; the first sentence describes a generic situation whereas the second sentence describes a particular event; and so on. The goal of formal semantics is to specify a procedure which, given meanings for individual words, together with information about how they have been combined, can be used to derive the meaning of a larger expression (e.g. Montague 1973, Partee 1986, Heim & Kratzer 1998).

However, despite being studied in separate fields, lexical relations and composition are highly interactive. The relations which exist between words can be preserved by composition, projecting to the level of entire sentences. For example, just as *run* has multiple senses – locomotion, standing in an election, managing a project or business, etc. – so the phrase *run well* or *run badly* has multiple senses, each corresponding to a particular sense of *run*. A whole sentence will often exclude all but one sense from consideration, as in *John ran down the street*, but sometimes multiple senses remain at the level of the sentence, requiring context to disambiguate, as in *John ran last year* (which might refer either to competing in a race or standing in an election). At present, the interaction between polysemy and composition is not easily explained within lexical/conceptual semantics, with its focus on sub-lexical structure, nor within compositional/formal semantics, which tends to treat individual word meanings as atomic.

The aim of this thesis is to start with a small group of words and 'look in both directions at once', exploring both their lexical semantics and their compositional semantics, with a view to developing a general theory of polysemy-composition interaction. The words I have chosen for this purpose are the English spatial adjectives – *high*, *low*, *tall*, *long*, *short*, *wide*, *narrow*, *deep*, *shallow*, *thick* and *thin* – which are an important point of intersection for lexical and compositional theories. On the lexical side, there have been a number of detailed studies of spatial adjectives by authors such as Vandeloise (1988, 1993), Dirven & Taylor (1986), Linde-Usiekniewicz (2002) and Vogel (2004). On the compositional side, spatial adjectives fall into the class of *gradable adjectives*, which are the main topic of interest in the area known as *degree semantics* (e.g. Cresswell 1976, Klein 1980, von Stechow 1984, Kennedy 1999). Because of this overlap, spatial adjectives are an ideal starting point for developing a hybrid theory.

In brief, this thesis proposes a compositional theory of lexical networks. The meaning of a word is represented by a network in which nodes correspond to senses and arrows correspond to implications between senses. Implication is transitive, so if a network contains an arrow $S_1 \to S_2$ and an arrow $S_2 \to S_3$, then it also contains the composed arrow $S_1 \to S_3$. Every sense in a network has the same semantic type: one can have a network of noun senses, a network of gradable adjective senses, a network of transitive verb senses, a network of generalized quantifier senses, and so on. The type of a network's senses dictates how it can be composed with other networks – for instance, a network of noun senses and a network of intersective adjective senses may be combined using intersective adjective + noun composition. The result of composing two networks is a third network whose structure reflects that of the two input networks. In other words, composition can preserve the implications between senses, lifting not only senses but also sense connections.

Implementing a theory of this kind requires a different formal framework from what is usually adopted in formal semantics. Since Montague, the dominant approach to formal semantics has been to translate natural language expressions into the simply-typed lambda calculus, which is then interpreted in a set-theoretic model. The simply-typed lambda calculus is not well-suited to describing lexical networks, since although it provides types for *senses* (nouns, gradable adjectives, generalized quantifiers, etc.), it does not provide types for *arrows*. We therefore need to extend the type system by introducing *dependent types* and some associated constructions, resulting in what is known as Martin-Löf Dependent Type Theory (Martin-Löf 1984). This greatly enriches the type system, endowing it with its own internal logic, and allowing it to play the model-theoretic role usually played by set theory. The result is a quite different approach to formal semantics in which natural language expressions are interpreted directly in the type system itself (Sundholm 1989, Ranta 1994, Luo 2012, Chatzikyriakidis & Luo 2020).

The structure of this introductory chapter is as follows. Section 1.2 introduces the concept of a lexical network and distinguishes two major types of networks which have been proposed in the lexical semantics literature. Section 1.3 is a literature review of lexical approaches to spatial adjectives. Section 1.4 is an introduction to degree semantics, which describes the compositional behaviour of spatial adjectives and other gradable adjectives. Section 1.5 addresses the question of whether formal semantics and cognitive semantics are compatible. Finally, Section 1.6 provides an outline of the rest of the thesis.

1.2 Lexical networks

1.2.1 Introduction to lexical networks

The origin of lexical networks is Wittgenstein's (1958) notion of a *family resemblance* category. Wittgenstein noted that many lexical concepts, such as the concept of a *game*, cannot be given a precise definition in terms of necessary and sufficient conditions, since for any proposed condition (e.g. 'a game must have multiple players'), one can give a counterexample (e.g. patience/solitaire). Rather, different types of game are related to each other like members of a family, by a network of overlapping features. No member of the category need possess all of the features, nor is there necessarily a single feature which all members of the category possess. Rather, certain combinations of the features are sufficient for membership in the category, and a member can always possess more features than strictly required. For instance, basketball possesses many conditions associated with being a game, including multiple players, competitiveness, enjoyment, rule following, physical activity, and so on, whereas solitaire satisfies only a few, namely enjoyment and rule following.

The family resemblance concept was extended by Eleanor Rosch (1973, 1975, 1999, etc.), who added the idea that *members which satisfy more conditions are judged to be more central/prototypical.* Rosch conducted numerous experiments showing that semantic judgements were subject to typicality effects. In one experiment, college students were asked to rate objects on a scale from 1 to 7 based on whether they were a good example of *furniture*. Results ranged from very good examples, such as *chair* and *sofa*, to middling examples such as *television*, to bad examples such as *fan* and *counter*. Similar typicality effects were shown for other categories, such as *fruit*, *vehicle*, *weapon*, and so on. Rosch's explanation for this was that participants were judging an object's degree of membership in a category by comparing it to a prototypical exemplar, which she identified with the most central node in the lexical network. The more conditions a member shares with the central node, the more prototypical it is judged to be.

Following the insights of Wittgenstein and Rosch, a major aim of lexical semantics has been to identify distinct senses of a word and provide a 'map' of how they are related to each other. This information is represented using a collection of nodes and arrows, where the nodes represent senses and the arrows represent sense connections. Each network is a miniature cognitive theory: it is intended to convey something about the lexicon as represented in the mind of a language user (Lakoff 1987, Tyler & Evans 2001). The structure of a network embodies claims about which senses people will judge to be typical and which atypical; which pairs of senses people judge to be similar and which dissimilar; and so on. In so far as predictions like these are found to be correct, the lexical network captures important generalizations.

Early lexical networks were based on Fillmore's (1976) concept of a *frame*. A frame is a collection of knowledge about the meaning of a word, which becomes available whenever that word is encountered, and relates it to other words in a language user's lexicon. For example, the verb *sell* evokes the frame COMMERCIAL TRANSFER, which includes knowledge about sellers, buyers, goods, money, shops, pricing, and so on – all the concepts involved in understanding and using the word.

The same frame is also evoked by numerous other words, such as *buy*, *cost*, *shop*, *barter*, and *supermarket*. Formally speaking, a frame is understood as a collection of roles with associated types, as well as knowledge about how the various roles relate to each other. For instance, in the COMMERCIAL TRANSACTION frame, the buyer and seller roles are of type *Person*, whereas the shop is of type *Location*, and the transaction itself is of type *Event*. For a recent incarnation of Fillmore's frame semantics, see the online lexical database FrameNet[1].

Other lexical networks are based on *image schemas*. An image schema is a reoccurring structure which is abstracted from basic perceptual and motor experiences and used to organize higher-level cognitive abilities such as conceptual reasoning and language understanding (Johnson 1987, Lakoff 1987). Examples of image schemas include OBJECT, VERTICAL AXIS, PATH, SURFACE, CONTAINER, CONTACT and BLOCKAGE (Johnson, ibid.). They are often considered to arise from experiences relating to one's own body – the VERTICAL AXIS schema derives from the experience of standing upright, the PATH schema from the experience of moving through space, the CONTACT schema from the experience of touching objects, and so on. Unlike frames, image schemes are often represented in geometric or topological terms, using diagrams rather than symbolic features or attributes.

1.2.2 Implication networks

An implication network is a lexical network in which all of the arrows represent implication. Other names for this type of network include: *family resemblance category*, *radial category* (Lakoff 1987), *preference rule system* (Jackendoff 1983) and *cluster concept* (Jackendoff 2002). A simple example of an implication network is Jackendoff's (2002, p. 353) analysis of the verb *climb*. Consider the following sentences:

(1) a. Bill climbed (up) the mountain. [rising and clambering]
b. The snake climbed (up) the tree. [rising only]
c. Bill climbed down the mountain. [clambering only]
d. ?The snake climbed down the tree. [neither]

Jackendoff analyses *climb* as involving two independent conditions: the individual is travelling upwards ('rising'), and the individual is moving with grasping motions ('clambering'). An event which satisfies both rising and clambering, as in (a), is a prototypical instance of *climb*; an event which satisfies only rising or only clambering, as in (b) and (c) respectively, is a less typical instance; and an event which satisfies neither condition, as in (d), cannot be described as *climbing* at all. Jackendoff writes:

> One possible account of this would be to say that *climb* is ambiguous or polysemous between the readings 'rise' and 'clamber'. But this does violence to intuition: [the sentence *Bill climbed the mountain*] is not ambiguous between these two senses. Rather, other things being equal,

[1] `https://framenet.icsi.berkeley.edu/fndrupal/`

> it satisfies both of them. Another possible account would be to say that the meaning of *climb* is the logical disjunction of the two senses: 'rise *or* clamber'. But this is too crude: a disjunction isn't "more prototypically satisfied" if both disjuncts are true, as *climb* is. (ibid., p. 353-354)

One way to express the prototype effect identified by Jackendoff is in terms of a simple lexical network with three senses – rising and clambering, rising only and clambering only. The structure of the network is shown below:

(2)
$$\begin{tikzcd} \text{climb}_{\text{rise}} & & \text{climb}_{\text{clamber}} \\ & \text{climb}_{\text{rise, clamber}} \arrow[ul] \arrow[ur] & \end{tikzcd}$$

This is an implication network because the arrows correspond to implications or inclusions: the leftmost arrow represents the fact that any instance of $\text{climb}_{\text{rise, clamber}}$ is also an instance of $\text{climb}_{\text{rise}}$; the rightmost arrow represents the fact that any instance of $\text{climb}_{\text{rise, clamber}}$ is also an instance of $\text{climb}_{\text{clamber}}$. Many implication networks display a similar radial structure, with multiple peripheral senses diverging from a common central prototype.

To give a slightly more complex example of an implication network, consider Norvig's (1989) analysis of the noun *meat*. According to Norvig, the central sense of *meat* is:

1. The edible muscle tissue of a mammal (especially a bovine), when stripped from a mammal and intended for consumption

This can then be weakened in several directions:

2. Allow fowl as well as mammals.
3. Allow fish as well as fowl and mammals.
4. Allow organ meat as well as muscle tissue.
5. Allow skin as well as muscle tissue.
6. The interior edible part of any food (e.g. *coconut meat*).
7. (metaphor) The core or essence of something.

To justify that each of these senses is distinct, we must be able to find a sentence which includes only what is covered by that sense, whilst excluding everything else. The following sentences can serve to distinguish senses 1-7:

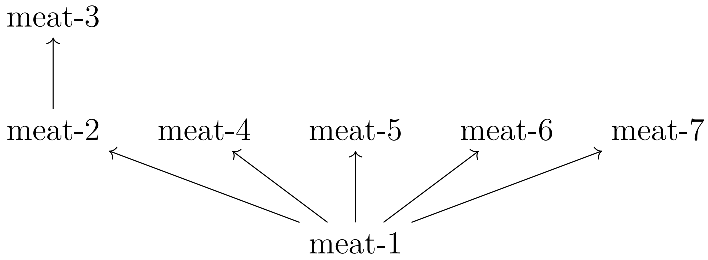
(3) a. Bring me beef, not chicken; I only eat meat_1.
  b. Bring me chicken, not fish; I only eat meat_2.
  c. Bring me fish, not rice; I only eat meat_3.
  d. Bring me liver, not skin; I only eat meat_4.
  e. Bring me skin, not liver; I only eat meat_5.

f. I'll eat the flesh of the apple, but not the skin; I only eat meat$_6$.

g. This part of the talk is just the introduction; I'm only interested in the meat$_7$.

According to Norvig, all of the peripheral senses are derived directly from sense 1, except for sense 3 which is derived from sense 2. This gives the following radial network:

(4)



As before, this is an implication network. For example, the arrow from meat-1 to meat-2 represents the fact that any instance of meat-1 is also an instance of meat-2. Likewise, any instance of meat-2 is also an instance of meat-3. Since implication is transitive, there is an implicit arrow from meat-1 to meat-3 which is not drawn on the diagram.

An important point to note about implication networks is that all of the senses are of the same syntactic type: for instance, all of the senses in the *climb* network are verbs and all of the senses in the *meat* network are nouns. It therefore makes sense to talk about adjective networks, noun networks, preposition networks, and so on. Moreover, the arrows in an implicational networks are idiosyncratic and do not express patterns which apply across the lexicon. Each implication network must be acquired separately by the learner: there is no sense in which the arrow from meat-1 to meat-2, for example, can be predicted on the basis of a general rule. As Lakoff puts it,

> variants [in an implication network] are not generated from the central model by general rules; instead, they are extended by convention and must be learned one by one. But the extensions are by no means random. The central model determines the possibilities for extensions, together with the possible relations between the central model and the extension models. (Lakoff 1987, p. 91)

Other examples of implication networks include Lakoff's (1987) *mother* network; Coleman & Kay's (1981) description of the verb *lie*; Jackendoff's analysis of the verb *see* (1983) and the preposition *in* (2002); and Zwarts's (2004) description of the preposition *around*.

Almost all of the implication networks proposed in the literature have a radial structure, consisting of a single prototype surrounded by peripheral senses. Authors have tended to assume that networks must be radially divergent because a number of key examples have that structure, notably categories such as FURNITURE and BIRD which were discussed by Rosch. As a result, authors often neglect parts of the network where senses *converge*. This mistake is reinforced by the idea that peripheral senses are ‘derived from the prototype’, since the term ‘derive’ suggests divergence: two senses cannot be ‘derived’ to the same thing. A term more suited to implication networks would be ‘weaken’: a prototype like meat-1 is ‘weakened’ in various ways, giving rise to more general senses. Convergence happens when two specific senses can be weakened to a more general sense which subsumes both of them.

To illustrate, a major problem with Norvig’s *meat* network is that it contains no sense which covers the intended meaning of the following sentence:

(5) I don’t eat meat, I’m a vegetarian.

The usual interpretation of *meat* here is very general: ‘any edible part, interior or otherwise, of any animal’. Norvig’s network does not contain this maximally general sense, covering only specific cases, e.g. skin as well as muscle tissue, organs as well as muscle tissue, fish as well as land animals, and so on. This is because Norvig does not consider the possibility of *convergence*: senses only ever become different from each other, never joining back up. The sense intended in (5), which I shall refer to as meat-8, appears as the common join of meat-3, meat-4 and meat-5, as follows:

(6)
$$
\begin{tikzcd}
 & \text{meat-8} & & & \\
\text{meat-3} \arrow[ur] & & & & \\
\text{meat-2} \arrow[u] & \text{meat-4} \arrow[uu] & \text{meat-5} \arrow[uul] & \text{meat-6} & \text{meat-7} \\
 & & \text{meat-1} \arrow[ull] \arrow[ul] \arrow[u] \arrow[ur] \arrow[urr] & &
\end{tikzcd}
$$

We can arrive at meat-8 by either (i) starting at meat-3 and allowing organ meat and skin, (ii) starting at meat-4 and allowing skin, fowl and fish, or (iii) starting at meat-5 and allowing organ meat, fowl and fish. Just as meat-1 is the strongest or most prototypical sense of *meat*, meat-8 is the weakest or least typical sense. (The reason why meat-8 is the default interpretation of (5) is the presence of negation, which reverses the direction of implication, making meat-8 the prototype.)

The general structure of an implication network is a *partial order*. In other words, given some implication network $\mathcal{N}$:

(7)
- For any two senses S_1, S_2 in $\mathcal{N}$, there can be at most one arrow $S_1 \to S_2$.
- Any sense S in $\mathcal{N}$ has an implicit arrow $S \to S$ (a sense always implies itself).
- For any two senses S_1, S_2 in $\mathcal{N}$, if $S_1 \to S_2$ and $S_2 \to S_1$ then $S_1 = S_2$ (two senses cannot be related in both directions).
- For any three senses S_1, S_2, S_3 in $\mathcal{N}$, if $S_1 \to S_2$ and $S_2 \to S_3$ then there is an implicit arrow $S_1 \to S_3$ (implication is transitive) .

Given two senses S_1 and S_2 in an implication network, we can talk about lower and upper bounds. An lower bound for S_1 and S_2 is a sense S' such that $S' \rightarrow S_1$ and $S' \rightarrow S_2$; conversely, an upper bound is a sense S'' such that $S_1 \rightarrow S''$ and $S_2 \rightarrow S''$. The most immediate lower bound, if unique, is called the *meet* and the most immediate upper bound, if unique, is called the *join*. For example, in the network in (6), the meet of meat-2 and meat-5 is meat-1, whereas the join of meat-2 and meat-5 is meat-8. Two senses with a common meet are derived from the same prototype, whereas two senses with a common join are subsumed by the same generalization.

Zeugma is a kind of semantic anomaly caused by trying to coordinate two distinct senses of the same word, as in:

(8) ? The dining table was very long and so was the meal.

This sentence strikes us as strange because it involves two distinct senses of *long* – a spatial sense and a temporal sense. Notice that the problem disappears in the following sentence:

(9) The dining table was very long$_1$ and the meal was very long$_2$.

which sounds stilted but is not unacceptable in the same way. The problem with (8) is that a single occurrence of *long* cannot be interpreted in both a spatial and a temporal sense. Two senses without a common join will always give rise to zeugma when coordinated. For instance, returning to the *meat* network, it sounds strange to say:

(10) ? John ate the [chicken and coconut] meat.

because there is no sense of *meat* which subsumes both meat-2 and meat-6. In other words, there is no single interpretation of *meat* which yields a consistent interpretation for the whole sentence. However, it does make sense to say:

(11) John ate the [skin and liver] meat.

since meat-4 and meat-5 are both subsumed by meat-8. We can use the presence or absence or zeugma as a diagnostic for whether two senses in an implication network share a common join. To put it another way, the structure of an implication network makes predictions about which senses give rise to zeugma when coordinated and which do not.

1.2.3 Derivation networks

All of the lexical networks discussed in this thesis are implication networks. However, there exists an entirely separate class of networks which are also frequently called ‘lexical networks’; I shall refer to these as *derivation networks*. Unlike implication networks, in which all of the senses have the same syntactic type, the senses in a derivation network can belong to distinct syntactic types. The arrows in a derivation network do not represent implication, but rather processes of meaning extension which are analogous to the operations of derivational morphology. These processes can be more or less productive, some occurring throughout the lexicon, others limited to specific cases. To give an example of a derivational network, consider the following meanings of the word *smoke*, discussed by Jackendoff (1996):

(12) smoke$_a$ 'wispy substance'

X smokes$_b$ 'X gives off smoke$_a$'

X smokes$_c$ Y 'X causes Y to smoke$_b$, where Y is a a cigarette, pipe, etc., by puffing'

X smokes$_d$ 'X smokes$_c$ something'

X smokes$_e$ Y 'X causes smoke$_a$ to go into Y, where Y is meat or fish, by hanging over a fire in an enclosed space'

As Jackendoff notes, the nominal sense smoke$_a$ forms a kind of semantic core, with the other senses arranged in a branching chain as follows:

$$\begin{array}{lllllll} \text{smoke}_\text{a} & \longrightarrow & \text{smoke}_\text{b} & \longrightarrow & \text{smoke}_\text{c} & \longrightarrow & \text{smoke}_\text{d} \\ \big\downarrow & & & & & & \\ \text{smoke}_\text{e} & & & & & & \end{array} \tag{13}$$

Notice that this is not an implication network. For instance, the arrow from smoke$_a$ to smoke$_b$ does not mean 'any instance of smoke$_a$ is an instance of smoke$_b$': it is not clear what this would even mean, since smoke$_a$ is a kind of substance whereas smoke$_b$ is a kind of event. Rather, the arrow indicates that the meaning of smoke$_b$ depends on or incorporates the meaning of smoke$_a$. Unlike in an implication network, the nodes can have distinct syntactic types: smoke$_a$ is a noun, smoke$_b$ is an intransitive verb, smoke$_c$ is a transitive verb, and so forth.

Unlike implicational networks, the concept of typicality does not seem to apply to derivational networks. Is smoke$_a$ a more typical sense than smoke$_b$? It is difficult to answer this question, because they belong to completely different syntactic and semantic types. To be sure, smoke$_a$ is more *fundamental* than smoke$_b$, in the sense that the definition of smoke$_b$ relies on that of smoke$_a$. Similarly, smoke$_b$ is more fundamental than smoke$_c$, which is more fundamental than smoke$_d$, and so on. However, fundamentality is not the same thing as typicality. For a sense S_1 to be more typical than a sense S_2, a speaker should consistently prefer S_1 over S_2 in a context where both senses are possible. However, because senses in a derivation network can be distinguished syntactically, this criterion cannot be applied.

An important characteristic of derivation polysemy is that the connections linking together the different alternatives can have parallels in other lexical items. Some of these connections are highly productive, occurring throughout the lexicon; others are found in only a few other cases. Jackendoff lists the following examples:

(14)
- smoke$_a$ $\to$ smoke$_b$ (N $\mapsto$ 'give off N'): *steam*, *sweat*, *smell*, *flower*
- smoke$_b$ $\to$ smoke$_c$ (V $\mapsto$ 'cause to V'): *open*, *break*, *roll*, *freeze*
- smoke$_c$ $\to$ smoke$_d$ (V $\mapsto$ 'V something'): *eat*, *drink*, *read*, *write*
- smoke$_a$ $\to$ smoke$_e$ (N $\mapsto$ 'put N into/onto something'): *paint*, *butter*, *water*, *powder*, *steam*

Although all the connections in the *smoke* network are also found in other words, the particular combination which *smoke* participates in is unique. What is more, each connection also adds some idiosyncratic information which is not present in parallel versions: for instance, smoke_c means not only 'cause to smoke', but 'cause to smoke by puffing, whilst taking smoke into the mouth'. As a result, the derivation network for each individual lexical item must be learned separately.

The reason for the term 'derivation polysemy' is that the connections in a derivation network resemble those of derivational morphology. As Jackendoff writes:

> [The various senses of *smoke*] are not more closely related to each other than they are to, say, *smokey* and *smoker* (the later meaning either a person who smokes_c, or a vessel in which one smokes_e things). And in a morphologically richer language than English they might not be phonologically identical (say *outsmoke* for [smoke_b] and *ensmoke* for [smoke_e]). (Jackendoff 1996, p. 113).

In other words, a connection in a derivation network behaves similar to a phonologically null derivational morpheme. This suggests that whatever mechanism is used to describe derivational morphology should also be applied to derivational networks. Jackendoff suggests that patterns such as this should be handled by *lexical rules*, semiproductive rules which capture generalizations inside the lexicon. For example, the lexical rule relating smoke_a and smoke_b might be expressed informally as follows:

(15) A noun N denoting a substance which is pronounced /X/ may be related to a verb also pronounced /X/, meaning 'give off N'

Such rules do not apply in all cases where their application conditions are met: there is no verb *water* meaning 'give off water'. Nor do they transparently predict the meaning of the derived form: the meaning of *sweat* 'give off sweat' is quite different from the meaning of *smoke* 'give off smoke', for example. Rather, one must individually learn cases in which the rule does apply, along with any idiosyncratic aspects of meaning associated with the derivation.

A single word may participate in both implication polysemy and derivation polysemy. For example, in their study of the preposition *over*, Brugman & Lakoff (1988) distinguish the following three senses (among several others):

(16) over_a 'located in an upwards direction from X', e.g. *the painting is over the mantle*

over_b 'path passing through a location over_a X', e.g. *the plane is flying over the hill*

over_c 'location at end of path over_b X', e.g. *Sam lives over the hill*

This is clearly a case of derivation polysemy. Over_b is derived from over_a by a rule which takes a location description and returns a path description that passes through the location, as in *John walked <u>in</u> the room*, *Mary jumped <u>out</u> the window*, and so on. Over_c is in turn derived from over_b by a rule which takes a path description and returns a location description corresponding to the endpoint of the path, as in *John was standing across the hall*, *the house is 2m along the river*, etc. We therefore have the following derivation network:

(17) $\text{over}_{\text{a}} \longrightarrow \text{over}_{\text{b}} \longrightarrow \text{over}_{\text{c}}$

However, as Jackendoff (1996, p. 24-25) points out, there is also a different kind of polysemy associated with the basic locative sense of *over*, which he illustrates as follows:

(18) a. The blimp is over the field. [vertical separation]
b. The cloth is over the table. [two-dimensional covering]

One feels that these two senses ought to be connected, but the connection is clearly not a matter of derivation, since neither is more fundamental than the other. Moreover, they belong to the same syntactic type, and the connection between them is idiosyncratic, suggesting that they are connected implicationally rather than derivationally. Since neither implies the other, they must be connected through additional senses. Assuming that the blimp in (18) is idealized as a zero-dimensional point whereas the cloth is idealized as a two-dimensional surface, I would propose the following tentative analysis:

(19)	$\text{over}^{\text{a}}_{\text{0d, vert}}$	vertical separation of pointlike object, e.g. *the blimp is over the field*
	$\text{over}^{\text{a}}_{\text{vert, cover}}$	vertical separation and two-dimensional covering, e.g. *the awning is over the patio*
	$\text{over}^{\text{a}}_{\text{vert}}$	general vertical separation, e.g. *the blimp is over the field and so is the cloud layer*
	$\text{over}^{\text{a}}_{\text{cover}}$	general two-dimensional covering, e.g. *the board is over the hole*

These would then be connected by implicational links as follows:

(20)
$$\begin{tikzcd}
\text{over}^{\text{a}}_{\text{0d, vert}} \arrow[dr] & \\
& \text{over}^{\text{a}}_{\text{vert}} \\
\text{over}^{\text{a}}_{\text{vert, cover}} \arrow[ur] \arrow[dr] & \\
& \text{over}^{\text{a}}_{\text{cover}}
\end{tikzcd}$$

The important point here is that the implicational network in (20) is not the same as the derivational network in (17); nor are they both part of some larger network. Rather, all of the senses in the implicational network pertain to a single node in the derivational network, over_{a} (hence the superscript a). The kind of polysemy represented by the implicational network is more fine-grained.

Let us temporarily refer to the nodes in an implicational network as *microsenses* and the nodes in a derivational network as *macrosenses*. Some authors represent microsenses and macrosenses as part of a single network. For example, the *over* network proposed by Brugman & Lakoff (1988) (as well as a more recent version by Tyler & Evans 2003) includes the senses over_{a}, over_{b} and over_{c} together with a separate two-dimensional covering sense corresponding roughly to $\text{over}_{\text{cover}}$. However, collapsing the two kinds of network in this way creates confusion and misses out on generalizations. For example, when we go from the locative macrosense over_{a} to the path macrosense over_{b}, every microsense gets mapped individually, giving rise to a new network of microsenses, as follows:

(21)
$$\begin{tikzcd}
\text{over}^{\text{a}}_{\text{0d, vert}} \arrow[dr] & & & \text{over}^{\text{b}}_{\text{0d, vert}} \arrow[dr] & \\
& \text{over}^{\text{a}}_{\text{vert}} & \arrow[r, mapsto] & {} & \text{over}^{\text{b}}_{\text{vert}} \\
\text{over}^{\text{a}}_{\text{vert, cover}} \arrow[ur] \arrow[dr] & & & \text{over}^{\text{b}}_{\text{vert, cover}} \arrow[ur] \arrow[dr] & \\
& \text{over}^{\text{a}}_{\text{cover}} & & & \text{over}^{\text{b}}_{\text{cover}}
\end{tikzcd}$$

where examples of the over_{b} microsenses are given below:

(22)	$\text{over}^{\text{b}}_{\text{0d, vert}}$	The blimb flew over the field.
	$\text{over}^{\text{b}}_{\text{vert, cover}}$	The cloud layer passed over the field.
	$\text{over}^{\text{b}}_{\text{vert}}$	The blimp and the cloud layer passed over the field.
	$\text{over}^{\text{b}}_{\text{cover}}$	Her eyelashes moved over her eyes.

In other words, derivation networks can preserve the structure of implication networks. This kind of behaviour is another motivation for distinguishing between macrosenses and microsenses. For the remainder of this thesis, I shall use the term *sense* to refer to microsenses, unless otherwise specified.

1.2.4 Implication networks and composition

A feature of implication networks which is of central importance to this thesis is the way they behave under composition. This closely resembles the interaction between implication and derivation discussed in the previous subsection. Recall Jackendoff's *climb* network, a radial network consisting of the prototype $\text{climb}_{\text{rise, clamber}}$ and the two peripheral senses $\text{climb}_{\text{rise}}$ and $\text{climb}_{\text{clamber}}$:

(23)
$$\begin{tikzcd}
\text{climb}_{\text{rise}} & & \text{climb}_{\text{clamber}} \\
& \text{climb}_{\text{rise, clamber}} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

As discussed, the two arrows represent weakening the prototype by dropping the clambering condition or the rising condition respectively. Now consider the sentence *John climbed quickly.* This is compatible with all three senses of *climb*, and therefore has three distinct interpretations: $[\text{John climbed quickly}]_{\text{rise}}$, $[\text{John climbed quickly}]_{\text{clamber}}$ and $[\text{John climbed quickly}]_{\text{rise, clamber}}$. Notice that, just as the most prototypical interpretation of *climb* is $\text{climb}_{\text{rise, clamber}}$, the most prototypical interpretation of *John climbed quickly* is $[\text{John climbed quickly}]_{\text{rise, clamber}}$. Moreover, just as $\text{climb}_{\text{rise, clamber}}$ can be weakened by dropping the clambering condition or the rising condition, so $[\text{John climbed quickly}]_{\text{rise, clamber}}$ can be weakened in the same way, giving either $[\text{John climbed quickly}]_{\text{rise}}$ or $[\text{John climbed quickly}]_{\text{clamber}}$. In other words, the interpretation of *John climbed quickly* is a network with exactly the same structure as *climb*:

(24)
$$\begin{tikzcd}
\text{climb}_{\text{rise}} & & [\text{John climbed quickly}]_{\text{rise}} \\
\text{climb}_{\text{rise, clamber}} \arrow[u] \arrow[d] \arrow[rr, mapsto] & & [\text{John climbed quickly}]_{\text{rise, clamber}} \arrow[u] \arrow[d] \\
\text{climb}_{\text{clamber}} & & [\text{John climbed quickly}]_{\text{clamber}}
\end{tikzcd}$$

As shown, the context *John X-ed quickly* preserves the structure of implication, allowing the arrows in the *climb* network to become entailments between entire sentences.

Some kinds of composition preserve implication, but in the opposite order. For example, consider the sentence *no one climbed.* As before, this has three possible interpretations corresponding to the different senses of the *climb* network: $[\text{no one climbed}]_{\text{rise, clamber}}$, $[\text{no one climbed}]_{\text{rise}}$ and $[\text{no one climbed}]_{\text{clamber}}$. However, the arrows linking them appear in the opposite order, as follows:

(25)
$$\begin{tikzcd}
\text{climb}_{\text{rise}} & & [\text{no one climbed}]_{\text{rise}} \arrow[d] \\
\text{climb}_{\text{rise, clamber}} \arrow[u] \arrow[d] \arrow[rr, mapsto] & & [\text{no one climbed}]_{\text{rise, clamber}} \\
\text{climb}_{\text{clamber}} & & [\text{no one climbed}]_{\text{clamber}} \arrow[u]
\end{tikzcd}$$

In other words if no one climbed in the rising sense, then this implies that no one climbed in the rising and clambering sense; likewise if no one climbed in the clambering sense then this also implies that no one climbed in the rising and clambering sense. When implication is reversed in this way, meets become joins and joins become meets. The most prototypical sense in the original network, in this case $\text{climb}_{\text{rise, clamber}}$ yields the least typical sense in the derived network, and vice versa.

The preservation of implication by composition is referred to as *monotonicity*. Suppose we have a function F which takes an implication network and returns a new implication network. F is said to be *monotone* iff:

(26) $S_1 \to S_2$ implies $F(S_1) \to F(S_2)$ for all senses S_1, S_2

that is, it preserves arrows without reversing their directions. Conversely, F is said to be *anti-monotone* iff:

(27) $S_1 \to S_2$ implies $F(S_2) \to F(S_1)$ for all senses S_1, S_2

that is, it preserves arrows but reverses their directions. As we have seen, the context *John X-ed quickly* lifts the *climb* network monotonically, whereas the quantifier *no one* lifts the *climb* network anti-monotonically. The distinction between the monotone and anti-monotone patterns is usually discussed in the context of generalized quantifiers and the distribution of negative polarity items. For instance, it is a well-known observation that anti-monotone quantifiers license negative polarity items like *ever*, *any*, *at all*, *somewhat*, and so on, whereas monotone quantifiers do not (Fauconnier 1975, Ladusaw 1979). When the meaning of words is represented by implication networks, monotonicity is no longer just a property of generalized quantifiers: any network which acts on another may do so monotonically or anti-monotonically.

Not all kinds of composition are arrow-preserving, i.e. monotone or anti-monotone. For example, consider the sentence *Mary enjoys climbing.* As before, we can identify three senses corresponding to the three senses of *climb*: [Mary enjoys climbing]$_{\text{rise, clamber}}$, [Mary enjoys climbing]$_{\text{rise}}$ and [Mary enjoys climbing]$_{\text{clamber}}$. However, these three senses are no longer connected by implicational arrows. The fact that Mary enjoys climbing in the 'rising and clambering' sense does not imply that she enjoys climbing in the 'clambering only' sense, nor the reverse. The context *Mary enjoys X-ing* lifts only the senses of *climb*, deleting the information about how they are connected together:

(28)
$$\begin{tikzcd} \text{climb}_{\text{rise}} & & \text{[Mary enjoys climbing]}_{\text{rise}} \\ \text{climb}_{\text{rise, clamber}} \arrow[u] \arrow[d] \arrow[rr, mapsto] & & \text{[Mary enjoys climbing]}_{\text{rise, clamber}} \\ \text{climb}_{\text{clamber}} & & \text{[Mary enjoys climbing]}_{\text{clamber}} \end{tikzcd}$$

A function with this behaviour is called *non-monotone*. We can think of it as taking a network and returning an unstructured set.

Following composition, some parts of an implicational network may collapse due to inconsistency. This is the case even in implication-preserving contexts. For example, consider the directional adverb *down*. Generally speaking, composition of *down* with an event description is monotonic. However, when *down* is combined with the climb network, as in *John climbed down*, the only surviving sense is climb$_{\text{clamber}}$ because climb$_{\text{rise, clamber}}$ and climb$_{\text{rise}}$ both necessarily involve upwards motion. Since intersective adverb composition in general preserves network structure, we must assume that anomalous senses are detected and removed following composition by a process of inconsistency detection, as follows:

(29)
$$
\begin{tikzcd}
\text{climb}_{\text{rise}} & & {[\text{John climbed down}]_{\text{rise}}} \\
\text{climb}_{\text{rise, clamber}} \arrow[u] \arrow[d] \arrow[rr, mapsto] & & {[\text{John climbed down}]_{\text{rise, clamber}}} \arrow[u] \arrow[d] \\
\text{climb}_{\text{clamber}} & & {[\text{John climbed down}]_{\text{clamber}}}
\end{tikzcd}
$$

$$
\longmapsto \qquad [\text{John climbed down}]_{\text{clamber}}
$$

As indicated, one can first derive the interpretation network for *John climbed down*, and then prune the resulting network by removing inconsistent senses. To recognise this inconsistency, the interpreter must identify a conflict between the semantics of *down* and the rising condition present in and $\text{climb}_{\text{rise}}$. $\text{Climb}_{\text{rise,clamber}}$ can then be ruled out automatically because it has an arrow into an inconsistent sense.

A common observation regarding prototypes and composition is that the most prototypical interpretation of an expression is not always a function of the most prototypical interpretations of its parts (Osherson & Smith 1981, Kamp & Partee 1995, Fodor & Lepore 1996). This is often called the *pet fish* phenomenon, since a typical *pet fish* is neither a typical *pet* nor a typical *fish*. As critics of prototype theory have pointed out, the pet fish phenomenon is problematic for a theory in which meanings are simply identified with prototypes, because whatever meanings are they are supposed to be compositional, and prototypes are clearly not compositional. The lexical network point of view avoids this criticism because it does not identify meanings with prototypes; rather, a prototype is just a maximal/initial sense in a lexical network. Assuming that lexical networks are subject to consistency checks following composition, it is no surprise that maximal interpretation of a composed network is not in general given by the maximal interpretations of its parts. (This is apparent from (29).) In the combination *pet fish*, the maximal interpretation of *pet* and the maximal interpretation of *fish* are presumably incompatible, so the resulting sense is pruned following composition.

A more serious challenge to the account of network composition sketched above is the claim by some authors that prototypes are *never* preserved by composition, and therefore have no role to play in formal semantics. Connolly et al. (2007) criticise what they call the *default to the stereotype* (DS) view of concept combination. Connolly et al. show that people's confidence in a sentence like *apples are round* is usually higher than their confidence in a sentence like *purple apples are round*. This contradicts DS, which predicts that their confidence in both sentences should be the same, the typical shape of *purple apple* being the same as that of *apple*. The authors conclude:

> Subjects do not default to the stereotypes of the conjuncts of a combined concept when interpreting a novel combination. This is hardly surprising since *the more words/concepts combine, the less likely it becomes that they*

refer to things that satisfy their stereotypes. We typically use adjectival modifiers in noun phrases when we are talking about something *other than* typical instances of the head noun. (ibid., p. 15)

As a replacement to DS, Connolly et al. propose a Classical Theory of composition, according to which "concepts remain inert under combination", with "separate machinery that introduces pragmatic and knowledge-dependent inferences" (ibid., p. 1).

Connolly et al. assume that prototypes are statistical generalizations about the properties which members of a category tend to have. This is indeed how prototypes are understood in some formulations of prototype theory (e.g. Smith et al. 1988). However, from the lexical network point of view, the notion of prototype is *logical* rather than statistical: the prototype is the *logically strongest* sense which implies all of the others, not the average or most common sense. In fact, the prototypical sense is always less common than any of the peripheral senses derived from it, because it is more specific. For instance, the prototypical sense of *meat* refers to the muscle tissue of a land animal, cooked and intended for consumption, but most instances of *meat* do not fall into this category. Similarly, the fact that the strongest interpretation of *purple apple* is one in which the apple is round does not imply that a *purple apple* chosen at random is likely to be round. Connolly et al.'s objection therefore applies only to the statistical notion of prototype, not the logical notion.

From the lexical network point of view, prototypes support inference, not because of their statistical likelihood, but because of pragmatic considerations. For example, suppose you hear the sentence

(30) We're having meat for dinner.

It is natural to infer that the speaker has in mind the flesh of a land animal, although there is a much more inclusive (and therefore more likely) interpretation which refers to the flesh of a land animal *or fish*. This inference is licensed by the standard Gricean assumption that the speaker is being as informative as possible, the same assumption which is needed to generate scalar implicatures (Grice 1975, Horn 1984). The idea that the interpreter should assume the strongest consistent interpretation of a sentence, unless there is some good reason why this interpretation does not apply, is sometimes known as the *Strongest Meaning Hypothesis* (Dalrymple et al. 1994, Winter 1996).

1.3 Lexical approaches to spatial adjectives

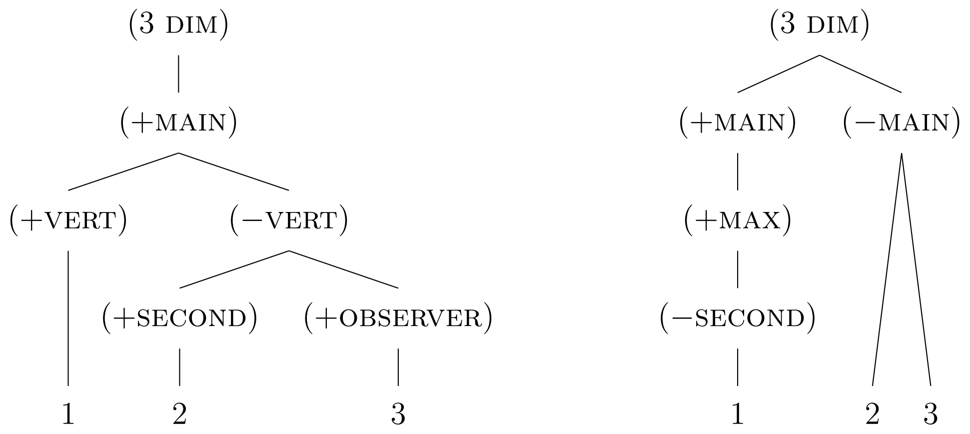
This section is a brief introduction to previous research on the lexical semantics of spatial adjectives. Two distinct categories of approaches can be identified: those based on distinctive features, and those based on ideas from cognitive linguistics. Featural approaches (e.g. Bierwisch 1967, Lang 1989, 2001, Weydt & Schlieben-Lange 1998, Stolz 1996) describe spatial adjectives using combinations of semantic features, which are also present in the lexical entries of nouns. A spatial adjective can only combine with a noun if the features in their lexical entries match up. The presence or absence of features allows nouns to be classified into different *dimensional types*, each of which admits a different combination of spatial adjectives. Cognitive approaches (e.g. Vandeloise 1988, 1993, Dirven & Taylor 1986, Goy 2002, Vogel 2004), on the other hand, are concerned with mapping the various senses of spatial adjectives in networks and grounding their meanings in perceptual/bodily experience.

1.3.1 Bierwisch's axis trees

The early work of Manfred Bierwisch (1967) was one of the first attempts to formally describe the properties of spatial adjectives. Like other authors at that time, Bierwisch was influenced by the phonological methods of the Prague linguistic circle, who described phonemes using collections of binary distinctive features. The concept of distinctive feature was imported into lexical semantics, for example by Katz & Fodor (1963), who call them *semantic markers*. Just as in phonology, the aim was to "construct a metatheory which contains an enumeration of the semantic markers from which the theoretical vocabulary of each particular semantic theory is drawn." Working within this tradition, Bierwisch proposed the following basic features which describe the axes of an object:

(31)	(*n* DIM)	encodes the dimension of an axis: 1, 2 or 3.
	(±MAIN)	an object's 'significantly extended' axes are (+MAIN), all other axes are (−MAIN)
	(±VERT)	vertical axes are (+VERT), horizontal axes are (−VERT)
	(±MAX)	a maximal axis is (+MAX), all other axes are (−MAX)
	(±SECOND)	a secondary axis is (+SECOND), other axes are (−SECOND)
	(±INHERENT)	an axis intrinsic to an object is (+INHERENT), an extrinsic axis is (−INHERENT)
	(±OBSERVER)	an axis aligned with the speaker is (+OBSERVER)
	(±ROUND)	round axes are marked +ROUND

These features are present in the lexical entries of both nouns and adjectives. In nouns, they are grouped into feature trees, whose structure is governed by special lexical rules. Each path down the tree from the root node corresponds to an axis of an object. For example, Figure 1.1a shows the lexical entry for *Schrank* ‘cupboard/closet’. The vertical axis corresponds to the path terminating at 1, the lateral (side-side) axis terminates at 2, and the frontal axis terminates at 3. Bierwisch’s theory also allows multiple axes to be assigned exactly the same features when they cannot be distinguished due to symmetry of the object, forming what he calls an ‘integrated’ axis. For example, in the lexical representation of *Stange* ‘rod’ shown in Figure 1.1b, the two dimensions of the rod’s thickness are grouped together under (−MAIN).



(a) *Schrank* ‘cupboard/closet’ **(b)** *Stange* ‘rod’

Figure 1.1: Part of the lexical entries for two nouns, from Bierwisch (1967).

The representation of dimensional adjectives take the form of rules which instruct the grammar to place the feature (±POL), indicating a larger- or smaller-than-usual extent, at a particular location (marked by ∗) in an axis tree. For this to happen, the axis tree given in the rule must unify with the axis tree of the noun. For example, the adjective *hoch* ‘high/tall’ is represented by the rule:

(32) (+POL) [(+MAIN)[∗[(−INHERENT)[(+VERT)]]]]

which instructs the grammar to insert (+POL) into the tree, in such a way that it is subordinate to (+MAIN), but subordinate to (−INHERENT) and (+VERT). Impossible combinations, such as ?*hocher Apfel* ‘high apple’ are the result of unification failure. Bierwisch’s theory covered only adjective + noun combination: it did not explain how a spatial adjective could be applied to an individual as in *John is taller than Mary.*

1.3.2 Lang’s object schemas

Lang (1989, 2001) took Bierwisch’s theory as a starting point. His aim was to simplify Bierwisch’s formalism whilst increasing its scope. Like Bierwisch, he was concerned with explaining which spatial adjective + noun combinations were possible and which impossible. He also took from Bierwisch the idea that nouns can be sorted into dimensional classes based on the presence or absence of features encoding properties of the object’s axes. Lang’s features are unary rather than multivalued, and include the following:

(33)	MAX	the most extended axis of an object
	SUB	either the minimal axis of a surface or the small cross-section of a cylinder
	DIST	the ‘inside’ extent of a hollow object, such as the internal diameter of a hollow tube or box
	VERT	the vertical axis of an object
	OBS	the axis of an object which is aligned with an observer’s line of sight
	ACROSS	a horizontal axis which is either secondary in comparison to MAX or orthogonal to OBS

Features are composed into *object schemas*, Lang’s equivalent of Bierwisch’s axis trees. Object schemas are matrices whose columns, labelled ⟨a b c⟩, represent the axes of an object ordered according to prominence. As in Bierwisch’s theory, axes can be integrated, meaning that their prominence cannot be distinguished, which is indicated by grouping axes in parentheses: thus, a pole is a ⟨a (b c)⟩ object, whereas a disk is a ⟨(a b) c⟩ object. The other rows of the matrix are used to assign features to axes according to a collection of well-formedness rules, with integrated axes being treated as a single axis for the purpose of feature assignment. For example, (34) shows the object schemas for *pole*, *desk* and *wine bottle*:

(34)

pole			*desk*			*wine bottle*	
⟨ a	(b c) ⟩		⟨ a	b	c ⟩	⟨ a	(b c) ⟩
MAX	MIN		MAX	OBS	VERT	MAX VERT	DIST

Like Bierwisch, Lang sees adjective + noun composition as involving a kind of unification. He associates each antonymous pair of German adjective with a particular feature, as follows:

(35)
- *lang* ‘long’ / *kurz* ‘short’: MAX
- *dick* ‘thick’ / *dünn* ‘thin’: MIN
- *weit* ‘wide/far’ / *eng* ‘narrow’: DIST
- *hoch* ‘high/tall’ / *niedrig* ‘low’: VERT
- *tief* ‘deep’ / *flach* ‘shallow’: OBS
- *breit* ‘broad/wide’ / *schmal* ‘narrow’: ACROSS

For an adjective to combine with a noun, either the relevant feature must be already present in the noun representation, or it must be possible to extend the noun representation so as to include the feature without violating the well-formedness rules. Lang’s analysis concerns the compatibility of adjectives and nouns, e.g. the fact that *long pole* is acceptable but ?*long apple* is not. It does not tell us what it *means* for a pole to be *long*: Lang leaves this as a question for the interpretive or model-theoretic component of semantics.

Lang's taxonomy of object schemas was taken up by Stolz (1996), who applies Lang's system to Yucatec Maya. She finds that the kinds of distinctions which are relevant to German spatial adjectives – dimensionality, integratedness, relative extension, vertical/horizontal orientation, alignment with the observer, and so on – are precisely what is needed to characterise the Yucatec spatial adjectives. Following Lang, Stolz groups nouns into dimensional types or 'combinatorial classes' depending on which dimensional terms they can occur with. For instance, Class 6 consists of all 2D objects with a maximal horizontal axis such as fields, streets and surfaces: all objects which can be described as *chowak* 'long' and *kòoch* 'wide'. One difference with Lang's work is that, for Stolz, membership in a particular combinatorial class is not lexical but a matter of how the object is conceptualized – for instance, a tortilla belongs to Class 2 (2D objects) when thought of as a flat two-dimensional object, but Class 15 (3D objects with an 2D integrated axis and a small thickness) when its thickness is taken into account. Rather than each noun being inherently associated with a particular dimensional class, each noun makes a range of different classes available depending on the geometry of the object it denotes.

1.3.3 Vandeloise: Spatial adjectives as complex categories

One obvious criticism of featural approaches is their use of tailor-made primitives, which seem to directly encode the concepts they are intended to explain. This aspect of featural approaches was criticised by the cognitive linguist Vandeloise (1988, 1993), who saw his work as a direct response to Bierwisch. Vandeloise rejected a straightfoward mapping between spatial adjectives and semantic features, instead opting for a detailed lexical analysis of their different senses. In his study of length and width, he criticises Bierwisch for only taking into account geometry and neglecting factors such as the motion and function of an object. He points out that the length of an object does not always refer to its maximal dimension: if this was the case, then a sentence like *the airplane is wider than it is long* would be contradictory. Instead, he distinguishes four different senses of length, each of which has a corresponding notion of width:

(36)
- L_1/W_1: The length/width of a linear entity or path is its extent evaluated along its actual shape / a direction perpendicular to its actual shape.
- L_2/W_2: The length/width of a mobile entity is its extent evaluated along a direction parallel/perpendicular to its direction of movement.
- L_3/W_3: The length/width of a immobile, relatively symmetric entity is its extent along a direction parallel/perpendicular to the speaker's general orientation.
- L_4/W_4: The length/width of an entity is its greatest/smallest non-vertical extent.

L_1/W_1 covers pathlike objects like roads, rivers and corridors, whose length is evaluated along a curve rather than in a straight line. L_2/W_3 covers mobile objects like vehicles and other artifacts with a canonical direction of motion; it explains why the frontal axis of a vehicle can be referred to as a length even when it is less than the object's width. L_3/W_3 covers large objects like buildings, which are described differently depending on the viewpoint of the speaker. Finally, L_4/W_4 covers other objects which do not fall into the previous categories, and corresponds to the usual 'maximal' definition of length described by Bierwisch.

Clearly, it is no accident that the same word is used to cover concepts L_1 through L_4. To explain how these senses are connected, Vandeloise proposes what he calls 'pragmatic bridges' – canonical situations in which two or more separate senses coincide:

(37)
- L_1 and L_2: When an object moves forward along a path, its direction of motion (L_2) is parallel to the tangent of the path (L_1) at every point.
- L_1 and L_3: When an observer travels forward along a path, their line of sight (L_3) is parallel to the tangent of the path (L_1) at every point.
- L_2 and L_4: The direction of motion of a mobile entity (L_2) is typically also its maximal dimension (L_4), because this minimizes air resistance.
- L_2 and L_3: When one walks towards an entity, the direction in which it exhibits relative motion (L_2) is the direction parallel to one's line of sight (L_3).

Vandeloise combines the senses and pragmatic bridges into a kind of lexical network, using bidirectional arrows to represent bridging rules:

(38)
$$\begin{tikzcd} L_1 \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & L_2 \arrow[d, leftrightarrow] \arrow[dl, leftrightarrow] \\ L_3 & L_4 \end{tikzcd}$$

There is also a width network which is isomorphic to the length network, since each concept of length has a corresponding notion of width. What was important about Vandeloise's approach was his general methodology of identifying distinct senses and sense connections, rather than trying to formulate abstract definitions to cover as many senses as possible. In a later paper entitled *The role of resistance in the meanings of thickness* (1993), he applies a similar methodology to the analysis of *thick* and *thin*.

The networks proposed by Vandeloise describe what I have called implicational polysemy: they express idiosyncratic relations between microsenses. Notice, however, that (38) does not have the same form as the implicational networks discussed in the previous section, due to its bidirectional arrows. To convert a Vandeloise network into an ordinary implicational network, the 'pragmatic bridges' must be turned into *meets*. Recall that a pragmatic bridge is a situation in which one or more senses is satisfied simultaneously, which is precisely the definition of a meet. The length and width networks would be rewritten as follows:

(39)
$$\begin{tikzcd}
L_1 & L_{1,2} \arrow[l] \arrow[r] & L_2 \\
L_{1,3} \arrow[u] \arrow[d] & L_{2,3} \arrow[ur] \arrow[dl] & L_{2,4} \arrow[u] \arrow[d] \\
L_3 & & L_4
\end{tikzcd}$$

Note how each bidirectional arrow has been converted into a sense with arrows into the two original senses. This is now a well-formed implicational network in the sense of Section 1.2.2.

1.3.4 Spatial adjectives and prototype theory

Another study to reject the featural approach in favour of complex categories was Dirven & Taylor (1986). This was one of the first applications of prototype theory, in the sense of Eleanor Rosch, to the semantics of adjectives. In an experimental paradigm inspired by Rosch's work, participants were asked to rate the acceptability of different combinations of '*tall* + noun' on a scale from 1 (most acceptable) to 7 (least acceptable). They found that combinations with high average scores included *tall person* (1.05), *tall building* (1.83) and *tall pillar* (2.00); medium average scores included *tall shadow* (4.25), *tall door* (4.52) and *tall infant* (4.99); and low average scores included *tall table* (6.04), *tall cloud* (6.40) and *tall hair* (6.94). To explain these results, the authors propose that the acceptability of *tall* + noun is governed by a conceptual prototype:

(40) *Tall* prototypically applies to objects (Dirven & Taylor 1986):
  a. with a canonical vertical orientation (*tall person* is better than *tall infant*)
  b. whose vertical dimension is maximal (*tall mountain* is better than *tall hill*)
  c. whose vertical dimension is sufficiently large (*tall fence* is better than *tall ribbon*)
  d. which are solid rather than hollow (*tall tree* is better than *tall wardrobe*)
  e. profiled against a background (*tall bookcase* is better than *tall door*)
  f. which have acquired their height through a process of growth or construction (*tall building* is better than *tall window*)

A situation in which the object meets all of these criteria is judged as highly typical (e.g. *tall person*), whereas a situation in which only some criteria are met is judged as less typical (e.g. *tall lorry*). Dirven & Taylor's model can form the basis for an implicational network in which the central sense is given by (40) and other more peripheral senses correspond to smaller sets of conditions which are nevertheless sufficient for an object to count as *tall*.

D&T's work was built on by Goy (2002) in her study of the Italian adjectives *alto* 'high/tall' and *basso* 'low'. Goy begins from the assumption that a complete

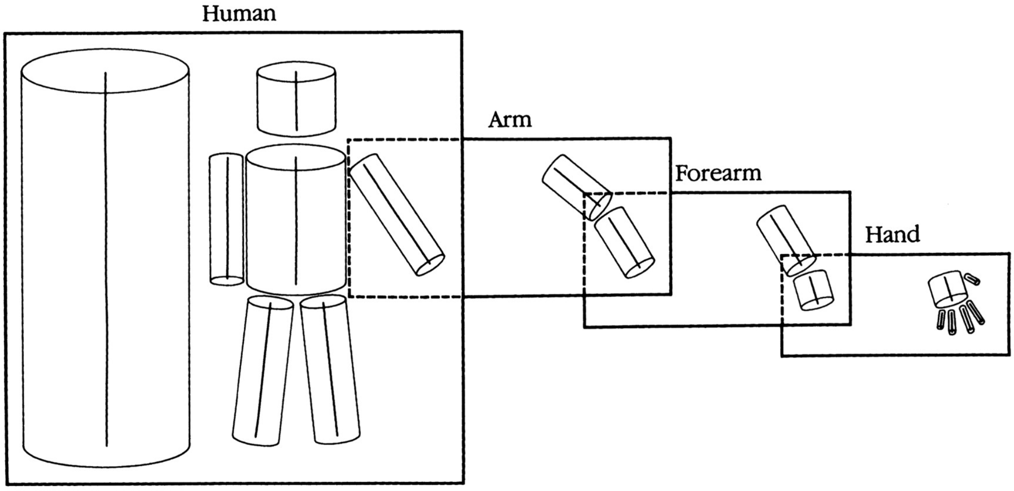


Figure 1.2: Marr's 3D Model system is based on the *generalized cylinder*, which is any volume generated by sweeping a cross section of constant shape but varying size along an axis. Any cylinder can be broken down by an 'elaboration rule' into a collection of smaller cylinders. This rule can be applied iteratively, resulting in a hierarchy of coordinate systems in which the top-level node contains the most abstract description of the shape and lower-level nodes contain more fine-grained detail.

account of lexical semantics requires a link between language and perception. Spatial adjectives are seen as accessing a level of spatial representation which is derived from perceptual experience. The theory of shape representation which Goy assumes is Marr's (1982) 3D Model, a viewpoint-independent system of shape representation designed for visual object recognition (see Figure 1.2). The lexicon contains a canonical shape description for each concrete noun in the speaker's lexicon. For example, the 3D model for *pesce* 'fish', specifies the shape of a typical fish, including typical values for body length, fin shape, tail shape, and so on.

Following D&T's methodology, Goy conducted a short experiment in which participants were presented with '*alto/basso* + noun' combinations and asked to rate their acceptability from 1 ('completely acceptable') to 7 ('not understandable'). Nouns scored highly if they could be conceptualized as vertical cylinders (e.g. *torre* 'tower'), or if they had a vertical axis which was at least as large as a human being (e.g. *muro* 'wall'). Goy's analysis is that *alto* and *basso* modify an object's *relevant vertical oriented axis* (RVOA). This can be either a top-level vertical axis, such as the generating axis of the human figure shown in Figure 1.2, or a particularly salient vertical axis lower down in the hierarchy, such as the vertical neck axis of a giraffe. According to Goy, The meaning of the adjectives *alto* and *basso* is a procedure which finds and modifies an object's RVOA by either increasing it (*alto*) or decreasing it (*basso*). The unique aspect of Goy's account was its reliance on a non-linguistic system of object representation which interfaces with spatial language.

1.3.5 Vogel: Bringing everything together

The cognitive approach to spatial adjectives culminated in the work of Anna Vogel (2004), which combines the complex categories of Vandeloise, the empirical approach of Dirven & Taylor, and Goy's concern with perceptual experience. One aspect of Vogel's analysis which is missing from previous theories is her use of the cognitive linguistics notion of *image schema*. Recall that an image schema refers to a basic pattern which is abstracted from perceptual/bodily experience, such as CONTAINER, CONTACT, BLOCKAGE, and so on (Johnson 1987, Lakoff 1987). Although image schemas have parts, they are supposed to be primitive in the sense that the whole is conceptually prior to the parts, being grasped as a kind of gestalt. Vogel uses image schemas to characterise the selectional restrictions associated with particular adjectives. For example, she links Swedish *djup* 'deep' and *grund* 'shallow' to the CONTAINER image schema which is also involved in English *in*, *into* and *inside*. For an object to be described as *djup* or *grund*, it must be possible to conceptualize it as a CONTAINER with an open top; hence combinations like *djup skål* 'deep bowl' and *djup brunn* 'deep well' are acceptable, whereas combinations like ?*djup cykel* 'deep bicycle' and ?*djup jordglob* 'deep globe' are not.

An adjective may be associated with more than one image schema, in which case the relevant schemas are grouped together into a network. The more central a node in this network, the more prototypical it is judged to be. Networks with multiple prototypes are called 'polysemous' networks; those with only a single prototype are called 'monosemous' networks. Vogel's methodological strategy is to identify the most prototypical sense(s) of an adjective using a combination of corpus data and questionnaire responses. She then uses the same data to enumerate a collection of peripheral senses, which she connects in various ways to the prototype. The links between nodes represent transformations which can change or drop elements of the prototype. For example, according to Vogel, the adjective *tjock* 'thick' prototypically refers to the cross-section of a cylinder, which is (a) minimal in extent, (b) graspable by the hands, and (e) resistent to deformation. Other senses of *tjock* are derived from this prototype by weakening or altering these conditions, giving the network shown in Figure 1.3.

Vogel's networks clearly represent what Section 1.2.2 refers to as implicational polysemy: they describe idiosyncratic relations between microsenses of the same lexical item. However, it can be difficult to interpret her lexical networks in implicational terms. For example, it is not clear how the 'hard flat object' sense of *tjock*, as in *tjocka väggar* 'thick walls', can be derived from the cylindrical sense, as in *tjock grubbe* 'thick man'. Nor can these two senses share a common meet, because being a flat object is incompatible with being a cylinder. Instead, they appear to share a common *join*; this is supported by the observation that, in English, one can say something like *the stick is thicker than the plate*, which compares a cylindrical thickness and a surface thickness. Other links proposed by Vogel seem to represent meets, and others straightforward implications. My view is that all of the network proposed by Vogel could be converted into standard implicational networks of the sort explained in Section 1.2.2, potentially with some additional senses.

Vogel's ideas were taken up by Shimotori (2013), in a comparative study of

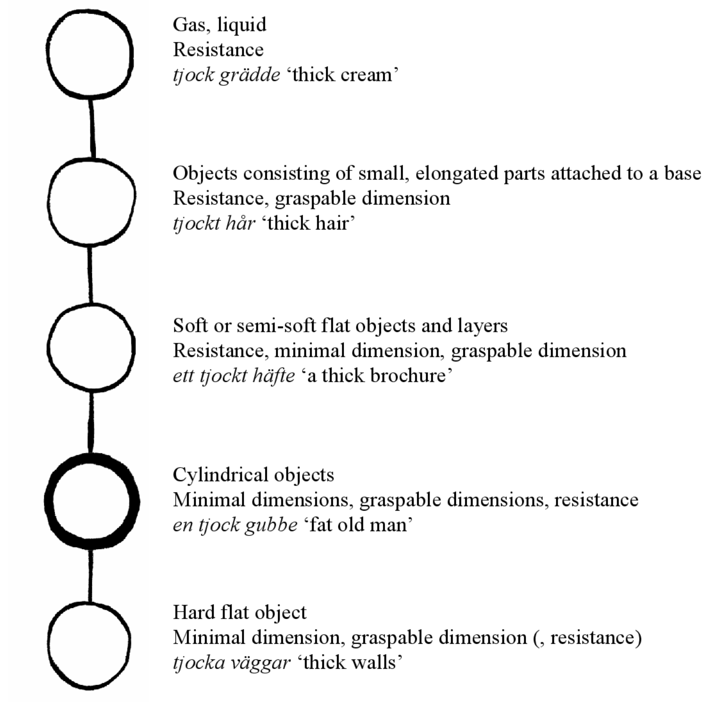


Figure 1.3: Vogel's (2004) network description of *tjock* 'thick', with the prototype shown in bold.

spatial adjectives in Swedish and Japanese. Shimotori found that the two languages show a great deal of agreement in the most central, spatial senses of these adjectives. For example, like Swedish *vid* 'wide', Japanese *horoi* 'wide' is mostly associated with empty two-dimensional spaces or areas, such as *umi* 'ocean', *sora* 'sky', *heya* 'room' and *sôgen* 'grassland'. A major difference is that Japanese distinguishes between *futoi* 'thick' / *hosoi* 'thin' for the thickness of a cylindrical object, and *atsui* 'thick' / *usui* 'thin' for the thickness of a flat object. When it came to the non-spatial or metaphorical usages of spatial adjectives, there was much less agreement. For example, thickness has a positive connotation in Japanese which is lacking in Swedish: someone who is kind-hearted is described as having *atsui ninjou* 'thick empathy'.

1.3.6 Summary of lexical approaches

Looking at the various lexical approaches to spatial adjectives, there appears to be quite widespread agreement on the kinds of conceptual primitives which are relevant to their semantics. These include concepts such the relative extension of axes, whether two axes are distinguished or integrated, the orientation of axes, the notion of a canonical observer, the concept of a container, the concept of path/passage, and so on. These primitives are important not only to the English spatial adjectives, but also to their analogs in other languages, suggesting that they originate in some non-linguistic faculty of spatial cognition. However, although the primitive notions appear to be similar across languages, the way they are packaged into adjectives can differ. For example, the English distinction between *high* and *tall* is not present in most languages, e.g. German *hoch* ‘high/tall’, Italian *alto* ‘high/tall’, Yucatec Maya *ka’nal* ‘high/tall’. Similarly, the distinction between cylindrical thickness and surface thickness, present in Japanese (*futoi* vs. *atsui*) Yucatec Maya (*polok* vs. *píim*), is neutralized in most Germanic languages.

A single spatial adjective does not correspond to a single primitive, but rather to a network of senses, where each sense is a collection of primitives. For instance, consider the English adjective *wide*. Relevant conditions include being secondary in extent (*wide ribbon*), being orthogonal to a canonical observer (*wide desk*), being horizontal, being an open area (*wide ocean*), being conceptualized as a passage (*wide corridor*), etc. Some of these conditions are sufficient by themselves, others are sufficient only in combination with others; some combinations are compatible, others incompatible; and so on. In short, they form a family resemblance category, also known in this thesis as an implicational network. Some adjectives, such as *tall*, form a radial category with only a single prototype; others, such as *wide*, contain multiple prototypes. Likewise, some adjective networks contain a single final sense, so that all of their senses are mutually compatible; others contain multiple final senses, giving rise to zeugmatic readings.

Lexical approaches are primarily concerned with explaining the *attributive* behaviour of spatial adjectives: for example, the fact that *tall person* is acceptable but ?*tall marble* is not. This can easily give the impression that a spatial adjective is a kind of procedure for modifying a noun (indeed, some authors such as Goy state this explicitly). However, there are also various *predicative* uses of spatial adjectives – such as the positive (*X is tall*), the comparative (*X is taller than Mary*), the superlative (*X is the tallest person in the room*), and measure phrase combination (*X is 150cm tall*) – where the adjective does not modify a noun but instead describes a property of individuals. The focus on the attributive in the lexical semantics literature creates a disconnect with formal semantics, where the predicative use is generally considered to be more basic, the attributive being derived through intersection of the noun denotation with the positive form of the adjective. The next section tells the other side of the story, introducing the formal semantics of degree constructions.

1.4 Introduction to degree semantics

Degree semantics is a branch of truth-conditional model-theoretic semantics which is concerned with gradable adjectives and the constructions they support, e.g. the comparative and superlative, as well as other gradable phenomena. The founding assumption in model-theoretic semantics is that the semantics of a natural language should be given in the same way as the semantics of a formal language (e.g. first-order logic), by specifying a relation between expressions of the language and elements of a *model*, a mathematical structure containing entities that the language 'talks about'. The relation, called *interpretation*, is such that composition of words in the language is mirrored by composition of elements in the model, a correspondence known as *compositionality*. The basic goal of this kind of semantics is to use the notion of interpretation with respect to a model to give a general characterisation of the conditions under which sentences are true or false. It is conventional to interpose a formal language F between the natural language and the model, the advantage being that the interpretation of formulas of F is already well-understood, so the difficult problem of how to directly interpret expressions in the natural language is replaced by the slightly less difficult problem of how to translate natural language expressions into formulas of F.

In this section, I assume the usual choice for F, which is the simply-typed lambda calculus. One introduces two basic types – individuals (e) and truth values (t) – together with an infinite ladder of function types relating them – $(e \to t)$, $(e \to (e \to t))$, $((e \to t) \to t)$, and so on – terms of which are lambda expressions. Proper nouns denote individuals and are typed e; common nouns, intersective adjectives and intransitive verbs denote properties and are typed $e \to t$; sentences denote truth values and are typed t. Compositionality is modelled by applying lambda expressions to arguments. Note that, although it is adopted here for expository purposes, the simply-typed lambda calculus is not the formalism which I shall eventually advocate in this thesis. For a more formal introduction to the simply-typed lambda calculus, see Section 2.2. For some of the limitations of Montague semantics, see Section 2.3.

1.4.1 Scale structure

Degree semantics begins with the intuition that some properties are not all-or-nothing, but rather come in degrees. For example, one cannot divide all individuals into those who are *happy* and those who are *not happy* – rather degrees of happiness form a scale, with some individuals being happier than others. This behaviour is not well-expressed by assigning *happy* the type $e \to t$, since then it behaves as an all-or-nothing property. In most versions of degree semantics, one therefore introduces a new ontological type, d, to describe the extent to which an individual possesses a property. One then distinguishes between *gradable* adjectives, which involve degrees, and *non-gradable* adjectives, which are all-or-nothing properties. An easy way to tell whether an adjective is gradable or non-gradable is to ask whether it supports comparison. Hence, *happy* is gradable, since one person can be happier than another, whereas *married* is non-gradable, since one person cannot be 'more married' than another.

Degrees are collected into ordered structures called *scales*, which are associated with particular properties – the length scale, the weight scale, the intelligence scale, and so on. A scale is generally thought of as a triple $\langle D, \leq, \delta \rangle$, where D is a set of degrees, $\leq$ is a total order relation on D, and δ is a dimension parameter telling you the dimensions of the scale – distance, time, weight, temperature, etc. (this formulation of scales comes from Kennedy & McNally 2005). For two degrees to be compared, they must lie on the same scale. Consider the following sentences:

(41) a. John is taller than Susan.
b. John is taller than the table is wide.
c. John is shorter than the table is narrow.
d. ? John is taller than Susan is old.

Sentence (a) is a typical example of comparison which compares the height of two people; (b) compares the height of a person to the width of a table, which is acceptable since *tall* and *wide* denote degrees on the same scale (the distance scale); (c) is also acceptable and resembles (b) except that the two adjectives are negative rather than positive. However, sentence (d) is unacceptable because *tall* and *old* involve degrees on different scales (distance and age respectively).

Besides having different dimensions, scales can also vary with respect to the order relation. Antonymous pairs, such as *long/short*, *big/small*, *old/new*, etc., have the same dimensions but opposite order relations. For instance, the scale associated with the adjective *big* is the same as that of *small*, but with the opposite order. There are two major advantages of this analysis. Firstly, given that scales like size are total orders, it allows us to derive the inference that *x is bigger than y* iff *y is smaller than x*. Secondly, the fact that *big* and *small* project their arguments onto distinct scales explains why they cannot be compared in a sentence like ?*the house is bigger than the car is small*, which on this account is unacceptable for the same reason as cross-dimensional comparisons like sentence (41d).

Another way in which scales can vary is the structure of the set of degrees itself. The topological possibilities for a scale are the same as those of an interval – in either direction (upwards or downwards), the scale may be either unbounded, bounded and closed, or bounded and open. For example, the scale of temperature is unbounded in both directions; the scale of distance is upwards unbounded but downwards bounded and open; the scale of purity, as in *the water is pure*, is upwards bounded and open but downwards bounded and closed; and the scale of fullness, as in *the glass is full*, is bounded and closed in both directions. The topology of scales is relevant to the selectional restrictions associated with proportional modifiers like *completely*, *mostly*, *half*, and so on (Kennedy & McNally 2005). For example, the modifier *completely* requires a scale which is upwards bounded and closed – hence *completely full* is acceptable, but ?*completely tall* is unacceptable. Note that if a scale is upwards bounded and closed, then its opposite is downwards bounded and closed (and vice versa). Hence *completely pure* is acceptable but ?*completely impure* is unacceptable.

1.4.2 Typing gradable adjectives

Although different degree-based approaches broadly speaking agree about the nature of scale structure, there is some disagreement on the semantic type of gradable adjectives. Recall that an ordinary intersective adjective has type $e \to t$, a function from individuals to truth values. Some authors (e.g. Cresswell 1976, Bierwisch 1989, Heim 2000) treat gradable adjectives as elements of type $d \to e \to t$, that is, relations between individuals and degrees. On this view, a gradable adjective like *tall* is interpreted as follows[1]:

(42) $[\![\text{tall}]\!] \coloneqq \lambda d \,.\, \lambda x \,.\, \mathbf{tall}(x) = d$

where **tall** is some function of type $e \to d$, which takes an individual and returns its degree of tallness. On this analysis, $[\![\text{tall}]\!](d)(x)$ is true iff x's degree of tallness is exactly d. An alternative analysis is to weaken the strict equality to 'greater than or equal to':

(43) $[\![\text{tall}]\!] \coloneqq \lambda d \,.\, \lambda x \,.\, \mathbf{tall}(x) \geq d$

In other words, $[\![\text{tall}]\!](d)(x)$ is true iff x's degree of tallness is *at least* d. The motivation behind this approach – which is advocated for example by Klein (1980) – is the intuition that a sentence like *the fence is 2m tall* should not rule out the possibility that the fence is actually taller than 2m. The exact equality reading is then explained as a scalar implicature arising from the fact that the speaker might have said *the fence is taller than 2m*.

One issue with this relational account of gradable adjectives, which was pointed out by Kennedy (1999), is the lack of scope ambiguities in certain contexts. To illustrate, consider the comparative morpheme *more/-er*, which on the relational account is interpreted as follows (assuming the exact equality analysis given in 42):

(44) $[\![\text{more/-er}]\!] \coloneqq \lambda g \,.\, \lambda d \,.\, \lambda x \,.\, \exists d'[d' > d \ \wedge\ g(d', x)]$

In other words, $[\![\text{more/-er}]\!](g)(d)(x)$ is true iff there exists some degree $d' > d$ which measures the extent to which x is g. The following example shows how this works for *John is taller than 150cm*:

(45) $[\![\text{John is taller than 150cm}]\!]$

$$
\begin{aligned}
&= [\![\text{more/-er}]\!]([\![\text{tall}]\!])([\![\text{150cm}]\!])([\![\text{John}]\!]) \\
&= \lambda g.\lambda d.\lambda x.\exists d'[d' > d \ \wedge\ g(d', x)]\ (\lambda d \,.\, \lambda x \,.\, \mathbf{tall}(x) = d)\ (\mathbf{150cm})\ (\mathbf{John}) \\
&= \lambda d.\lambda x.\exists d'[d' > d \ \wedge\ \mathbf{tall}(x) = d']\ (\mathbf{150cm})\ (\mathbf{John}) \\
&= \lambda x.\exists d'[d' > \mathbf{150cm} \ \wedge\ \mathbf{tall}(x) = d']\ (\mathbf{John}) \\
&= \exists d'[d' > \mathbf{150cm} \ \wedge\ \mathbf{tall}(\mathbf{John}) = d']
\end{aligned}
$$

[1] To avoid the need for type specifications on arguments, I use the variables $\{x, y, z\}$ for type e, $\{d, d', d''\}$ for type d, $\{g, g', g''\}$ for type $e \to d \to t$, $\{f, f', f''\}$ for type $e \to d$, and $\{p, p', p''\}$ for type $e \to t$.

As shown, on this analysis, *John is taller than 150cm* is true iff there is some degree d' greater than 150cm which measures the extent to which John is tall. Now consider the well-known scope ambiguity associated with sentences like the following:

(46) Everybody loves a song.

a. $\forall x : \mathbf{person}(x)[\exists y : \mathbf{song}(y)[\mathbf{love}(x, y)]]$

b. $\exists x : \mathbf{song}(x)[\forall y : \mathbf{person}(y)[\mathbf{love}(y, x)]]$

where (a) corresponds to the reading 'for each person, there is some song that they love', and (b) corresponds to 'there is some song which every person loves'. If the existential analysis of comparatives is correct, then we might expect comparatives to show a similar ambiguity. For example, the sentence *everybody is taller than 150cm* should have the following two interpretations:

(47) Everybody is taller than 150cm.

a. $\forall x : \mathbf{person}(x)[\exists d'[d' > \mathbf{150cm} \ \wedge \ \mathbf{tall}(x) = d']]$

b. $\exists d'[\forall x : \mathbf{person}(x)[d' > \mathbf{150cm} \ \wedge \ \mathbf{tall}(x) = d']]$

where (a) can be paraphrased as 'for each person, their degree of height is greater than 150cm', and (b) can be paraphrased as 'there is some degree d greater than 150cm, such that every person is d-tall'. However, interpretation (b), under which everybody in the context has the same height, does not occur to people, suggesting that the existential analysis of *more/-er* is mistaken.

To solve this problem, Kennedy (1999) proposed an alternative account of gradable adjectives, following an earlier suggestion by Bartsch & Vennemann (1974). The idea is to treat gradable adjectives purely as elements of type $e \to d$, that is functions from individuals to degrees. In fact, functions of this type were already presupposed in the relational account, where there is a function like $\mathbf{tall} : e \to d$ for each adjective. We now simply equate the meaning of the adjective with this function:

(48) $[\![\text{tall}]\!] \coloneqq \mathbf{tall}$

On this view, relations between individuals and degrees are not an inherent part of the meaning of a gradable adjective, but are introduced separately by degree morphology. The comparative morpheme no longer involves an existential quantifier, but is represented as follows:

(49) $[\![\text{more/-er}]\!] \coloneqq \lambda f \,.\, \lambda d \,.\, \lambda x \,.\, f(x) > d$

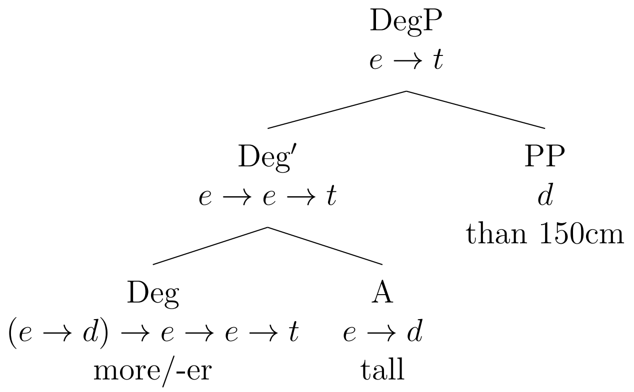
that is, given a gradable adjective f (now a pure measure function of type $e \to d$), a degree d and an individual x, $[\![\text{more/-er}]\!](f)(d)(x)$ is true iff f applied to x is greater than d. Given this definition, the sentence *everybody is taller than 150cm* has only one reading, namely:

(50) $\forall x : \mathbf{person}(x)[\mathbf{tall}(x) > \mathbf{150cm}]$

which is what we would expect.

Kennedy's analysis of gradable adjectives as measure functions is associated with a standard syntactic analysis, according to which a gradable adjective must appear wrapped inside a Degree Phrase (DegP), just as a verb appears wrapped inside an Inflectional Phrase (e.g. Abney 1987, Corver 1991, Grimshaw 1991, Kennedy 1999, 2007). The head of a DegP is a degree morpheme (e.g. *more/-er*, *most/-est*, *very*, *too*, *enough*, *half*), which takes the adjective as an internal argument. For instance, a phrase like *taller than 150cm* would have the following structure:

(51)



This transparently supports the semantic analysis given in (45), where ⟦more/-er⟧ takes ⟦tall⟧ as an argument.

1.4.3 The positive/unmarked form

The term *positive* is used to refer to the unmarked form of a gradable adjective, as in *John is tall*, *the Nile is long*, *semantics is interesting*, and so on[1]. The positive is *vague*, meaning that it does not separate individuals into two sharply bounded sets. For instance, people with a height of 200cm are clearly *tall*, whereas people with a height of 100cm are clearly *short*; but there are people in-between who are difficult to categorize as either *tall* or *short*. Compare this to other degree constructions such as the comparative or measure phrase combination, where there is a definite fact of the matter. For example, the truth of *John is taller than Mary* and *John is 150cm tall* can be decided by someone who knows all the relevant facts, namely the heights of John and Mary. In contrast, one can know John's height and still be unsure whether *John is tall* is true or false. For this reason, despite its apparent morphological simplicity, the positive has proven the most difficult degree construction to describe from a semantic point of view.

In addition to vagueness, the positive also exhibits *context-dependence*, meaning that its interpretation depends on a comparison class. For example, someone with a height of 180cm might be considered tall in the context of people in general, but short in the context of basketball players. The comparison class can appear explicitly in the sentence, as in *John is tall/short <u>for a basketball player</u>*. If no

[1] The use of the term positive to refer to a kind of degree construction should not be confused with positive vs. negative polarity.

explicit comparison class is provided, then the comparison class is provided either by the noun which the adjective modifies (e.g. *John is a tall* <u>*man*</u>) or by the general discourse context (e.g. "*John is tall*", said in a room full of basketball players). Context-dependence occurs not only in the positive form, but also in the superlative – for instance, the truth of the sentence *John is the tallest* depends on the set of people to whom John is being compared. In contrast, a comparative sentence like *John is taller than Sally* is not context-dependent since it does not presuppose a comparison class.

The usual analysis of the positive in degree-based approach is in terms of a contextually determined standard value (Bartsch & Vennemann 1974, Cresswell 1976, von Stechow 1984, Klein 1980, Kennedy 1999, Kennedy & McNally 2005). The basic idea is that a sentence like *John is tall* can be paraphrased as "the degree to which John is tall exceeds some contextually-determined standard degree s". Context-dependence is explained by the dependence of the standard on a comparison class. Theories differ as to exactly how the comparison class enters into the semantics, with some treating it as a logical argument to the positive morpheme and others as a free variable whose value is filled in by pragmatic considerations. The vagueness of the positive is explained through the 'fuzziness' or lack of certainty associated with the standard. Again, theories differ as to how exactly this fuzziness is understood.

To give a typical example, Bartsch & Vennemann (1974) treat the positive as a silent morpheme with the following denotation:

(52) $[\![\textsc{pos}]\!] := \lambda f\,.\,\lambda p\,.\,\lambda x\,.\,f(x) > \mathbf{norm}(p)(f)$

where f is a measure function, p is a property representing the comparison class, and $\mathbf{norm} : (e \to t) \to (e \to d) \to d$ is a function which returns the average degree to which members of p are f. Given this definition, the interpretation of *John is tall for a basketball player* would be:

(53) $[\![\text{John is } \textsc{pos} \text{ tall for a basketball player}]\!]$

$= [\![\textsc{pos}]\!]([\![\text{tall}]\!])([\![\text{basketball player}]\!])([\![\text{John}]\!])$

$= \lambda f\,.\,\lambda p\,.\,\lambda x\,.\,f(x) > \mathbf{norm}(p)(f)\ (\mathbf{tall})\ (\mathbf{basketball\text{-}player})\ (\mathbf{John})$

$= \lambda p\,.\,\lambda x\,.\,\mathbf{tall}(x) > \mathbf{norm}(p)(\mathbf{tall})\ (\mathbf{basketball\text{-}player})\ (\mathbf{John})$

$= \lambda x\,.\,\mathbf{tall}(x) > \mathbf{norm}(\mathbf{basketball\text{-}player})(\mathbf{tall})\ (\mathbf{John})$

$= \mathbf{tall}(\mathbf{John}) > \mathbf{norm}(\mathbf{basketball\text{-}player})(\mathbf{tall})$

That is, the sentence is true iff John's degree of height is greater than that of the average height of a basketball player.

1.4.4 The alternative: gradable adjectives as vague predicates

In a standard degree-based approach, the initial formalization of gradable adjectives is crisp, and vagueness is added 'on top' in the form of supervaluation or some other mechanism for constructing fuzzy interpretations. There is an alternative approach to gradable adjectives, sometimes called the 'delineation' or 'vague predicate' approach, in which vagueness is built in from the beginning (Kamp 1975, Klein 1980, 1982). On this account, the difference between non-gradable and gradable adjectives is that non-gradable adjectives denote functions from individuals to bivalent truth values, whereas gradable adjectives denote functions from individuals to *trivalent* truth values, which are either true, false or undefined. For example, the adjective *tall* would partition its domain into three sets: a positive extension containing individuals which are definitely tall, a negative extension containing individuals which are definitely not tall, and an extension gap containing individuals which fall in-between or whose tallness is undefined. To explain context-dependence, the three-way partition is treated as depending on a context parameter c.

The price of the vague predicate approach is that the idea of comparison, which is built in to the degree-based approach, must be constructed in a more roundabout fashion. Given a gradable predicate like *tall*, each context gives us a three-way partition into positive extension, negative extension and extension gap. For coherence, we assume that, given a context c where x is in the positive extension and y is in the negative extension, there can be no context c' where the order is reversed so that y is in the positive extension and x is in the negative extension. Given this, the family of partitions indexed by c will generate a partial order which can be used as a basis for comparison. A sentence like *John is taller than Mary* will be true iff there is some context c where John is in the positive extension of *tall* and Mary is in the negative extension. Similarly, *John is the tallest person in the room* is true iff there is some context c where John is in the positive extension of *tall* and all the other people in the room are in the negative extension.

The major issue with the vague predicate account, which was pointed out by Kennedy (1999), is the difficulty of explaining cross-scalar and cross-polar incommensurability. Recall that, in order to be compared, two adjectives must have both the same dimensions and the same polarity. However, the vague predicate analysis does not seem to require this. For example, the analysis of the comparative sketched in the previous paragraph allows us to assign interpretations to sentences like the following:

(54) a. ?*John is taller than Susan is old*: there is some context c such that John is in the positive extension of *tall* at c and Susan is in the negative extension of *old* at c

b. ?*John is taller than Susan is short*: there is some context c such that John is in the positive extension of *tall* at c and Susan is in the negative extension of *short* at c

Both sentences can be satisfied given the right context, which is undesirable since they are perceived to be anomalous. This problem arises because the extensions of *tall*, *short* and *old* all contain objects of the same sort, namely individuals. Attempting to solve this problem requires constructing sortal distinctions between different kinds of scales, leading to a version of degree semantics in which degrees are derivative rather than primitive semantic elements. See Lassiter (2017) for an analysis along these lines, which he calls 'degree semantics without degrees'. Such an approach is virtually indistinguishable from an analysis in which degrees are introduced from the start as primitive semantic elements.

1.5 On the compatibility of formal and cognitive semantics

The aim of this thesis is to develop a description of spatial adjectives – and more generally an approach to semantics – which reconciles the concerns of both formal and lexical/cognitive approaches. This means unifying the description of polysemy in authors like Vandeloise (1988), Dirven & Taylor (1986) and Vogel (2004), with the compositional description of degree constructions developed by authors such as Bartsch & Vennemann (1974), Klein (1980) and Kennedy (1999). This project of trying to unite formal and cognitive semantics will seem wrongheaded to those – from both disciplines – who see them as having nothing to do with each other. For example, the philosopher of language and proponent of formal semantics, David Lewis, wrote:

> I distinguish two topics: first, the description of possible languages or grammars as abstract semantic systems whereby symbols are associated with aspects of the world; and second, the description of the psychological and sociological facts whereby a particular one of these abstract semantic systems is the one used by a person or population. Only confusion comes of mixing these two topics. (Lewis 1972, p. 19)

A similar idea is expressed by Vyvyan Evans, a cognitive linguist:

> The final difference that we mention here relates to the model of truth-conditional semantics that is adopted by most formal models of linguistic meaning. This approach assumes an objectivist position, which means that it assumes an objective external reality against which descriptions in language can be judged true or false. In this way, it builds a model of semantic meaning that can be made explicit by means of a logical metalanguage . . . This view stands in direct opposition to the experientialist view adopted within cognitive semantics, which describes meaning in terms of human construal of reality. (Evans 2006, p. 171-172)

Both authors present essentially the same distinction: formal semantics is about relating language to "aspects of the world" or "an objective external reality"; whereas cognitive semantics is concerned with "psychological and sociological facts" and therefore "describes meanings in terms of human construal of reality."

The idea that formal semantics and cognitive semantics have nothing to do with each other stems from a particular interpretation of formal semantics, which we can call the *realist* or *externalist* interpretation. On this view, formal semantics describes relations between linguistic expressions and elements of reality, where 'elements of reality' can include individuals, properties, relations, functions, possible worlds, propositions, contexts, and so forth – all the machinery needed to describe the truth-conditions of sentences. The model of the world on which formal semantics depends is held to be a mathematical description of the way the world is in itself, independent of how people conceptualize it. As Lewis puts it, semantic relations are "relations between symbols and the world of non-symbols" (ibid., p. 19). From this point of view, formal semantics and cognitive semantics have completely distinct aims: the former concerns relations between language and the world, whereas the latter concerns relations between language and the mind.

An alternative to the realist interpretation is the *conceptualist* or *internalist* interpretation. This is the view that “linguistic expressions refer to entities in the world *as conceptualised by the language user*” (Jackendoff 1998, p. 1). Conceptualism is sometimes considered to be incompatible with truth-conditional model-theoretic semantics. However, as Jackendoff points out, the conceptualist can accept the usual framework, whilst disagreeing with the realist interpretation of that framework:

> [Conceptualism] does not commit one to abandoning model-theoretic semantics. It *does* commit one to abandoning the usual construal of model-theoretic semantics, in which reference is to the model, and the model is taken to be the world or the infinite set of possible worlds . . . it is absolutely possible to adopt an alternative construal of model-theoretic semantics, in which the model is instead taken to be the world *as conceptualized by the language user*. [. . .] Similarly, [conceptualism] does not does not commit one to abandoning truth-conditional semantics – only to placing truth-conditions inside the mind of the language user rather than treating them as framework-free. (Jackendoff 1998, p. 2)

In other words, the conceptualist views the set-theoretic model as a mental construct, not a depiction of the real world. Examples of authors who have taken this point of view towards model-theoretic semantics include Bach (1986), Verkuyl (1989) and Zwarts & Verkuyl (1994). From the conceptualist point of view, formal semantics is not completely orthogonal to cognitive linguistics: the two fields have similar goals since they both involve relating linguistic expressions to human conceptualizations.

A consequence of ‘pushing the model inside the mind’ is that notions such as truth, entailment, reference, and so on, get relativized to particular speakers. As Jackenfoff puts it,

> in a mentalistic account of language, terms like “grammatical”, “true”, “analytic”, “refer” are taken to be at their foundation dependent on a language user (just as in special relativity, distances and times are taken to be dependent on an observer’s initial frame). They can be taken to be user-independent [. . .] only to the extent that it is useful and possible to conveniently ignore the “reference frame” – the presupposition of agreement among members of a (relatively) homogeneous speech community.

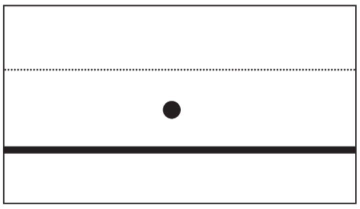
For the conceptualist, a statement like ‘sentence S uttered in context C is true’ must be considered as shorthand for ‘a typical speaker judges sentence S uttered in context C to be true’. This is analogous to the syntactician’s claim that ‘sentence S is grammatical’, or the phonologist’s claim ‘syllable S is well-formed’.

The perspective of this thesis is that of conceptual truth-conditional, model-theoretic semantics. The goal is to describe the truth conditions of sentences with respect to possible models of the world. However, the possible models should not be conceived as possible versions of reality, but rather as possible versions of reality *as conceptualized by an idealized language user.* The various elements of the model are not constituents of reality but human conceptualizations. To give an example taken from Janssen & Zimmermann (2021), we talk about a substance such as *water* as though every part of it can also be described as *water* – as though it has no minimal parts – although physically speaking this is not the case. The partless conceptualization of substances belongs to our ordinary, commonsense ontology: what the philosopher Wilfrid Sellers (1963) called the 'Manifest Image'. Likewise, all the semantic entities assumed in this thesis – individuals, types, predicates, degrees, measure functions, spatial vectors, paths, and so on – should be understood as belonging to Sellers' Manifest Image.

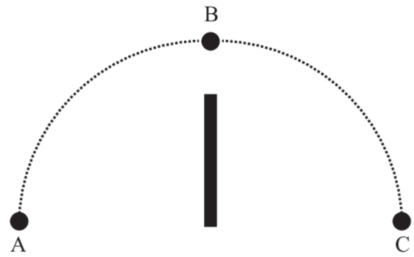
If the model is taken to be a conceptual construct, located inside the head of language users, then the various elements of the model must be identified with mental symbols of some kind or another. For a conceptualist, therefore, semantics describes a translation between symbols of one kind – natural language expressions – to symbols of another kind – concepts. One might worry whether anything is really achieved by this translation. David Lewis put this point as follows:

> Semantic markers are symbols: items in the vocabulary of an artificial language we may call *Semantic Markerese.* Semantic interpretation by means of them amounts merely to a translation algorithm from the object langugage to the auxiliary language Markerese. But we can know the Markerese translation of an English sentence without knowing the first thing about the meaning of the English sentence: namely, the conditions under which it would be true. Semantics with no treatment of truth conditions is not semantics. Translation into Markerese is at best a substitute for real semantics, relying either on our tacit competence (at some future date) as speakers of Markerese or on our ability to do real semantics at least for the one language, Markerese. (Lewis 1972, p. 1)

I agree with Lewis that translation into Markerese would be unilluminating if Markerese was anything like a natural language. However, let us suppose that what is meant by Markerese is a formal language for encoding and reasoning about arbitrary human concepts – a kind of formal theory of the Manifest image. The Markerese translation of a word should encode all the knowledge associated with it: for example, the Markerese translation for the word *cat*, should tell us the typical shape of a cat, its part structure and possible configurations, typical behaviours associated with a cat, their relationship with humans, and so on. Then it is not at all obvious that "we can know the Markerese translation of an English sentence without knowing the first thing about the meaning of the English sentence, namely the conditions under which it would be true". Rather, the Markerese translation of a sentence would encode the complete concept associated with this sentence, including a specification of the kind of situation in which it is true. The goal of conceptualist semantics is to develop something like Markerese in this sense.



(a) The proto-scene for *over*.



(b) The trajectory sense of *over*.

Figure 1.4: Two senses of *over*, from Tyler & Evans (2003).

Many cognitive linguists share with Lewis the intuition that an abstract system of symbols must be devoid of any real semantic content. However, rather than invoking relations between symbols and the real world, as Lewis does, they instead try to describe concepts in non-symbolic or ‘sub-symbolic’ terms. A common approach is to use a kind of imagistic or diagrammatic notation. For example, in their analysis of the preposition *over*, Tyler & Evans (2003) identify a number of distinct senses, which they describe using diagrams, as shown in Figure 1.4. The diagram in (a) represents the most typical sense of *over* (the ‘proto-scene’), as in *the helicopter was over the field*, whereas the diagram in (b) represents the trajectory sense, as in *the cat jumped over the wall*. The claim is that “the meanings associated with many individual lexemes are instantiated in memory not in terms of features, nor as abstract propositions, but rather as imagistic, schematic representations” which “arise from perceptual reanalysis of recurring patterns in everyday physical experience”.

This kind of diagrammatic notation often gives the impression of being iconic, of directly portraying the intended concept in such a way as to remove the need for symbols. However, image schema notation is based on a number of conventions – dots represent entities, dotted lines represent relations, line thickness represents salience, arrows represent actions, and so on – which are no less symbolic in character than the kind of predicate calculus representations which the theory is intended to replace. As Jackendoff puts it,

> Consider also Deane’s treatment of *rise*, *fall*, *on* and *onto*. In his diagrams, the notation TR [trajector] is situated in some position relative to a line, which is close to the notation LM [landmark]. Nothing says that TR is supposed to be in contact with the line (as opposed to simply near it); nothing tells us that by contrast LM is not supposed to be near the line; it is supposed to *be* the line. If you know what the diagram is supposed to mean, it is sort of iconic, so it appeals to common sense in a way that the austerity of features and functions does not. But at bottom it is no more explicit, and no more psychologically real, than feature and function notation. [. . .] I have no objection in principle to using circles, squares and arrows instead of square brackets, parentheses and functions. We should just be very clear about their status. (1996, p. 110)

Far from removing the need for Markerese, image schema notation is simply another form of Markerese. Some reasons for preferring a logic-based formalism over image schema notation are that logic is less ambiguous, has an accompanying theory of inference, and can be understood by those in other disciplines such as philosophy and computer science.

For a conceptualist, the symbols of Markerese have genuine semantic content. This is not a matter of direct reference to external reality, as for Lewis. Nor is it due to the structure of symbols in some sense iconically resembling that to which they refer. Rather, the semantic content of a Markerese symbol derives, on the one hand, from its inferential role within Markerese itself, and on the other hand, from being appropriately connected to other cognitive systems such as vision, audition, motor control, emotional understanding, and so forth. For example, the Markerese translation for *cat* would be appropriately connected to other symbols such as those for *animal*, *feline*, *pet*, *human*, *dog*, *tail*, and so forth. At the same time, it would also be connected to information from visual cognition and spatial reasoning (the shape and configuration of a cat), audition (the sound of a cat), motor control (how to interact with a cat), and so on. A Markerese symbol is like a symbol inside a computer: it has no significance by itself, but only by virtue of its complex interaction with other symbols.

1.6 Outline of remaining chapters

As explained above, the aim of this thesis is to investigate a general phenomenon through a specific lens. The general phenomenon is the interaction between implicational networks and composition; the specific lens is the semantics of English spatial adjectives. My hope is to develop a formal, compositional framework capable of implementing a fine-grained approach to word meaning, and to give an impression of what formal semantics using such a framework might be like. Regarding spatial adjectives specifically, the aim is to reconcile the kind of detailed lexical analysis described in Section 1.3 with the compositional framework of degree semantics described in Section 1.4.

The goal of Chapter 2 is to introduce a formal framework for doing compositional semantics with implicational networks. I begin with a formal presentation of traditional Montague-style semantics, followed by a brief discussion of some of its limitations. I show how Montague semantics is not well suited to describing lexical networks because it does not treat proofs as first-class objects. This is followed by an introduction to Martin-Löf Dependent Type Theory, the formal framework which provides the setting for the rest of the thesis, and a discussion of what formal semantics looks like in Dependent Type Theory. I show how Dependent Type Theory can support a compositional semantics of lexical networks in which sense connections are fully-fledged semantic objects capable of being composed alongside senses.

Having outlined a general framework, Chapter 3 turns to a detailed case study of spatial adjectives. The aim of this chapter is to describe the lexical semantics of the English adjectives *high*, *low*, *tall*, *long*, *short*, *wide*, *narrow*, *deep shallow*, *thick* and *thin*, focusing primarily on their spatial meanings. Following Zwarts & Winter (2000), I take the basic semantic primitives in the domain of spatial language to be spatial vectors together with a collection of primitive predicates relating vectors and individuals. The meaning of an adjective is given by an implicational network of senses in which meets correspond to prototypes and joins correspond to abstract generalizations. This Chapter draws heavily on the lexical approaches to spatial adjectives outlined in Section 1.3.

Chapter 4 turns to degree semantics. The aim of this chapter is to formalise degree morphemes like *more/-er*, *most/-est*, *very*, *completely*, and so on, in such a way that they operate, not only on senses, but on entire lexical networks. My analysis of these morphemes is heavily based on previous approaches, particularly the work of Kennedy (1999, 2005, 2007). However, one difference with previous approaches, besides the incorporation of polysemy, is the focus on the dynamic aspect of degree morphology. Following Discourse Representation Theory, I adopt an approach to sentence meaning whereby presuppositional content, modelled as constraints on the background context, is separated from assertive content, modelled as an instruction for updating the background context with new information. The chapter concludes with a discussion of how dynamic semantics and polysemy might interact.

Finally, Chapter 5 summarizes the major contributions of the thesis and offers a brief comparison between my approach and other approaches which describe something resembling compositional lexical networks.

Chapter 2

Compositional Lexical Networks with Dependent Type Theory

Contents

2.1 Motivation

As discussed in the previous chapter, this thesis attempts to do justice to the following intuition. Consider a polysemous word, like the adjective *high*, which has various related senses covering spatial position (*high airplane*), spatial dimension (*high tower*), emotion (*to feel high*), social status (*high standing*), intensity (*high difficulty*), and so on. Some of these senses, such as spatial position and spatial dimension, are connected. When *high* occurs in a larger expression like *very high*, *higher than usual*, *too high*, and so on, the larger phrase inherits the polysemy of the adjective: it too has a position sense, a dimension sense, an emotion sense, a sense relating to social status, a sense relating to intensity, and so on. Crucially, if there is a connection between two original senses, then there should also be a connection between the derived senses. The positional and dimensional senses of *very high* should be connected in exactly the same way as the positional and dimensional senses of *high*. Some combinations, such as *high voice* or *150m high* select part of the *high* network, preserving certain senses and sense connections but not others.

This chapter will argue that the traditional setting for formal semantics, Simple Type Theory, is not rich enough to formalize a compositional theory of lexical networks. Instead, we need to extend Simple Type Theory by allowing types to depend on values, giving us what is called Martin-Löf Dependent Type Theory (DTT). DTT was originally developed in the 1970s by the Swedish Logician Per Martin-Löf as a foundation for constructive mathematics. Nowadays the original system has many variants, such as the Calculus of Constructions (Coquand 1986) and Homotopy Type Theory (UFP 2013); it also forms the basis for a number of programming languages, such as Coq (1997) and Agda (2009). The idea of basing natural language semantics on DTT is almost as old as the framework itself: proponents have included Sundholm (1989), Ranta (1994), Boldini (2000), Cooper (2005, 2012), and Luo (2012, 2020). There are many advantages to this framework besides the implementation of lexical networks, some of which I shall also touch upon.

The chapter is organised as follows. Section 2.2 introduces Simple Type Theory and explains its role in traditional Montague semantics. Section 2.3 discusses some of the limitations of Montague semantics which motivate the introduction of dependent types. Section 2.4 introduces and motivates the main ideas of DTT. Section 2.5 is an introduction to formal semantics in DTT, which explains how nouns, verbs, adjectives, and so on, can be analysed. Finally, Section 2.6 brings everything together by showing how DTT can support a compositional theory of lexical networks.

2.2 Semantics with Simple Type Theory

2.2.1 The basic setup

Before exploring what it means to do semantics in DTT, it is worth reviewing the more traditional Montague approach which is based on Simple Type Theory (STT). By Montague semantics, I mean any truth-conditional model-theoretic approach to natural language in which STT is used as an intermediary language to relate linguistic expressions to their set-theoretic denotations (e.g. Montague 1973, Partee 1986, Dowty et al. 2012). The basic architecture of this approach can be visualized as follows:

(55)
$$\begin{tikzcd} \text{natural language} \arrow[d, "\text{translation}"] \\ \text{STT expressions} \arrow[d, "\text{interpretation wrt a model}"] \\ \text{denotations} \end{tikzcd}$$

As shown, the denotation of an expression is given by first translating it into an expression of STT, and then interpreting this with respect to a model. Translation is a mapping from natural language to the formal language of STT, whereas interpretation with respect to a model M, written $[\![\cdot]\!]^M$, can be thought of as a mapping from linguistic to non-linguistic entities. There are three primary kinds of rules: *translation rules*, which govern the mapping from linguistic expressions to STT terms; *syntax rules*, which govern the formation of STT terms; and *semantic rules*, which govern the interpretation of STT terms with respect to a model. The basic goal of Montague semantics is to use the mapping illustrated in (55) to give a general description of the conditions under which sentences are true or false.

An important feature of this kind of framework is that the model is allowed to vary, the goal being to describe those aspects of meaning which remain invariant under a change of model. As Dowty et al. (2012, p. 45) puts it:

> The various choices of a model then are intended to represent the various ways we might effect the fundamental mapping from basic expressions to things in the world, while the fixed remainder represents the contribution to semantic values (and in particular, to truth values of sentences) made by the semantic theory itself.

One example of an aspect of meaning which survives a change of model are the interpretations of logical words such as *and*, *not*, *all*, *some*, and so on; Montague semantics is therefore ideally suited to studying this kind of vocabulary. On the other hand, the interpretation of content words like *dog*, *house*, *eat*, *see*, and so on, does not survive a change of model, so Montague semantics has comparatively little to say about them.

A key property of Montague semantics is *compositionality*: the meaning of a complex expression is derived from the meaning of its parts and the way they are syntactically combined. Syntactic combination is usually translated as *function application* in STT. For example, suppose that the translation of *John* is $\mathbf{John} : e$ and the translation of *eats* is $\mathbf{eats} : e \to t$ (I use bold text to distinguish an expression in natural language from its translation in STT). Then the translation of *John eats* is $\mathbf{eats}(\mathbf{John}) : t$. The semantic rules ensure that the composition of terms in STT is mirrored by the behaviour of denotations, so that, for instance, $[\![\mathbf{eats}(\mathbf{John})]\!]^M = [\![\mathbf{eats}]\!]^M([\![\mathbf{John}]\!]^M)$. Given this correspondence, we can draw a compositional version of the diagram in (55) as follows:

(56)
$$\begin{tikzcd} \text{syntactic combination} \arrow[d, "\text{translation}"] \\ \text{composition of STT expressions} \arrow[d, "\text{interpretation wrt a model}"] \\ \text{composition of denotations} \end{tikzcd}$$

2.2.2 The syntax of STT

I shall now introduce the syntax of STT as a formal type system. It is by no means necessary to set up STT in this manner (see Farmer 2008 for a simpler presentation); however, familiarity with type rules is crucial for Dependent Type Theory, so they are introduced here in preparation for Section 2.4. Moreover, since Dependent Type Theory is an extension of STT, the rules introduced here will carry over into further sections, albeit in a slightly modified form. A type system is set up by means of basic sentences called *judgements*, which have the following structure:

(57) $\Gamma \vdash d$

Γ stands for a typing environment or context, which is an ordered list of distinct variables together with their types, of the form $x_1 : A_1, \ldots, x_n : A_n$; d stands for some kind of declaration, the free variables of which must be drawn from Γ. To set up STT, we need four basic kinds of judgement, which are distinguished by the form of the declaration. These are listed below, together with an explanation of their meaning:

(58)

Γ ctx	Γ is a well-formed context
$\Gamma \vdash A$	A is a well-formed type in Γ
$\Gamma \vdash x : A$	x is a term of type A in Γ

$\Gamma \vdash x \equiv y : A \quad x$ and y are definitionally equal terms of type A in Γ

Note that it is important to distinguish between definitional equality (written $\equiv$) which is used to describe computational behaviour, and propositional equality (written $=$) which is a statement within the type system itself and has to be established by means of proof. This distinction will be explained in greater detail in Section 2.4.

Type rules are inference rules relating judgements. They assert that, given a collection of judgements known to be valid, some other judgement follows. A general type rule has the form:

(59) $$\frac{\Gamma_1 \vdash d_1 \quad \ldots \quad \Gamma_n \vdash d_n \quad (\text{extra conditions})}{\Gamma \vdash d} \,(\text{rule name})$$

$\Gamma_1 \vdash d_1, \ \ldots, \ \Gamma_n \vdash d_n$ are the premise judgements and $\Gamma \vdash d$ is the conclusion. To get the process of inference off the ground, there is an axiom to the effect that the empty context $\varnothing$ is a well-formed environment:

(60) $$\frac{}{\varnothing \ \text{ctx}} \,(\text{Empty Context})$$

Another fundamental rule states that, given a well-formed type A and a variable symbol x not already in the current context, one can extend the context by appending $x : A$

(61) $$\frac{\Gamma \vdash A \quad (x \text{ does not appear in } \Gamma)}{\Gamma, \, x : A \ \text{ctx}} \,(\text{Extend Context})$$

There is also a trivial rule stating that, given a context containing the declaration $x : A$, one can always conclude $x : A$

(62) $$\frac{\Gamma, \, x : A, \, \Gamma' \ \text{ctx}}{\Gamma, \, x : A, \, \Gamma' \vdash x : A} \,(\text{Memory})$$

The final component needed is a collection of *basic types*. A basic type is automatically well-formed in any context:

(63) $$\frac{\Gamma \ \text{ctx} \quad (T \in \mathit{Basic})}{\Gamma \vdash T} \,(\text{Basic Type Formation})$$

For the purposes of natural language semantics, we need at least the basic types e, for entities, and t, for truth values. For the purpose of exposition, I shall make the minimal choice $\mathit{Basic} \coloneqq \{e, t\}$. In more elaborate versions of Montague semantics, one also encounters types for possible worlds, events, times, degrees, and other semantic objects.

Having introduced some fundamental rules, which are sensible in any type system, we can now proceed to set up STT. We begin by introducing *function types*, the rules for which are given below:

(64)
$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \to B} \text{ (Function Formation)}$$

$$\frac{\Gamma,\ x : A \vdash b : B}{\Gamma \vdash (\lambda x : A\ .\ b) : A \to B} \text{ (Function Introduction)}$$

$$\frac{\Gamma \vdash f : A \to B \quad \Gamma \vdash x : A}{\Gamma \vdash f(x) : B} \text{ (Function Elimination)}$$

$$\frac{\Gamma,\ x : A \vdash b : B \quad \Gamma \vdash a : A \quad (a \text{ is free for } x \text{ in } b)}{\Gamma \vdash (\lambda x : A\,.\,b)(a) \equiv b[x \coloneqq a] : B} \text{ (Function Computation)}$$

$$\frac{\Gamma \vdash f : A \to B \quad (x \text{ is not free in } f)}{\Gamma \vdash f \equiv (\lambda x : A\,.\,f(x)) : A \to B} \text{ (Function Uniqueness)}$$

As shown, given two types A and B, one can form the function type $A \to B$. Terms of $A \to B$ are introduced by λ-abstraction in the usual way. Given a term of type $A \to B$ one can apply it to a term of type A to get a term of type B. The Function Computation rule, also known as β-reduction, tells us how to compute the result of an application: by substituting all occurrences of the bound variable with the argument expression. Finally, function Uniqueness, also known as η-reduction, ensures that lambda abstracting over the argument of a function yields the original function back.

In addition to function types, it is also useful to introduce *product types* or ordered pairs. The rules for products are as follows:

(65)
$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \times B} \text{ (Product Formation)}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash (a, b) : A \times B} \text{ (Product Introduction)}$$

$$\frac{\Gamma \vdash p : A \times B}{\Gamma \vdash \pi_1(p) : A} \text{ (Product Elimination 1)}$$

$$\frac{\Gamma \vdash p : A \times B}{\Gamma \vdash \pi_2(p) : B} \text{ (Product Elimination 2)}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash \pi_1(a, b) \equiv a : A} \text{ (Product Computation 1)}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash \pi_2(a, b) \equiv b : B} \text{ (Product Computation 2)}$$

$$\frac{\Gamma \vdash p : A \times B}{\Gamma \vdash p \equiv (\pi_1(p), \pi_2(p)) : A \times B} \text{ (Product Uniqueness)}$$

As shown, given two types A and B, we can form the type $A \times B$. Terms of $A \times B$ are ordered pairs of the form (a, b), where $a : A$ and $b : B$. Given an element of $A \times B$, we can project out the first component to get a term of type A, or the second component to get a term of type B. The computation rules ensure that the first projection of a pair (a, b) is the element a, and the second projection is the element b. The Product Uniqueness rule ensures that pairing together the first and second projections recovers the original pair.

It is useful for many purposes to introduce a *unit type*, which I shall write as $\top$. The unit type contains a single unique element, written $* : \top$, which may be introduced in any context:

(66) $$\frac{\Gamma\ \mathrm{ctx}}{\Gamma \vdash \top}\ (\text{Unit Formation})$$

$$\frac{\Gamma\ \mathrm{ctx}}{\Gamma \vdash * : \top}\ (\text{Unit Introduction})$$

$$\frac{\Gamma \vdash v : \top}{\Gamma \vdash v \equiv * : \top}\ (\text{Unit Uniqueness})$$

The Unit Uniqueness rule ensures that $* : \top$ is unique by requiring that any other term $v : \top$ is equal to it. The unit type plays a special role because given some other type A, there is a one-to-one correspondence between terms of A and functions $\top \to A$. For example, the type t has two elements, *true* and $false$, so there are exactly two functions of type $\top \to t$: the function which sends $*$ to *true*, and the function which sends $*$ to $false$.

2.2.3 The semantics of STT

Thus far, I have presented a very general type system which could be applied to many different natural languages. For a version of STT which can be used in the analysis of a particular language, e.g. English, we need what is known as a *signature* or *vocabulary*. An *STT signature* is a pair $\Sigma = (S, Type)$, where S is a collection of symbols, and $Type$ is a function assigning each symbol in S to an STT type. The symbols in S form the primitive constants of the type theory which are used to build larger terms. For example, Figure 2.1 shows an SST signature for a fragment of English, where I have indicated the type of each symbol by placing it on an arrow (for example, the symbol **Sally** lies on an arrow from $\top$ to e, so it has type $\top \to e$). Different word classes are associated with different logical types: proper names have type $\top \to e$; nouns, adjectives and intransitive verbs have type $e \to t$; transitive verbs have type $e \times e \to t$; quantificational determiners have type $(e \to t) \to (e \to t) \to t$, and so on. Some of the symbols in the signature, such as $\wedge$, $\vee$, $\exists$, $\forall$, $\neg$, and so on, have standard definitions in terms of lambda abstraction, function application, and equality.

STT is an interpreted language, whose semantics is given with reference to a background set theory. We begin by assigning an interpretation to the constants in the signature through a choice of model. Given an STT signature $\Sigma = (S, Type)$, a *model for* Σ is a pair $M = (D_\alpha, I)$, where:

(67)
- D_α is a family of domains (sets) for every type α, such that
  - D_t is the set of Booleans $\{0, 1\}$
  - $D_\top$ is the singleton set $\{*\}$

- $D_{A \to B}$ is the set of total functions from D_A to D_B
- $D_{A \times B}$ is the Cartesian product of D_A and D_B

- I is a function sending every symbol $s \in S$ to an element of $D_{Type(s)}$

Note that the only degree of freedom remaining for D_α is the choice of the domain of entities D_e: every other domain is then fixed by the rules.

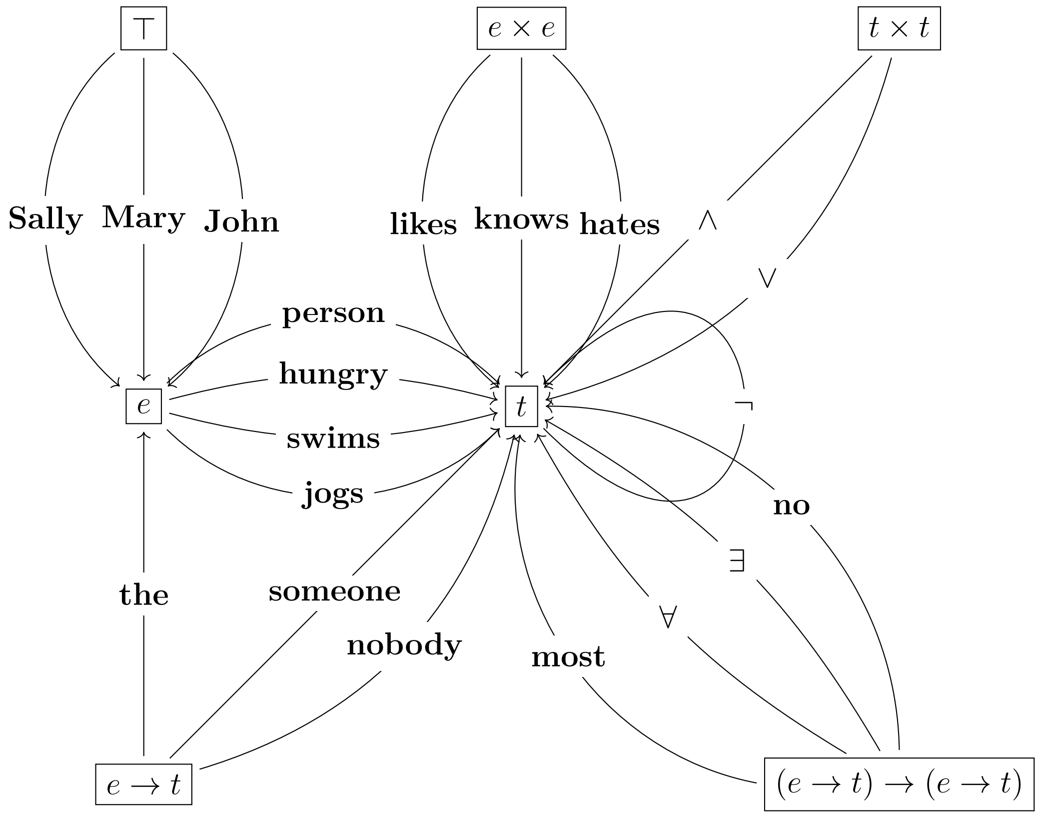


Figure 2.1: An STT signature for a fragment of English.

In order to interpret arbitrary terms, we need to be able to interpret not only the constants given by the signature but also variables. Given a model $M = (D_\alpha, I)$, a *variable assignment into* M is a function φ, mapping every variable $x : A$ to an element in D_A. Once we have a model and a variable assignment, we can interpret any term in the type system built on the signature. The interpretation function $[\![\cdot]\!]^{M,\varphi}$ is defined in (68):

(68) Given a model $M = (D_\alpha, I)$ and a variable assignment φ, an arbitrary term t is interpreted as follows:

- If t is a constant, then $[\![t]\!]^{M,\varphi} = I(t)$
- If t is a variable then $[\![t]\!]^{M,\varphi} = \varphi(t)$
- If t is of the form $f(x)$ then $[\![t]\!]^{M,\varphi} = [\![f]\!]^{M,\varphi}([\![a]\!]^{M,\varphi})$
- If t is of the form $(\lambda \mathrm{x} : A \,.\, b)$, where $b : B$, then $[\![t]\!]^{M,\varphi}$ is the function $f \in D_{A \to B}$ given by $f(a) = [\![b]\!]^{M,\varphi'}$, where φ' is like φ but with the additional assignment $x \mapsto a$

- If t is of the form (a, b) then $[\![t]\!]^{M,\varphi} = ([\![a]\!]^{M,\varphi}, [\![b]\!]^{M,\varphi})$

Roughly speaking, interpretation can be thought of as a map sending every type to a set, and every term to a function between sets, in a way which preserves the behaviour of function types and product types.

As mentioned, the goal of Montague semantics is to give a general account of the truth conditions of natural language sentences. This is now automatic from the way the theory is set up. A sentence $S : t$ is true with respect to a model M iff $[\![S]\!]^{M,\varphi} = 1$. For example, the sentence *John doesn't swim* is true with respect to M iff $[\![\textbf{John doesn't swim}]\!]^{M,\varphi} = [\![\textbf{not}]\!]^{M,\varphi}([\![\textbf{swims}]\!]^{M,\varphi}([\![\textbf{John}]\!]^{M,\varphi})) = 1$. Along with an account of truth, model-theoretic semantics also provides an concept of entailment. A sentence S_1 entails a sentence S_2 iff there is no model M in which $[\![S_1]\!]^M = 1$ but $[\![S_2]\!]^M = 0$. For example, *John is a hungry person* entails *John is a person* because in every model M where $[\![\textbf{John is a hungry person}]\!]^M = 1$, it is also the case that $[\![\textbf{John is a person}]\!]^M = 1$. A logical tautology is a sentence which is interpreted as 1 under every model.

2.3 Limitations of semantics with STT

The theory described in the previous section, together with modest extensions, has formed the basis for a great deal of work in semantics. It has been particularly successful at studying 'logical' aspects of language such as generalized quantifiers (Montague 1973, Barwise & Cooper 1981), negative polarity (Ladusaw 1980), plurals and mass terms (Link 1980, Carlson 1977), modality and conditionals, and gradability (Cresswell 1976). Because of the success of Montague-style semantics, any proposal which radically alters it requires a thorough justification. This section will examine various shortcomings with a semantics based on STT, focusing on those which motivate the introduction of Dependent Type Theory. Some of these limitations are well-known and already have established solutions; I argue that Dependent Type Theory is to be preferred over the existing solutions.

2.3.1 Proofs are not first-class objects

From the perspective of this thesis, the most important limitation of STT is that it does not implement proofs directly as elements of the type system. (More accurately, it does not implement proofs in predicate logic.) Proofs appear in lexical semantics as the arrows linking together the different senses in a lexical network. For example, recall the simple lexical network discussed by Jackendoff (2002, p. 353), consisting of three different senses of the verb *climb*:

(69) $\text{climb}_{\text{rise, clamber}}$ rising and clambering, e.g. *the man climbed the tree*

$\text{climb}_{\text{rise}}$ rising only, e.g. *the snake climbed the tree*

$\text{climb}_{\text{clamber}}$ clambering only, e.g. *the monkey climbed down the tree*

The three senses form the simple network shown below:

(70)
$$\begin{tikzcd}
\text{climb}_{\text{rise}} & & \text{climb}_{\text{clamber}} \\
& \text{climb}_{\text{rise, clamber}} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

The arrows can be thought of as proofs of universal statements. The arrow from $\text{climb}_{\text{rise, clamber}}$ to $\text{climb}_{\text{rise}}$ represents the proof that any instance of rising and clambering is also an instance of rising (by dropping the clambering component); and likewise the arrow from $\text{climb}_{\text{rise, clamber}}$ to $\text{climb}_{\text{clambering}}$ represents the proof that any instance of rising and clambering is an instance of clambering (by dropping the rising component).

How might a network like (70) be implemented in STT? One possibility would be a decompositional approach. Instead of introducing $\mathbf{climb}_{\mathbf{rise,\,clamber}}$, $\mathbf{climb}_{\mathbf{rise}}$ and $\mathbf{climb}_{\mathbf{clamber}}$ as basic predicates, one would instead define them in terms of simpler predicates. A potential decomposition is given below (assuming a neo-Davidsonian event semantics in which the type of events is added to the set of basic types):

(71) $\mathbf{climb}_{\substack{\mathbf{rise}\\\mathbf{clamber}}} := \lambda x.\lambda e.\mathbf{rising}(e) \wedge \mathbf{theme}(e, x) \wedge \mathbf{clambering}(e) \wedge \mathbf{agent}(e, x)$

$\mathbf{climb}_{\mathbf{rise}} := \lambda x.\lambda e.\mathbf{rising}(e) \wedge \mathbf{theme}(e, x)$

$\mathbf{climb}_{\mathbf{clamber}} := \lambda x.\lambda e.\mathbf{rising}(e) \wedge \mathbf{theme}(e, x)$

Given these definitions, one can prove that, for all x and e, $\mathbf{climb}_{\text{rise, clamber}}(x)(e)$ entails $\mathbf{climb}_{\text{rise}}(x)(e)$, and $\mathbf{climb}_{\text{rise, clamber}}(x)(e)$ entails $\mathbf{climb}_{\text{clamber}}(x)(e)$. However, this proof takes place entirely within the metalanguage of judgments, by chaining together inference rules; there is no way to 'internalise' the proof as a term of the type system itself.

Because proofs are not the kinds of things which can be taken and returned by functions, it is not possible to implement a compositional theory of lexical networks in STT. That is, it is not possible to implement a function which takes the network in (70) and returns a new network, mapping both senses and arrows, as follows:

(72)
$$
\begin{array}{ccc}
\text{climb}_{\text{rise}} & & [\text{climb quickly}]_{\text{rise}} \\
\uparrow & & \uparrow \\
\text{climb}_{\text{rise, clamber}} & \xmapsto{\text{quickly}} & [\text{climb quickly}]_{\text{rise, clamber}} \\
\downarrow & & \downarrow \\
\text{climb}_{\text{clamber}} & & [\text{climb quickly}]_{\text{clamber}}
\end{array}
$$

Instead, the best that we can do is operate on each sense individually, and then verify that the entailments still hold following composition by means of some post-hoc inference procedure, which we might visualize as follows:

(73)
$$
\begin{array}{ccc}
\text{climb}_{\text{rise}} & & [\text{climb quickly}]_{\text{rise}} \\
\uparrow & & \\
\text{climb}_{\text{rise, clamber}} & \xmapsto{\text{quickly}} & [\text{climb quickly}]_{\text{rise, clamber}} \\
\downarrow & & \\
\text{climb}_{\text{clamber}} & & [\text{climb quickly}]_{\text{clamber}}
\end{array}
$$

$$
\begin{array}{cc}
 & [\text{climb quickly}]_{\text{rise}} \\
 & \uparrow \\
\xmapsto{\text{(inference)}} & [\text{climb quickly}]_{\text{rise, clamber}} \\
 & \downarrow \\
 & [\text{climb quickly}]_{\text{clamber}}
\end{array}
$$

Assuming that the arrows in lexical networks instantiate useful information, a theory in which entailments are lifted automatically, as in (72), is preferable to one in which they are 'reconstructed' following composition, as in (73), because proof discovery is difficult to automate and the proofs required can often be quite complicated.

An alternative strategy for formulating the *climb* network in Montague semantics would be to make use of *meaning postulates* (Carnap 1952, Montague 1973). Meaning postulates are axioms which are provided alongside a signature Σ; the class of possible models for Σ is then restricted to those models which satisfy the axioms. In the case of the *climb* network, instead of decomposing the predicates $\mathbf{climb}_{\mathbf{rise,\,clamber}}$, $\mathbf{climb}_{\mathbf{rise}}$ and $\mathbf{climb}_{\mathbf{clamber}}$ as in (71), one would simply stipulate the relationship between them by introducing the following two axioms:

(74) $\forall x \,.\, \forall e \,.\, \mathbf{climb}_{\mathbf{rise,\,clamber}}(x)(e) \Rightarrow \mathbf{climb}_{\mathbf{rise}}(x)(e)$
$\forall x \,.\, \forall e \,.\, \mathbf{climb}_{\mathbf{rise,\,clamber}}(x)(s) \Rightarrow \mathbf{climb}_{\mathbf{clamber}}(x)(s)$

Any model would be required to satisfy these two axioms by insisting that:

(75) $[\![\forall x \,.\, \forall e \,.\, \mathbf{climb}_{\mathbf{rise,\,clamber}}(x)(e) \Rightarrow \mathbf{climb}_{\mathbf{rise}}(x)(e)]\!]^{M,\varphi} = 1$
$[\![\forall x \,.\, \forall e \,.\, \mathbf{climb}_{\mathbf{rise,\,clamber}}(x)(s) \Rightarrow \mathbf{climb}_{\mathbf{clamber}}(x)(s)]\!]^{M,\varphi} = 1$

for any M and φ. One could view these two meaning postulates in (74) as in some sense implementing the two arrows in the *climb* network.

Like proofs, meaning postulates are not the kind of things which can be taken and returned by functions, so it is difficult to make them compositional. There is a sense in which meaning postulates can be made compositional by adding additional postulates. For example, we could try to describe the lifting property of *quickly* via the following postulate:

(76) $\forall V, W \,.\, \forall x \,.\, \forall e \,.$
$(V(x)(e) \Rightarrow W(x)(e)) \Rightarrow$
$(V(x)(e) \wedge \mathbf{quickly}(e) \Rightarrow W(x)(e) \wedge \mathbf{quickly}(e))$

which states that, for all intransitive verbs V, W, if V entails W, then the intersection of V with **quickly** entails the intersection of W with **quickly**. I see two issues with such an approach. The first is that it requires a vast number of meaning postulates: one for every word capable of acting on an implicational network. The second is that it is not very explanatory: it does not tell us *why* intersection of two event predicates with **quickly** should preserve an arrow connecting them, but simply stipulates that this is the case. It is difficult to see how a rule like this could be discovered by a learner without being told it explicitly.

One could try to solve both of these problems simultaneously by replacing specific postulates like (76), which pertain to individual words, with more general postulates like 'intersection preserves implication', which apply in a wide variety of situations. However, then we would have to reason from the fact that *quickly* is an intersective predicate to the fact that it preserves implication, which is a more complicated process. The issue is that the more general meaning postulates become, the more they begin to resemble abstract rules of inference, and we are back with the problem of how to design an effective inference engine. A similar point is made by Jackendoff (1989) in relation to the following inference:

(77) a. x killed y $\Rightarrow$ y died

As Jackendoff points out, a system of meaning postulates must either explain this via a unique postulate, in which case it misses the common generalization ‘x caused E to occur ⇒ E occurred’ – or else it must encode this generalization itself as a meaning postulate, combined with a causal decomposition of the verb *kill*. In the latter case, there is little difference between applying a meaning postulate and discovering a proof.

To summarize, STT is not well-suited to describing a compositional theory of lexical networks. The basic problem is that it does not treat the arrows which link senses together as first-class objects which can be taken and returned by functions. As we shall see, Dependent Type Theory does treat proofs as first-class objects, allowing for a compositional theory of lexical networks with no need for either ad hoc rules or background inference.

2.3.2 Selectional restrictions

Another limitation of a semantics based on STT is describing the selectional restrictions which predicates place on their arguments. For example, consider the colour adjective *red*. In its basic non-metaphorical sense, *red* can only apply to a concrete physical object which is capable of having a colour. Hence *red ball* and *red dog* are fine, but ?*red meeting* and ?*red idea* are unacceptable. Similar kinds of restrictions apply to spatial adjectives, which are the subject of the next chapter. For example, the dimensional sense of the adjective *deep* requires a hollow object with an opening and an internal axis. Hence, *deep cup* and *deep cupboard* are acceptable, but ?*deep stick* and ?*deep slab* are not acceptable – rather, they trigger a different, ‘positional’ sense of *deep*. The same phenomenon occurs in verbs: for example, the verb *walk* generally requires an animate subject – a person or a dog can *walk* but not a car or a desk.

These kinds of restrictions are a problem for ordinary Montague semantics, where adjectives and verbs are generally assigned types like $e \to t$, which apply to all individuals. One potential solution is to preserve uniform types for adjectives and verbs, but treat anomalous combinations as contradictory. On this account, sentences like *X is a red idea* or *the car walked down the road* would be inherently false because they involve contradictory predicates. However, this fails to capture the intuitive distinction between nonsense sentences and sentences which are perfectly meaningful but false. A meaningful but false sentence like *Berlin is the capital of France* makes a clear claim about the world which happens not to be the case. A nonsense sentence like ?*Beauty is a red idea* fails to be true or false because it is not clear what it is claiming. It involves a kind of presupposition failure, the presupposition in this case being that ideas are the kinds of things which can have colours.

A better technique for dealing with selectional restrictions within STT is to encode them using the types themselves, by greatly expanding the number of basic types. The most developed system along these lines is that proposed by Asher (2011). Asher extends STT by the introduction of a *subtype* relation between types, which involves a new kind of judgement:

(78) $\Gamma \vdash A < B$ $\quad$ A is a subtype of B in Γ

The basic idea is that every value described by A is also described by B in a unique way, so any term of type A can be safely used by a procedure expecting a term of type B. Asher implements this behaviour through the following rules:

(79)
$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash A < B}{\Gamma \vdash a : B} \text{ (Subsumption)}$$

$$\frac{\Gamma \vdash A' < A \quad \Gamma \vdash B < B'}{A \to B < A' \to B'} \text{ (Function Subtypes)}$$

Having introduced the subtype relation, Asher then introduces many additional basic types, all of which are subtypes of e. Examples include PHYSICAL OBJECT, ANIMAL, INFORMATION, EVENT, INSTITUTION. The extended selection of basic types is used to represent the selectional restrictions of predicates. For example, **heavy** is assigned type PHYSICAL OBJECT $\to t$ as it selects a physical object, whereas **interesting** is assigned type INFORMATION $\to t$ as it selects an informational object. The Subsumption rule ensures that an entity belonging to a specific type can be described by a predicate on a more general type. For example, a member of ANIMAL can be described as **heavy**, because ANIMAL is a subtype of PHYSICAL OBJECT.

One problem with Asher's system which arises due to the structure of STT is a loss of type uniformity. For example, there can be no general type of intersective adjectives (the analogue of Montague's $e \to t$) because two intersective adjectives will in general have different selectional restrictions. Similarly, there is no unified type of intransitive verbs, transitive verbs, gradable adjectives, and so on. Type uniformity is important because one often needs to write functions which can take elements with distinct selectional restrictions. For example, a tense morpheme like *-ed* takes an arbitrary verb meaning, and the degree morpheme *more/-er* takes an arbitrary gradable adjective. In Dependent Type Theory, this problem does not arise because one can write functions which are parametric in the selectional restrictions of predicates. However, this requires constructions such as type universes and the dependent product, which are not available in STT.

Another issue with Asher's system is that it appears to duplicate the information which is already available in predicates. Asher introduces types like ANIMAL, INFORMATION, PERSON, and so on, which are also encoded by nouns *animal*, *information*, *person*, with no explanation of the connection between the two, other than a stipulation that any noun meaning can give rise to a basic type. Ideally, one would like to avoid this duplication by either identifying noun denotations with types, or somehow deriving types like ANIMAL, INFORMATION and PERSON from their corresponding noun denotations. However, neither of these two options will work in STT because there is no mechanism for relating predicates and types.

In the literature on Montague semantics, one sometimes sees the following kind of notation:

(80) $\lambda x : e \mid P(x)$

to indicate a partial function which is defined only for those individuals which satisfy the predicate P. For example, Kennedy (2007, p. 14) proposes that combinations of the form 'Adj for a NP', e.g. *expensive for a Honda*, should be analysed as partial functions from $e \to d$ (where d is the type of degrees) whose domain is restricted by the NP denotation:

(81) $\lambda x : e \mid \mathbf{honda}(x) \,.\, \mathbf{expensive}(x)$

One can imagine implementing the selectional restrictions of adjectives and verbs along similar lines. For example, the adjective *red* might be translated as the following partial function from $e \to t$:

(82) $\lambda x : e \mid \mathbf{has\text{-}colour}(x) \,.\, \mathbf{red}(x)$

ensuring that the predicate **red** is only applied to individuals satisfying the condition **has-colour**. In STT, it is not immediately clear how an expression like (82) should be typed, or what its semantics ought to be, since STT only allows for total functions. In Dependent Type Theory, on the other hand, this kind of notation makes perfect sense, and we can think of '$e \mid \mathbf{has\text{-}colour}(x)$' as a type, namely the type of all entities which satisfy **has-colour**. A partial function from $e \to t$ is simply a total function whose domain is some restricted subtype of e. The more standard notation for $e \mid \mathbf{has\text{-}colour}(x)$ in Dependent Type Theory is:

(83) $\sum_{x:e} \mathbf{has\text{-}colour}(x)$

This is an example of a *dependent product type*, which is explained in Section 2.4.2.

2.3.3 Dynamic interpretation

Another limitation of traditional Montague semantics which motivates the introduction of Dependent Type Theory is the well-known problem of binding without c-command. Consider the following example:

(84) Jane owns a cat. <u>It</u> is grey.

The pronoun of *it* in the second sentence should be co-referential with the entity Jane's cat, which was introduced by the first sentence. This causes a problem for traditional Montague semantics, where the expression *a cat* is not translated as a referential expression, but rather as an existential quantifier. (This analysis seems to be forced upon us by sentences like *Jane does not own a cat*, which do not permit subsequent reference to Jane's cat.) Theoretical approaches aimed at solving this problem go under the heading of *dynamic semantics*. They share the feature that interpreting a sequence of sentences involves the growth of information in time, with each new sentence providing additional information. Certain indefinite noun phrases can introduce discourse referents into the context.

Approaches to dynamic semantics fall into two major groups. On the one hand there are frameworks like Discourse Representation Theory (Kamp 1980) where the primary objects of interpretation are discourses or narratives. On this view, it is an entire discourse which has truth conditions, the role of an individual sentence being to prompt the hearer to modify their representation of the discourse. This necessitates a radical revision of the meaning language, which becomes a language of discourse contexts. On the other hand, there are approaches like Dynamic Predicate Logic (Groenendijk & Stokhof 1991), where the syntax of the meaning language is much closer to that of Montague semantics, but sentences are analysed as denoting relations between information states, modelled as variable assignments. The Dependent Type Theory approach to dynamic semantics is more similar to the approach of Discourse Representation Theory: contexts are modelled as syntactic objects which get updated by sentences. However, it is also different from both approaches in that the meaning language is not interpreted in some further set-theoretic model.

The advantage of the Dependent Type Theory approach to dynamic semantics is that there is no need to introduce anything into the theory beyond what is fundamentally available. As many authors have noted (e.g. Sundholm 1989, Ranta 1994, Boldini 2000, Chatzikyriakidis & Luo 2014), the basic ideas of dynamic semantics are already latent within type theory itself. As we saw in Section 2.2.2, terms in type theory are not formed in a vacuum, but depend on variables of particular types. The collection of variables in use at any one time, together with their associated types, forms the context Γ. The form of a variable may depend on that of a variable introduced earlier. In Dependent Type Theory, not only the form but also the *type* of a variable can depend on a previously introduced variable. This means that contexts to be used to represent information states of a discourse or narrative. For example, consider the following small story:

(85) A woman owns a cat. She loves it very much. One day, she returns from work and the cat is missing. She searches a nearby field and finds it. It is stuck in a fence.

In a type theoretic approach to dynamic semantics (e.g. Ranta 1994), this can be represented by one long context, as follows:

(86) $x_1 : e,\ x_2 : e,\ x_3 : Woman(x_1),\ x_4 : Cat(x_2),\ x_5 : Loves(x_1, x_2),\ x_7 : Returns(x_1),\ x_8 : Missing(x_2),\ x_9 : Searches\text{-}For(x_1, x_2),\ \ldots$

Notice how the dependence structure of the context mirrors that of the narrative. Variables introduced earlier, such as the woman and the cat, are used to form the types of variables introduced later. Facts and situations, such as the fact that x_1 is a woman, or the event of x_1 searching for x_2, are also assigned labels and can be referred to later in the discourse. Note that (86) does not make sense in STT because it includes types like $Woman(x_1)$ and $Loves(x_1, x_2)$ which depend on terms.

In addition to its inherent treatment of contexts, Dependent Type Theory automatically implements the idea that existential quantification introduces a new referent. In fact, this was the original motivation for type theoretic semantics (Sundholm 1989). Consider the following version of the famous 'Donkey sentence':

(87) If John owns a Donkey then he beats it.

This causes a problem for traditional Montague semantics because the standard translation of *a Donkey* as an existential quantifier leaves the variable *it* unbound:

(88) $\exists x[\mathbf{donkey}(x) \wedge \mathbf{owns}(\mathbf{John}, x)] \Rightarrow \mathbf{beats}(\mathbf{John}, x)$

Moreover, trying to widen the scope of the quantifier to incorporate the unbound x results in an incorrect translation:

(89) $\exists x[\mathbf{donkey}(x) \wedge \mathbf{owns}(\mathbf{John}, x) \Rightarrow \mathbf{beats}(\mathbf{John}, x)]$

which is satisfied by substituting any non-donkey object for x. In Dependent Type Theory, a proposition is represented by a type whose terms are proofs of the proposition. Giving a proof that John owns a donkey means exhibiting an individual which is both a donkey and is owned by John. This is expressed by the following type:

(90) $\sum_{x:e}(\mathbf{donkey}(x) \times \mathbf{owns}(\mathbf{John},\ x))$

As the notation suggests, this type plays a similar role to the expression $\exists x[\mathbf{donkey}(x) \wedge \mathbf{owns}(\mathbf{John}, x)]$. However, a crucial difference is that, given a term of this type, we can *access* the individual x using the first projection π_1. This allows us to write a type corresponding to the entire sentence, which is given below:

(91) $\prod z : \left(\sum_{x:e}(\mathbf{donkey}(x) \times \mathbf{owns}(\mathbf{John},\ x))\right) \mathbf{beats}(\mathbf{John},\ \pi_1(z))$

A term of this type is a function which takes a proof that John has a donkey and returns a proof that he beats it. This is a well-formed version of the non-well-formed formula in (88). Other dynamic frameworks also provide an analysis of Donkey sentences, but the Dependent Type Theory solution is particularly natural as it follows automatically from the introduction of the Π and Σ constructions. As Sundholm puts it:

> In this manner, the type-theoretic abstractions suffice to solve the problem of the pronomial back-reference in [Donkey sentences]. It should be noted that there is nothing *ad hoc* about the treatment, since all the notions used have been introduced for mathematical reasons in complete independence of the problem posed by [Donkey sentences]. (1989, p. 503)

The meaning of the Π and Σ symbols is explained further in the next section.

2.4 Introduction to Dependent Type Theory

2.4.1 Propositions as types

The starting point for Dependent Type Theory (DTT) is a precise analogy between the behaviour of types and the behaviour of propositions. I shall begin by introducing this correspondence in the context of STT. Recall the product construction given in Section 2.2.2, which given any two types A and B allows us to form a new type $A \times B$. We can deconstruct an element of $A \times B$ by projecting out the first component to get an element of A, or the second component to get an element of B. Now consider the standard natural deduction rules for logical conjunction, given below:

(92)
$$\frac{A \quad B}{A \wedge B} \text{ (Conjunction Introduction)}$$

$$\frac{A \wedge B}{A} \text{ (Conjunction Elimination 1)}$$

$$\frac{A \wedge B}{B} \text{ (Conjunction Elimination 2)}$$

Notice how these exactly parallel the rules Product Introduction, Product Elimination 1 and Product Elimination 2 given in Section 2.2.2. We are led to the conclusion that logical conjunction behaves in exactly the same way as the product: if we think of the two propositions A and B as types, then the proposition $A \wedge B$ corresponds to the type $A \times B$. In a similar way, the proposition $A \Rightarrow B$ ('A implies B') corresponds to the function type $A \to B$. Again, this can be easily seen from the natural deduction rules for implication: Implication Introduction corresponds to lambda abstraction and Modus Ponens corresponds to applying a function to an argument.

As my notation suggests, the type-theoretic analog of the true proposition is the unit type $\top$. To complete the correspondence between type theory and propositional logic, we need constructions corresponding to the false proposition, logical disjunction, and negation. The analogue of the false proposition is the *empty type*, which I shall write $\bot$. It is defined by the rules:

(93)
$$\frac{\Gamma \text{ ctx}}{\Gamma \vdash \bot} \text{ (Empty Formation)}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash e : \bot}{\Gamma \vdash \text{abort}_{\text{A}}(e) : A} \text{ (Empty Elimination)}$$

$$\frac{\Gamma \vdash e : \bot \quad \Gamma \vdash A \quad \Gamma \vdash x : A}{\Gamma \vdash \text{abort}_A(e) \equiv x : A} \text{ (Empty Uniqueness)}$$

As shown, given a term of $\bot$, one can construct an arbitrary term $x : A$, where A is any type. This is the analogue of the logical principle that the false proposition implies every other proposition (the 'Principle of Explosion'). As in logic, it is impossible to construct a proof of $\bot$, since there is no introduction rule.

The type-theoretic analog of logical disjunction $A \vee B$ is the sum type $A + B$. A term of type $A + B$ is either a term of type A, tagged as 'inl', or a term of type B, tagged as 'inr'. The rules are given below[1]:

(94)
$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A + B} \text{(+ Formation)}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash B}{\Gamma \vdash \mathrm{inl}(a) : A + B} \text{(+ Introduction 1)}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash b : B}{\Gamma \vdash \mathrm{inr}(b) : A + B} \text{(+ Introduction 2)}$$

$$\frac{\Gamma \vdash s : A + B \quad \Gamma, x : A \vdash c_1 : C \quad \Gamma, y : B \vdash c_2 : C}{\Gamma \vdash \begin{pmatrix} \text{case s of inl}(x) \text{ then } c_1 \\ \hfill \mid \text{inr}(y) \text{ then } c_2 \end{pmatrix} : C} \text{(+ Elimination)}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A \vdash c_1 : C \quad \Gamma, y : B \vdash c_2 : C}{\Gamma \vdash \begin{pmatrix} \text{case inl}(a) \text{ of inl}(x) \text{ then } c_1 \\ \hfill \mid \text{inr}(y) \text{ then } c_2 \end{pmatrix} \equiv c_1[x \coloneqq a] : C} \text{(+ Computation 1)}$$

$$\frac{\Gamma \vdash b : B \quad \Gamma, x : A \vdash c_1 : C \quad \Gamma, y : B \vdash c_2 : C}{\Gamma \vdash \begin{pmatrix} \text{case inr}(b) \text{ of inl}(x) \text{ then } c_1 \\ \hfill \mid \text{inr}(y) \text{ then } c_2 \end{pmatrix} \equiv c_1[y \coloneqq b] : C} \text{(+ Computation 2)}$$

$$\frac{\Gamma \vdash s : A + B \quad \Gamma, h : A + B \vdash c : C}{\Gamma \vdash \begin{pmatrix} \text{case } s \text{ of inl}(x) \text{ then } c[h \coloneqq \text{inl}(x)] \\ \hfill \mid \text{inr}(y) \text{ then } c[h \coloneqq \text{inr}(y)] \end{pmatrix} \equiv c[h \coloneqq s] : C} \text{(+ Uniqueness)}$$

As shown, a term $a : A$ is included in the sum type as $\mathrm{inl}(a) : A + B$, whereas a term $b : B$ is included in the sum type as $\mathrm{inr}(b) : A + B$. The Sum Elimination rule states that, to use a term of $A + B$, we should specify what should be done for the two ways in which it could have been constructed. The notation I shall use for this is a case expression with two branches, as shown in the rule. The computation rules ensure that an element tagged with inl triggers the left branch of the case expression, whereas an element tagged with inr triggers the right branch. Finally, the Sum Uniqueness rule ensures that unpacking an element of $A + B$ and then packing this back into $A + B$ is the same as not unpacking at all. The rules Sum Introduction 1, Sum Introduction 2 and Sum Elimination are the type-theoretic analogs of the rules of disjunction in logic.

If each proposition is really a type then how should we think about its terms? The answer is the terms of a proposition P are *proofs* or *guarantees* that P is true. Proving P corresponds to constructing a term $p : P$. A crucial point is that, before a term of P has been constructed, P cannot be assigned a definite truth value. The logic of type theory is therefore a *constructive* or *intuitionistic* logic. It lacks

[1] All the rules involving substitution have an additional 'freeness' requirement. For example, to form the expression $c_1[x \coloneqq a]$, a must be free for x in c_1.

the classical laws of Excluded Middle ($P \vee \neg P$) and Double Negation ($\neg\neg P \Rightarrow P$) which are not valid from the point of view of a finite agent whose knowledge is bounded by what they can prove. Instead, Excluded Middle and Double Negation become theorems which hold only in certain domains. Intuitionistic logic therefore applies to a broader range of situations than classical logic, just as geometry without Euclid's parallel postulate applies to a broader range of situations than Euclidean geometry. In an intuitionistic logic, the negation of a proposition P is defined as:

(95) $\neg P \equiv P \Rightarrow \bot$

That is, proving $\neg P$ means showing that given a proof of P, one can derive a proof of $\bot$. Since there are no proofs of $\bot$, it follows that P itself must not have any proofs. This definition carries over into type theory, where it is written as:

(96) $\neg P \equiv P \rightarrow \bot$

In other words, an element of $\neg P$ is a function from P to the empty type.

We have now informally described a complete correspondence between intuitionistic propositional logic and STT, which is known as the *Curry-Howard correspondence*. It is summarized in the following table:

(97)

intuitionistic logic	STT
a proposition A	a type A
a proof of A	a term $x : A$
conjunction $A \wedge B$	product type $A \times B$
disjunction $A \vee B$	sum type $A + B$
implication $A \Rightarrow B$	function type $A \rightarrow B$
true $\top$	the unit type $\top$
false $\bot$	the empty type $\bot$
negation $\neg A$	the type $A \rightarrow \bot$

Note that we do not yet have a fully fledged *predicate* logic, because we do not have type-theoretic analogs of predicates and quantifiers: this is what is added by DTT. It is worth pointing out that, although STT has its own internal logic – intuitionistic propositional logic – this is not the kind of logic which is relevant to traditional Montague semantics. Rather, Montague semantics uses STT to implement the syntax of classical higher-order logic, whose semantics is then given in model-theoretic terms.

If every proposition corresponds to a type, does every type correspond to a proposition? In the original version of DTT described by Martin-Löf (1984), the answer was yes: even types which represent sets have a propositional interpretation. However, modern versions of DTT, such as Homotopy Type Theory (UFP 2013), work with a more restricted notion of proposition, according to which a proposition is a type with *at most one inhabitant*. The idea is that, given a proposition P, the important question is whether P is true (inhabited) or false (uninhabited): we are not interested in keeping track of which particular proof of P is being used. This is in contrast to set-like types (e.g. e, t), where we care about which element of the set we are working with. The collection of all propositions is denoted *Prop*. The correspondence given in (97) should be thought of as holding when A and B are elements of *Prop*.

2.4.2 Dependent types

The key innovation of DTT is that types are permitted to depend on terms. This is achieved by blurring the distinction between terms and types, so that types themselves belong to types. Every judgement of the form:

(98) $\Gamma \vdash A$

meaning 'A is a well-formed type in Γ', is replaced by a typing judgement of the form:

(99) $\Gamma \vdash A : \mathit{Type}_i$

meaning 'A belongs to the i-th level type universe'. All the types we have encountered up to this point – including basic types like e, t and $\top$, as well as all function and product types derived from them – are members of the 0th level type universe Type_0. Type_0 is not a member of itself, which would be inconsistent: rather, it belongs to the 1st level type universe Type_1, which in turn belongs to Type_2, and so on in an infinite hierarchy. The rules for type universes are:

(100) $$\frac{\Gamma\ \text{ctx}}{\Gamma \vdash \mathit{Type}_i : \mathit{Type}_{i+1}}\ \text{(Universe Introduction)}$$

$$\frac{\Gamma \vdash A : \mathit{Type}_i}{\Gamma \vdash A : \mathit{Type}_{i+1}}\ \text{(Cumulative Universes)}$$

Since the majority of types in this thesis belong to the 0th level I shall often refer to this without a subscript, writing *Type* for Type_0.

Unlike in STT, Types are allowed to depend on variables from the context. For example, given a natural number n, we can set up the type $\mathit{Leq}(n)$, whose terms are the natural numbers less than or equal to n. This type would be written as follows:

(101) $n : \mathit{Nat} \vdash \mathit{Leq}(n) : \mathit{Type}$

Each natural number gives rise to a distinct type: $\mathit{Leq}(0)$, $\mathit{Leq}(1)$, $\mathit{Leq}(2)$, and so forth. Dependent types can be used to represent predicates. For example, the dependent type:

(102) $n : \mathit{Nat} \vdash \mathit{IsEven}(n) : \mathit{Type}$

expresses the fact that n is an even number. Unlike $\mathit{Leq}(n)$, which is set-like and can have many distinct inhabitants, $\mathit{IsEven}(n)$ is a proposition, meaning it is has at most one inhabitant. Given some n, an element of $\mathit{IsEven}(n)$ is a proof that n is even: hence $\mathit{IsEven}(0)$, $\mathit{IsEven}(2)$ and $\mathit{IsEven}(4)$ are inhabited, whereas $\mathit{IsEven}(1)$, $\mathit{IsEven}(3)$ and $\mathit{IsEven}(5)$ are uninhabited.

Given a dependent type, there are two fundamental constructions: the *dependent sum* and the *dependent product*. I shall introduce these constructions by presenting an intuitive example, followed by the general rules. Consider the dependent type $n : Nat \vdash Leq(n) : Type$ assinging to each $n : Nat$ the type of numbers less than or equal to n. The dependent sum, written

(103) $\sum_{n:Nat} Leq(n)$

consists of all pairs (n, m), where $n : Nat$ and $m : Leq(n)$. Notice that the type of the second component depends on the value of the first component: 0 is paired only with elements of $Leq(0)$, 1 only with elements of $Leq(1)$, 2 only with elements of $Leq(2)$, and so on. We can visualize the type $\sum_{n:Nat} Leq(n)$ as follows:

(104)
$$\boxed{\begin{array}{cccccc} & & & & (4,4) & \\ & & & (3,3) & (4,3) & \\ & & (2,2) & (3,2) & (4,2) & \\ & (1,1) & (2,1) & (3,1) & (4,1) & \\ (0,0) & (1,0) & (2,0) & (3,0) & (4,0) & \cdots \end{array}}$$

Notice the difference between this type and the product type $Nat \times Nat$, in which every natural number is paired with every other. The rules for the dependent sum are given in (105):

(105)
$$\frac{\Gamma \vdash A : Type_i \quad \Gamma, x : A \vdash B(x) : Type_i}{\Gamma \vdash \sum_{x:A} B(x) : Type_i} \ (\Sigma \text{ Formation})$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B(a)}{\Gamma \vdash (a, b) : \sum_{x:A} B} \ (\Sigma \text{ Introduction})$$

$$\frac{\Gamma \vdash p : \sum_{x:A} B(x)}{\Gamma \vdash \pi_1(p) : A} \ (\Sigma \text{ Elimination 1})$$

$$\frac{\Gamma \vdash p : \sum_{x:A} B(x)}{\Gamma \vdash \pi_2(p) : B(\pi_1(p))} \ (\Sigma \text{ Elimination 2})$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B(a)}{\Gamma \vdash \pi_1(a, b) \equiv a : A} \ (\Sigma \text{ Computation 1})$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B(a)}{\Gamma \vdash \pi_2(a, b) \equiv b : B(a)} \ (\Sigma \text{ Computation 2})$$

$$\frac{\Gamma \vdash p : \sum_{x:A} B(x)}{\Gamma \vdash p \equiv (\pi_1(p), \pi_2(p)) : \sum_{x:A} B(x)} \ (\Sigma \text{ Uniqueness})$$

As shown, terms of a dependent sum are introduced by pairing and eliminated by projection, just like the ordinary product type, and are subject to similar computation and uniqueness rules, the only difference being that the type of the second component is allowed to depend on the value of the first. The ordinary product can be defined as a version of dependent sum in which there is no dependence, as follows:

(106) $A \times B :\equiv \sum_{x:A} B$

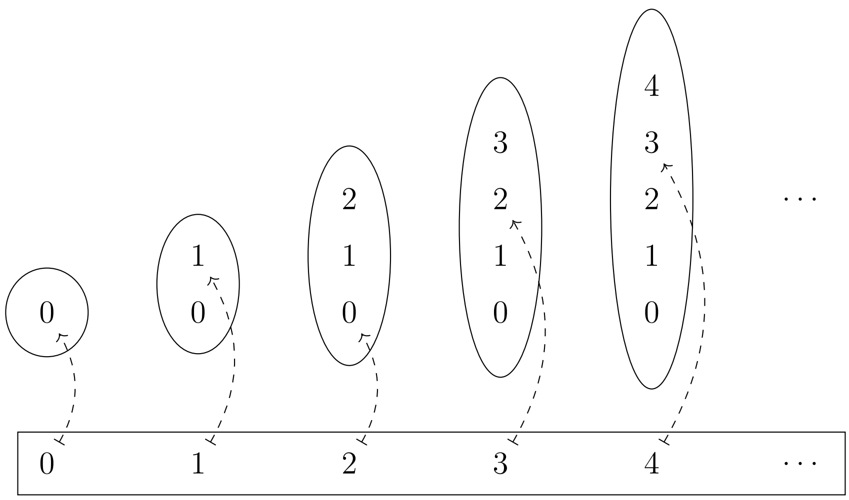
In practice, I shall continue to write the ordinary product using $\times$, since this notation is more intuitive.

The other fundamental operation on a dependent type is the *dependent product*. Returning to our example of the dependent type $n : Nat \vdash Leq(n)$, the dependent product is written:

(107) $\prod_{n:Nat} Leq(n)$

An element of this type is a function sending each natural number n to a number in $Leq(n)$. The range of the function is not fixed but varies depending on the value of its input: 0 must be sent to a number in $Leq(0)$, 1 to a number in $Leq(1)$, 2 to a number in $Leq(2)$, and so on. This is different to the ordinary function type, where the range remains fixed across different input values. We can visualize an element of $\prod_{n:Nat} Leq(n)$ as follows:

(108)



As indicated, each such function has one choice for where it sends 0, two choices for where it sends 1, three choices for where it sends 2, and so forth. It should be clear that there are far fewer elements of $\prod_{n:Nat} Leq(n)$ than of the function type $Nat \to Nat$, where there are an infinite number of choices for where to send each natural number. The rules for the dependent product are:

(109)
$$\frac{\Gamma \vdash A : Type_i \quad \Gamma, x : A \vdash B(x) : Type_i}{\Gamma \vdash \prod_{x:A} B(x) : Type_i} \ (\Pi \text{ Formation})$$

$$\frac{\Gamma, x : A \vdash b : B(x)}{\Gamma \vdash (\lambda x : A \,.\, b) : \prod_{x:A} B(x)} \ (\Pi \text{ Introduction})$$

$$\frac{\Gamma \vdash f : \prod_{x:A} B(x) \quad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B(a)} \ (\Pi \text{ Elimination})$$

$$\frac{\Gamma, x : A \vdash b : B(x) \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x : A \,.\, b)(a) \equiv b[x := a] : B(a)} \ (\Pi \text{ Computation})$$

$$\frac{\Gamma \vdash f : \prod_{x:A} B(x)}{\Gamma \vdash f \equiv (\lambda x : A \,.\, f(x)) : \prod_{x:A} B(x)} \ (\Pi \text{ Uniqueness})$$

As shown, terms of a dependent product are introduced by lambda abstraction and eliminated by application, in the same way as ordinary functions, and obey analogous computation and uniqueness rules, the only difference being that the range type can depend on the input value. The ordinary function type is a special case of the dependent product, which can be defined as as follows:

(110) $A \to B \equiv \prod_{a:A} B$

As before, I shall continue to use the symbol $\to$ to denote the ordinary function type.

When they operate on propositions, the dependent sum and dependent product correspond to existential and universal quantification respectively. For an intuitive sense of why this is the case, recall the predicate

(111) $n : Nat \vdash IsEven(n) : Type$

where $IsEven(n)$ expresses the proposition that n is an even number. Now consider the proposition 'there exists an even number'. In an intuitionistic logic, proving an existential statement requires exhibiting an object which satisfies the relevant property, so a proof of 'there exists an even natural number' must exhibit a natural number n together with a proof that n is even. Such a proof is precisely a term of the dependent sum type $\sum_{n:Nat} IsEven(n)$. Conversely, consider what it would mean to have a term of the dependent product type $\prod_{n:Nat} IsEven(n)$. This would be a function taking every natural number n to a proof that n is even. In other words, it would constitute a proof of the universal statement 'every natural number is even'. Of course, there is no such proof, so the type $\prod_{n:Nat} IsEven(n)$ is uninhabited. In contrast, the type $\prod_{n:Nat} IsEven(n) + IsOdd(n)$, representing the proposition 'every natural number is even or odd', is inhabited. (Its inhabitant is an inductive proof that every natural number is either exactly divisible by 2 or divisible by 2 with remainder 1.) We can now extend the correspond between intuitionistic logic and type theory to include predicates and quantifiers, as follows:

(112)

intuitionistic logic	type theory
a predicate $B(x)$	a dependent type $x : A \vdash B(x) : Type$
existential quantification $\exists x.B(x)$	dependent sum type $\sum_{x:A} B(x)$
universal quantification $\forall x.B(x)$	dependent product type $\prod_{x:A} B(x)$

2.4.3 Identity types

There is one important concept missing from the table in (112) which is needed to have a fully-fledged logic, namely *propositional equality*. Given any two elements x and y belonging to the same type A, we introduce an *identity type*, written $x =_A y$, which encodes the proposition that x and y are equal terms of type A. As mentioned previously, this should not be confused with *definitional equality* $x \equiv y : A$, which is an assertion that x and y are equal by definition, i.e. reduce to the same normal form. The following rules for identity types will be relevant to this thesis:

(113)
$$\frac{\Gamma \vdash A : Type_i \quad \Gamma \vdash x : A \quad \Gamma \vdash y : A}{\Gamma \vdash x =_A y : Type_i} \text{ (Identity Formation)}$$

$$\frac{\Gamma \vdash A : Type_i \quad \Gamma \vdash x : A}{\Gamma \vdash \mathrm{refl}_x : x =_A x} \text{ (Identity Introduction)}$$

The introduction rule ensures that any term $x : A$ can be considered trivially equal to itself through the automatic introduction of a reflexivity term $\mathrm{refl}_x : x =_A x$. It follows from Identity Introduction that definitional equality implies propositional equality, since if we have that $x \equiv y : A$, we can get a proof of $x =_A y$ by substitution. The reverse does not hold: two terms can be propositionally equal without being equal by definition. For the elimination and computation rules for identity types, readers should consult the standard textbook on Homotopy Type Theory (UFP 2013, Section 1.12) which covers this topic at length.

Two facts about identity types will be of particular importance in this thesis. The first is that, for any type A and elements $x, y, z : A$, there is a function of type:

(114) $(x =_A y) \times (y =_A z) \to (x =_A z)$

which performs the *concatenation* of identity proofs. Given some $p : x =_A y$ and $q : y =_A z$, the concatenation is written $p \bullet q$. Reflexivity proofs act as identities with respect to concatenation, so that $\mathrm{refl}_x \bullet p = p \bullet \mathrm{refl}_y = p$. The second important fact is that identity proofs are automatically lifted by functions. For any function $f : A \to B$, there is an operation:

(115) $\mathrm{ap}_f : (x =_A y) \to (f(x) =_B f(y))$

which takes identities of elements in A to identities between their images in B. Application of a function respects composition and reflexivity proofs. In other words, we have that:

(116)
$$\mathrm{ap}_f(p \bullet q) = \mathrm{ap}_f(p) \bullet \mathrm{ap}_f(q)$$
$$\mathrm{ap}_f(\mathrm{refl}_x) = \mathrm{refl}_{f(x)}$$

For definitions of concatenation and the ap operation, I again refer the reader to the Homotopy Type Theory textbook, Sections 2.1 and 2.2.

We can use identity types to formally define what is meant by a proposition. Recall that propositions are supposed to be those types with at most one element. A type A is a proposition iff the following type is inhabited:

(117) $IsProp(A) \coloneqq \prod_{x:A} \prod_{y:A} (x =_A y)$

That is, for every two terms $x, y : A$, there should be a proof that they are equal (if A happens to be empty, then this is automatically satisfied). The type *Prop* is then defined as a dependent sum over elements of a type universe:

(118) $Prop \coloneqq \sum_{A:Type} IsProp(A)$

Any type A can be 'truncated' to a proposition, written $|A|$, by forcing all of its terms to be equal. $|A|$ is inhabited if there is at least one term of type A, and uninhabited otherwise. An important fact about propositional truncation is that if B is a proposition and we have some function $f : A \to B$, then we get an automatic function $|f| : |A| \to B$. In other words, any proposition which follows from an element of A follows from the mere fact that A is inhabited. For more details on the truncation operation, see UFP (2013, Section 3.7).

2.4.4 Useful notation

Before discussing what it means to do semantics in DTT, it is useful to adopt some additional features for the purpose of clear and concise notation. A data type which gathers together many different components, such as a discourse context, generally necessitates a large sequence of Σ types. It is therefore convenient to adopt an alternative notation for Σ types based on the syntax of *records* (e.g. Cooper 2005, 2012). Given a sequence of Σ types:

(119) $\sum_{x_1:A} \sum_{x_2:A_2} \dots \sum_{x_{n-1}:A_{n-1}} A_n$

I shall write this as follows:

(120) $$\begin{bmatrix} \mathrm{x}_1 : A_1 \\ \mathrm{x}_2 : A_2 \\ \vdots \\ \mathrm{x}_{n-1} : A_{n-1} \\ A_n \end{bmatrix}$$

where the highest label corresponds to the outermost summand. A term of this type:

(121) $(a_1, (a_2, (\dots, (a_{n-1}, a_n)))) : \sum_{x_1:A} \sum_{x_2:A_2} \dots \sum_{x_{n-1}:A_{n-1}} A_n$

will instead be written as:

(122) $$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix} : \begin{bmatrix} \mathrm{x}_1 : A_1 \\ \mathrm{x}_2 : A_2 \\ \vdots \\ \mathrm{x}_{n-1} : A_{n-1} \\ A_n \end{bmatrix}$$

As indicated, a dependent sum type is written in square brackets whereas a term of a dependent sum types is written in parentheses. It is also convenient to define projection functions which are more suited to this 'flat' structure. Given a term a of type $\sum_{x_1:A}\sum_{x_2:A_2}\dots\sum_{x_{n-1}:A_{n-1}} A_n$, I shall use the notation $a.i$ to refer to the i-th projection. This is defined in terms of the ordinary first and second projection functions as follows:

(123) $$a.i := \begin{cases} \pi_1 \circ \underbrace{\ldots \circ \pi_2 \circ \pi_2 \circ \pi_2}_{n-1 \text{ times}}(a) & \text{if } 1 \leq i < n \\ \underbrace{\ldots \circ \pi_2 \circ \pi_2 \circ \pi_2}_{n-1 \text{ times}}(a) & \text{if } i = n \end{cases}$$

It is worth pointing out that, despite the similarity in notation, this is not intended as an implementation of record types, as in Type Theory with Records (Cooper 2005, 2012), but simply as an alternative notation for Σ types. In a Σ type, the label we choose for the summand is irrelevant: that is, $\sum_{n:Nat} Even(n)$ is the same type as $\sum_{x:Nat} Even(x)$. This is not the case for genuine record types, where two types with identical data but distinct labels are not considered equal.

Another feature which is useful from the point of view of notation is the ability to locally define one expression for use in another, larger expression. This is known as a *let expression*. I shall write

(124) $$\text{let } x := t_1 \text{ in } t_2$$

to mean 'substitute every occurrence of x in t_2 with the result of evaluating t_1'. (Note that, like a lambda abstraction, the binding $x := t_1$ is associated with a particular scope and cannot be accessed outside of that scope.) To give an example, the expression:

(125) $$\text{let } x := 2 * 5 \text{ in } x^2 + x$$

reduces to $10^2 + 10 = 110$. The motivation for let expressions is to avoid having to write out the same expression multiple times. They are defined by the following rules:

(126) $$\frac{\Gamma \vdash a : A \quad \Gamma, x : A \vdash b : B}{\Gamma \vdash (\text{let } x := a \text{ in } b) : B}\ (\text{Let Formation})$$

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A \vdash b : B}{\Gamma \vdash (\text{let } x := a \text{ in } b) \equiv b[x := a] : B}\ (\text{Let Computation})$$

In other words, let expressions simply implement the meta-theoretic concept of substitution.

When writing a term of a dependent sum type, it is frequently useful to bind the first component to a variable which can then be used in writing the second component. That is, one often needs to write something of the form:

(127) $$\text{let } x := a_1 \text{ in } \begin{pmatrix} x \\ a_2(x) \end{pmatrix}$$

This situation is so common that I shall abbreviate it by writing the binding $x \coloneqq a_1$ inside the parentheses, as follows:

$$(128) \quad \begin{pmatrix} x \coloneqq a_1 \\ a_2(x) \end{pmatrix}$$

This should be understood as simply an alternative notation for (127).

2.5 Semantics with DTT

2.5.1 The basic setup

Like Montague semantics, DTT Semantics is concerned with giving an abstract description of the truth conditions of sentences. However, its analysis of truth conditions is quite different because of the richer structure of the type system. Consider the sentence:

(129) All penguins live in Antarctica or South America.

In Montague semantics, the truth conditions of (129) are described by first translating it into a formula of STT, roughly as follows:

(130) $\forall x\,.\,\mathbf{penguin}(x) \Rightarrow \mathbf{lives\text{-}in}(\mathbf{Antarctica})(x) \vee \mathbf{lives\text{-}in}(\mathbf{South\ America})(x)$

One then considers the interpretation of (130) with respect to a model M, representing a mapping from basic expressions to things in the world. The truth conditions of the sentence are expressed in terms of the interpretation function. In DTT semantics, on the other hand, a sentence like (129) is translated as a *type*:

(131) $\prod_{x:e} \mathbf{penguin}(x) \to \mathbf{lives\text{-}in}(\mathbf{Antarctica})(x) + \mathbf{lives\text{-}in}(\mathbf{South\ America})(x)$

Having made this translation, the truth conditions are completely fixed, *without reference to a further interpretation*: the sentence is true iff there exists a term of type (131), and false otherwise. In general terms, the difference between DTT Semantics and Montague Semantics is that, instead of the type system functioning as an intermediary for relating natural language and set theory, it becomes the end result of the semantic derivation. Following Luo (2010), I use semantic brackets to indicate the mapping from natural language to DTT, and refer to this as 'interpretation':

(132) $$\text{natural language} \xrightarrow{\quad [\![\,\cdot\,]\!] \quad} \text{dependent type theory}$$

Notice that, in this approach, we loose the distinction between translation and interpretation, because DTT subsumes the role of both logic and set theory.

One might worry that by not invoking a model in which the type system is interpreted, we are inappropriately fixing the truth of sentences. The job of semantics is not to specify which sentences are true or false, but to give a general description of the *conditions* under which sentences are true or false. In Montague semantics, the use of an *arbitrary* model reflects the fact that neither speakers nor semanticists have complete knowledge about the state of the world: one does not have access to the set of all penguins and their locations, for example. In DTT Semantics, the locus of this kind of uncertainty is the background context Γ. A proposition like (130) is not judged as true or false simpliciter, but true or false in a particular context. Some contexts entail (130), others entail $\neg$(130), and others do not entail either (130) or $\neg$(130). Because interpretation does not specify the background context in which a proposition is asserted, it does not inappropriately fix the truth of sentences.

DTT Semantics is sometimes described as proof-theoretic, in contrast with model-theoretic approaches. 'Proof-theoretic' refers to the idea that the meaning of a proposition or logical connective should be described in terms of its overall inferential role within the system of inference it inhabits (Dummett et al. 1991). For example, from a proof-theoretic point of view, the meaning of the logical connective 'and' is given by its introduction and elimination rules, rather than by its set-theoretic interpretation. The semantics of DTT itself can be viewed as proof-theoretic since each of its constructions is exhaustively described in terms of inference rules without dependence on a set-theoretic background. However, as Luo (2014) has pointed out, when DTT is used for the purpose of natural language semantics, it plays a role analogous to set theory, representing semantic entities such as individuals, properties, sets, situations, and so on. The correct statement, therefore, is that the semantics *of* DTT is proof-theoretic, whereas natural language semantics *with* DTT is model-theoretic. I use the term 'DTT Semantics' to refer to the latter.

2.5.2 Common nouns

I now turn to the question of how different sorts of semantic objects should be typed in DTT. A fundamental distinction in DTT Semantics is between approaches in which common nouns are represented by predicates (e.g. Krahmer & Piwek 1999, Bekki 2014, Tanaka et al. 2017), and those in which common nouns are represented by types (e.g. Ranta 1994, Luo 2012, Chatzikyriakidis & Luo 2017*b*, 2020). In the former approach, a noun like *dog* is interpreted as a function $Ind \to Prop$ (the analog of Montague's $e \to t$) or alternatively $Ind \to Type$; the sentence *Rover is a dog* is interpreted as $Dog(Rover)$; and the type of other semantic elements such as adjectives, verbs, quantifiers, and so on, closely resembles their Montague type. In the alternative approach, *dog* is interpreted as a type Dog; the sentence *Rover is a dog* becomes the typing judgement $Rover : Dog$ (or a proposition encoding this judgement); and the types of adjectives, verbs, quantifiers and other elements must be revised accordingly.

In this thesis, I shall adopt the more conservative position and interpret nouns as predicates. Nevertheless, most of the central ideas of the thesis – including implication networks and the automatic preservation of implication by composition – are neutral with regard to this question and could equally be developed within a 'Nouns as Types' framework. One could argue that treating nouns as types is closer to the spirit of DTT. Just as in mathematics, where a nuanced type theory does away with general concepts like 'number' in favour of specific types like Nat, Int, $Bool$, $Real$, and so on, so the Nouns as Types approach abolishes the indiscriminate type of 'entities', replacing it with functionally meaningful distinctions like $Person$, $Animal$, $Institution$, $Information$, etc. Moreover, a powerful feature of the Nouns as Types approach is that it allows each noun to specify its own identity criteria. Consider the following two sentences, discussed by Chatzikyriakidis & Luo (2017*b*):

(133) a. EasyJey has transported 1 million people in 2010.
b. EasyJet has transported 1 million passengers in 2010.

(a) and (b) are not equivalent because where one and the same $Person$ can count as two different instances of $Passenger$ on different journeys. In the Nouns as Types framework, this is explained by the different identity criteria associated with $Person$ and $Passenger$.

However, the same features of the Nouns as Types approach which make it powerful can also be difficult to work with in a decompositional setting. For example, suppose one wants to give a decompositional analysis of a noun like *dog*. In the Nouns as Types framework, this means defining a type *Dog*, together with its identity conditions, i.e. what counts as a proof of $x =_{Dog} y$. Presumably, *Dog* is not a primitive type, but built from a complicated product of domains like shape, size, colour, behaviour, relationship with humans, and so on. However, most of this information is not relevant to the identity conditions: a dog's shape, size, colour, behaviour, and so on, can change throughout its life without it becoming a different dog. It is very difficult to formulate appropriate identity conditions for dogs, or indeed any other category, on the basis of a semantic decomposition. One is left with the statement that two dogs are the same iff they are the same 'underlying individual', but this assumes a general type of individuals, about which 'dogness' is predicated, which is essentially equivalent to the nouns as predicates approach.

My assumption in this thesis will be that a common noun like *dog* is typed as

(134) $[\![\text{dog}]\!] : \mathit{Ind} \to \mathit{Type}$

that is, a function from individuals to types[1]. The reason for typing nouns as $\mathit{Ind} \to \mathit{Type}$ rather than the more obvious choice $\mathit{Ind} \to \mathit{Prop}$ is that nouns carry information: for instance, each dog has a certain shape, size, colour, behaviour, etc. The information provided by nouns must be accessed by adjectives and verbs: a *large dog* is a dog whose size is big, a *black dog* is a dog whose colour is black, and so on. It follows from this that terms of $[\![\text{dog}]\!](x)$, representing all the different ways that x can be a dog, should not be collapsed into a single proposition. Note, however, that although a proof of $[\![\text{dog}]\!](x)$ carries a great deal of information, this information is not relevant to the identity conditions for x, which are instead determined by *Ind*. Given the noun meaning $[\![\text{dog}]\!] : \mathit{Ind} \to \mathit{Type}$, we can use propositional truncation to derive the type of all dogs, as follows:

(135) $$\mathit{Dog} := \begin{bmatrix} \text{x} : \mathit{Ind} \\ |[\![\text{dog}]\!](\text{x})| \end{bmatrix}$$

As shown, an element of this type consists of an individual x, together with a proof that $[\![\text{dog}]\!](x)$ is inhabited. The identity conditions for this type are determined by *Ind*, because all elements of $|[\![\text{dog}]\!](x)|$ are identical. Notice that in this approach, the type of all dogs is derived from the noun meaning $[\![\text{dog}]\!]$, rather than these being the same thing as in Chatzikyriakidis & Luo (2017*b*).

[1]Henceforth, I use *Ind* instead of e for the type of individuals, this being a more common notation in DTT Semantics.

2.5.3 Sentences

If types are not used to represent common noun meanings, then how should we think of them? I shall follow the interpretation of types in authors such as Cooper (2012) and Tanaka et al. (2017), where they are used to represent *discourse contexts*. Contexts are already present on the meta-theoretic level, where they appear on the left hand side of judgements, and there is a straightforward way to take any context and represent it internally as a type. Given any context of the form:

(136) $x_1 : A_1,\ x_2 : A_2(x_1),\ \ldots,\ x_n : A_n(x_1, x_2, \ldots, x_{n-1})$

this is represented by the following dependent sum type:

(137) $$\begin{bmatrix} x_1 : A_1 \\ x_2 : A_2(x_1) \\ \ldots \\ x_n : A_n(x_1, x_2, \ldots, x_{n-1}) \end{bmatrix}$$

where the position of a variable in the dependent sum corresponds to its position in the context. Like contexts, dependent types can also be 'internalized' where they become functions into a type universe. Given a dependent type:

(138) $x_1 : A_1,\ x_2 : A_2(x_1),\ \ldots,\ x_n : A_n(x_1, x_2, \ldots, x_{n-1})$
$\vdash B(x_1, x_2, \ldots, x_n) : \mathit{Type}$

this is internalized as a function:

(139) $$B : \begin{bmatrix} x_1 : A_1 \\ x_2 : A_2(x_1) \\ \ldots \\ x_n : A_n(x_1, x_2, \ldots, x_{n-1}) \end{bmatrix} \to \mathit{Type}$$

sending $(x_1, x_2, \ldots, x_n) \mapsto B(x_1, x_2, \ldots, x_n)$. We can think of the types A_1, $A_2(x_1)$, $\ldots$, $A_n(x_1, x_2, \ldots, x_{n-1})$ as *presuppositions* which must be satisfied in order for the type $B(x_1, x_2, \ldots, x_n)$ to be formed.

The representation of discourse contexts as types leads naturally to a type-theoretic implementation of dynamic semantics in which the meaning of a sentence is represented as an instruction for updating a context with new information. The presuppositions of a sentence appear as conditions which the context must satisfy prior to the update, whereas the assertion is the information added by the update itself. I shall represent a context update by an element of the type:

(140) $$\mathit{Update} := \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \mathit{Prop} \end{bmatrix}$$

Terms of this type are pairs (P, A), where P is a type encoding the presuppositions of the update, and A is a proposition dependent on P representing the assertion. The dependence of A on P ensures that the assertion can only be formed provided its propositions are satisfied. Note that the presupposition type is a context (an element of *Type*) because there can be many different situations which satisfy it, whereas the assertion is a proposition (an element of *Prop*), because it is a simple statement which is either true or false[1]. *Update* is similar to the type used to represent context updates in some versions of Type Theory with Records (e.g. Larsson 2015, Cooper 2016); it also resembles what is called a ‘preliminary representation’ in Discourse Representation Theory (Kamp et al. 2011).

To give an example of a context update, consider the sentence *John quit smoking*. Without going into the details of how the meaning of this sentence is derived compositionally, we might interpret it as follows:

(141) $[\![\text{John quit smoking}]\!] : \mathit{Update}$

$$[\![\text{John quit smoking}]\!] = \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{t} : \mathit{Time} \\ \mathrm{s} : \mathit{State} \\ \text{smoker(s)} \\ \text{theme}(\mathrm{s}, \mathit{John}) \\ \text{holds-at}(\mathrm{t}, \mathrm{s}) \\ \text{past(t)} \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \neg\text{holds-at}(\mathit{now}, \mathrm{p}.2) \end{pmatrix}$$

In other words, *John quit smoking* presupposes that there is some time in the past at which the state of John smoking held, and asserts that it no longer holds at the present time. If there is no time in the past at which John was a smoker, then the assertion that he quit smoking cannot be formed.

One advantage of explicitly separating presupposed information from asserted information is that it allows us to write functions which manipulate presuppositions and assertions differently. For example, it is a well-known observation that presuppositions project out of negative contexts, whereas assertions do not: *John quit smoking* and *John didn’t quit smoking* have identical presuppositions, despite their opposite assertions. This motivates the following analysis of the negative particle *not*:

(142) $[\![\text{not}]\!] : \mathit{Update} \to \mathit{Update}$

$$[\![\text{not}]\!] \coloneqq \lambda \mathrm{U} : \mathit{Update}\ .\ \begin{pmatrix} \mathrm{U}.1 \\ \lambda \mathrm{p} : \mathrm{U}.1\ .\ \neg \mathrm{U}.2(\mathrm{p}) \end{pmatrix}$$

As shown, *not* is a modifier of sentence meanings which takes a context update and negates its assertion, leaving its presuppositions untouched. This has the intended

[1]Strictly speaking, the type *Update* ought to be indexed to a particular type universe:

$$\mathit{Update}_i \coloneqq \begin{bmatrix} \mathrm{P} : \mathit{Type}_i \\ \mathrm{P} \to \mathit{Prop}_i \end{bmatrix}$$

because the presupposition type may contain types belonging to universes higher than Type_0. I use the expression *Update* polymorphically to mean Update_i for any i.

result that presuppositions project out of negated contexts. Other facts relating to presupposition projection can be explained in a similar way.

Given a current context $C_t : \mathit{Type}$, and an update $U : \mathit{Update}$, one can try to use U to update C_t with new information. The first step in the update procedure is to find a function $s : C_t \to U.1$, which shows how the current context satisfies the presuppositions of the update. Assuming such a function can be found, one then forms the updated context as follows:

(143) $$C_{t+1} \coloneqq \begin{bmatrix} \mathrm{c} : C_t \\ U.2(s(\mathrm{c})) \end{bmatrix}$$

The updated context contains both the original context C_t together with the assertion of U. The presuppositions of U which are needed to derive the assertion are supplied by means of the function s. In general, there might be several different choices of s, leading to multiple potential new contexts. If no supply function can be found, then one can attempt to accommodate the information present in U's presupposition type which cannot be supplied by the context. The context update procedure is discussed in greater detail in Section 4.7.

2.5.4 Intersective adjectives

An intersective adjective like *red* or *10cm tall* is the kind of thing which takes an individual and returns a sentence meaning. It is therefore an element of the type:

(144) $\mathit{Ind} \to \mathit{Update}$

The selectional restrictions of the adjective appear as the presuppositions of the resulting update. For example, the adjective *red* would be interpreted roughly as follows:

(145) $[\![\mathrm{red}]\!] : \mathit{Ind} \to \mathit{Update}$

$$[\![\mathrm{red}]\!] \coloneqq \lambda \mathrm{x} : \mathit{Ind} \ . \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{c} : \mathit{Colour} \\ \text{has-colour}(\mathrm{x}, \mathrm{c}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \ \mathrm{red}'(\mathrm{p}.1) \end{pmatrix}$$

where Colour is some type of colour values, and $\mathit{red}' : \mathit{Colour} \to \mathit{Prop}$ is a predicate of colour values. In other words, $[\![\mathrm{red}]\!](\mathrm{x})$ presupposes that x has a certain colour value, and asserts that this colour value can be described as red. If $[\![\mathrm{red}]\!]$ is applied to an individual which is incapable of having a colour, then the resulting presupposition type will be empty and the update infelicitous.

The treatment of selectional restrictions as presuppositions contrasts with approaches where they are treated as domain restrictions on predicates. For example, in the Nouns as Types approach of Chatzikyriakidis & Luo (2017*b*), intersective adjectives have type $D \to \mathit{Prop}$, where D is the domain to which the adjective can apply. Hence, $[\![\mathrm{happy}]\!]$ would be typed as $\mathit{Person} \to \mathit{Prop}$, $[\![\mathrm{heavy}]\!]$ as $\mathit{Physical} \to \mathit{Prop}$, $[\![\mathrm{interesting}]\!]$ as $\mathit{Information} \to \mathit{Prop}$, and so on. One disadvantage of this approach is that there is no overall type for intersective adjectives: different adjectives have different types reflecting their different selectional restrictions. In order to write a function which applies to arbitrary intersective adjectives, one needs to introduce a special type universe containing all the types which can function as selectional restrictions (Chatzikyriakidis & Luo call this CN); one then writes functions which are polymorphic over this universe. In my approach, this solution is unnecessary because all intersective adjectives belong to a unified type, namely $Ind \to Update$.

Unlike in Montague semantics, I do not treat common nouns and intersective adjectives as having the same type. This reflects their distinct syntactic behaviour: for example, a quantificational determiner can apply to a noun (*every boy*) but not to an adjective (**every red*). Moreover, nouns do not have selectional restrictions in the same way as adjectives: to say that something is *red* is to presuppose that it has a colour, whereas to say that something is a *dog* carries no presuppositions. Accordingly, nouns are functions from individuals to *contexts* (elements of $Ind \to Type$) whereas adjectives are functions from individuals to context *updates* (elements of $Ind \to Update$). This correlates with the distinction in philosophy between sortal and non-sortal predicates.

Intersective adjective + noun composition is exactly analogous to the process of updating a context with new information. It can be viewed as a kind of 'local' context update in which the noun provides the background context and the adjective provides the update. The result is an updated noun, that is a new element of $Ind \to Type$. To illustrate how this works, consider the adjective *red*, interpreted as shown in (145), and the noun *apple*, which for the sake of example I shall represent as follows:

(146) $[\![\text{apple}]\!] : Ind \to Type$

$$[\![\text{apple}]\!] \coloneqq \lambda \text{x} : Ind \,.\, \begin{bmatrix} \text{c} : Colour \\ \text{has-colour}(\text{x}, \text{c}) \\ \text{apple-colour}(\text{c}) \\ \text{s} : Shape \\ \text{has-shape}(\text{x}, \text{s}) \\ \text{apple-shape}(\text{s}) \\ \text{t} : Taste \\ \text{has-taste}(\text{x}, \text{t}) \\ \text{apple-taste}(\text{t}) \\ \dots \end{bmatrix}$$

where $Colour$, $Shape$, $Taste$ are perceptual spaces representing possible colours, shapes and tastes respectively. In order for *red* to combine with *apple*, it must be possible to use the information provided by *apple* to supply the presuppositions of *red*. In other words, there must be a function:

(147) $s : \prod_{\text{x}:Ind} [\![\text{apple}]\!](\text{x}) \to [\![\text{red}]\!](\text{x}).1$

Given our assumptions, this is implemented as follows:

$$(148)\quad s := \lambda \mathrm{x} : \mathit{Ind} \ . \ \lambda \mathrm{p} : \begin{bmatrix} \mathrm{c} : \mathit{Colour} \\ \text{has-colour}(\mathrm{x}, \mathrm{c}) \\ \text{apple-colour}(\mathrm{c}) \\ \mathrm{s} : \mathit{Shape} \\ \text{has-shape}(\mathrm{x}, \mathrm{s}) \\ \text{apple-shape}(\mathrm{s}) \\ \mathrm{t} : \mathit{Taste} \\ \text{has-taste}(\mathrm{x}, \mathrm{t}) \\ \text{apple-taste}(\mathrm{t}) \\ \ldots \end{bmatrix} . \begin{pmatrix} \mathrm{p.1} \\ \mathrm{p.2} \end{pmatrix}$$

As shown, s simply projects out the first and second components of $[\![\text{apple}]\!](\mathrm{x})$. Given this supply function, the combination $[\![\text{red apple}]\!]$ is defined as follows:

(149) $[\![\text{red apple}]\!] : \mathit{Ind} \to \mathit{Type}$

$$[\![\text{red apple}]\!] = \lambda \mathrm{x} : \mathit{Ind} \ . \begin{bmatrix} \mathrm{c} : [\![\text{apple}]\!](\mathrm{x}) \\ [\![\text{red}]\!](\mathrm{x}).2(s(\mathrm{c})) \end{bmatrix}$$

$$= \lambda \mathrm{x} : \mathit{Ind} \ . \begin{bmatrix} \mathrm{c} : \begin{bmatrix} \mathrm{c} : \mathit{Colour} \\ \text{has-colour}(\mathrm{x}, \mathrm{c}) \\ \text{apple-colour}(\mathrm{c}) \\ \mathrm{s} : \mathit{Shape} \\ \text{has-shape}(\mathrm{x}, \mathrm{s}) \\ \text{apple-shape}(\mathrm{s}) \\ \mathrm{t} : \mathit{Taste} \\ \text{has-taste}(\mathrm{x}, \mathrm{t}) \\ \text{apple-taste}(\mathrm{t}) \\ \ldots \end{bmatrix} \\ \text{red}'(\mathrm{c.1}) \end{bmatrix}$$

The first component of the combination contains all of the information associated with the noun *apple* and the second component contains the assertion of *red*. The presuppositions associated with *red* are supplied by applying the function s to the first component.

To generalize from this example, given an arbitrary noun meaning $N : \mathit{Ind} \to \mathit{Type}$ and intersective adjective meaning $A : \mathit{Ind} \to \mathit{Update}$, the combination is given by:

$$(150)\quad AN := \lambda \mathrm{x} : \mathit{Ind} \ . \ \mathrm{Update}(N(\mathrm{x}), A(\mathrm{x}))$$

where 'Update' refers to the context update procedure described in the previous section. Implicit in this definition is the assumption that adjective + noun composition involves a process of *proof discovery*. In general, we cannot assume that the presuppositions required for the adjective are 'just sitting there' in the noun, since adjectives can have very specific and idiosyncratic restrictions. For example, the adjectives *thick* and *thin* have unique selectional restrictions, requiring a physical object which can be schematized as either a cylinder (*thick stick*) or a surface (*thick slab*). The possibility of combining with *thick/thin* cannot be immediately ascertained from a noun's lexical representation. Rather, one must first discover an appropriate 'schematization' which shows how an entity of that type can be conceptualized as a cylinder or a surface.

The analysis of selectional restrictions as involving active proof discovery contrasts with approaches where it is handled automatically by means of subtyping (e.g. Luo 2012, Chatzikyriakidis & Luo 2017*b*). In the subtyping approach, the fact that *red apple* is a valid combination is a consequence of the fact that $[\![\text{apple}]\!]$ is a subtype of the domain type of $[\![\text{red}]\!]$ (i.e. the type of coloured objects). This analysis predicts that a noun should always be compatible with an adjective in a unique way, since a subtype is included in a supertype by means of a unique coercion. However, this does not appear to be the case: for example, the expression *silver tree* might refer either to the colour of the tree's leaves, or to the colour of its bark, suggesting the presence of two distinct coercions. This phenomenon is very common in spatial adjectives: *thick spoon* might refer to the thickness of a spoon's handle, the thickness of its bowl, or some combination of the two. In the context update approach to selectional restrictions, the possibility of a noun satisfying the presuppositions of an adjective in more than one way is automatic.

As in discourse-level processing, adjective + noun composition can involve accommodation. For example, consider the combination *tall stick*. The adjective *tall* requires an axis which is either vertical in the environment or can be somehow conceptualized as vertical (e.g. in the object's canonical orientation). Neither of these conditions is provided by *stick*, which does not contain orientation information in its lexical entry. However, it is possible to modify the meaning of *stick* by adding the concept of verticality without inconsistency, leading to the concept of a stick which is pointing upwards, or which is canonically pointing upwards. The modified noun meaning can then be felicitously combined with the adjective *tall* in the normal way.

2.5.5 Determiners

This thesis will not have a great deal to say about determiners; nevertheless I include them here for the sake of completeness. In Montague semantics, the standard type of quantificational determiners like *all*, *some*, *most*, *exactly four*, etc., is $(e \to t) \to (e \to t) \to t$, that is a relation between sets of entities (Montague 1973, Barwise & Cooper 1981). Quantified noun phrases like *every student* or *most apples* then become generalized quantifiers, that is elements of $(e \to t) \to t$. In DTT Semantics, the appropriate type for quantifiers depends on one's assumptions about how nouns and properties are typed. Given that we are representing nouns as elements of $Ind \to Type$ and properties as elements of $Ind \to Update$, the appropriate type for a quantificational determiner is:

(151) $(Ind \to Type) \to (Ind \to Update) \to Update$

that is, a function from nouns and properties to sentence meanings. A quantified noun phrase like *all students* will therefore have type $(Ind \to Update) \to Update$.

To give an example, the determiner *all* can be implemented as follows:

(152) $[\![\text{all}]\!] : (Ind \to Type) \to (Ind \to Update) \to Update$

$$[\![\text{all}]\!] \coloneqq \lambda \mathrm{N} : Ind \to Type\ .\ \lambda \mathrm{P} : Ind \to Update\ .\ \begin{pmatrix} \mathrm{C} \coloneqq \prod_{\mathrm{x}:Ind} \mathrm{N(x)} \to \mathrm{P(x).1} \\ \lambda \mathrm{c} : \mathrm{C}\ .\ \prod_{\mathrm{x}:Ind} \prod_{\mathrm{n:N(x)}} \mathrm{P(x).2(c(x)(n))} \end{pmatrix}$$

Notice how *all* involves two universal propositions: one as a presupposition and one as an assertion. The presupposition is that every individual which satisfies the noun argument also satisfies the presuppositions of the property argument. This is needed to explain the difference between:

(153) a. All apples are red.
b. ? All meetings are red.

Sentence (a) is felicitous, albeit false, because the presupposition that every apple has a colour is satisfied, whereas sentence (b) is infelicitous since (in most contexts) it is not clear what it means for a meeting to have a colour.

Determiners like *all*, *some* and *no*, which do not make explicit reference to the size of collections, can be easily implemented in DTT. Things become more complicated with determiners like *most*, *at least 2*, *exactly 12*, and so forth, which require explicit reference to cardinality. The ease with which these are formalized in Montague semantics has to do with its set-theoretic background. In DTT, one does not know ahead of time what all the elements of a type are: they have to be explicitly constructed. The only way to know that a type has a certain cardinality is by constructing a bijection between it and a type of known cardinality. For instance, one knows that a type has a cardinality of 3 if one can construct a bijection between it and a special three-element type **3**. For a formalization of numerosity-dependent generalized quantifiers in terms of bijections, see Tanaka (2014).

When it comes to articles such as *the* and *a/an*, I shall assume that these can also be treated as quantificational determiners. I adopt an approach to the definite article in which existence appears as a presupposition:

(154) $[\![\text{the}]\!] : (Ind \to Type) \to (Ind \to Update) \to Update$

$$[\![\text{the}]\!] \coloneqq \lambda \mathrm{N} : Ind \to Type\ .\ \lambda \mathrm{P} : Ind \to Update\ .\ \begin{pmatrix} \mathrm{C} \coloneqq \begin{bmatrix} \mathrm{x} : Ind \\ \mathrm{N(x)} \\ \mathrm{P(x).1} \end{bmatrix} \\ \lambda \mathrm{c} : \mathrm{C}\ .\ \mathrm{P(c.1).2(c.3)} \end{pmatrix}$$

As shown, $[\![\text{the}]\!]$ presupposes (i) the existence of an individual x, (ii) that x satisfies the noun argument N, and (iii) that x satisfies the presupposition of the property argument P. Given these presuppositions, the assertion is that x satisfies the assertion of P[1].

Table 2.4 summarizes the correspondence between semantic categories and types which I shall assume for the remainder of the thesis. In the next chapter, this table will be updated to include positive and negative gradable adjectives. In making these choices, I have been guided by the principle, carried over from Montague semantics, that semantic types should reflect syntactic behaviour. Hence syntactically relevant categories like 'intersective adjective' and 'quantificational determiner' correspond to unified types. Nuances such as the different selectional restrictions of intersective adjectives or the distinct presuppositions of different determiners are not reflected on the type level, but are packaged into the presuppositional information carried by context updates. Nevertheless, much of what I have to say in this thesis is not affected by these specific decisions, and could be translated into any framework which relies on DTT as a modelling language.

Semantic Category	Type
Context	$Type$
Assertion	$Prop$
Common noun	$Ind \to Type$
Sentence	$Update := \begin{bmatrix} \mathrm{P} : Type \\ \mathrm{P} \to Prop \end{bmatrix}$
Sentential modifier	$Update \to Update$
Intersective adjective	$Ind \to Update$
Quantificational determiner	$(Ind \to Type) \to (Ind \to Update) \to Update$
Generalized quantifier	$(Ind \to Update) \to Update$

Table 2.4: Some semantic categories and their corresponding types.

[1]Note that I have opted not to include uniqueness in the presuppositions of $[\![\text{the}]\!]$, although this could easily be done. Following authors such as Ludlow & Segal (2004), I assume that the unacceptability of *the* in contexts where more than one individual satisfies P is a matter of ambiguity rather than presupposition failure.

2.6 Lexical networks with DTT

2.6.1 Implementing basic networks

We are now in a position to begin developing a compositional theory of lexical networks in DTT. As discussed in Chapter 1, lexical networks are motivated by a basic observation in lexical semantics, which is that most words cannot be given a precise definition in terms of necessary and jointly sufficient conditions. Rather, natural categories consist of clusters of conditions in which no one condition is necessary and various different combinations can be sufficient (Rosch 1975). This same organisation is seen across many different semantic categories, including nouns, adjectives, verbs and prepositions. As we saw in Chapter 1, the structure of a cluster concept can be represented by an implicational network in which nodes correspond to senses and arrows correspond to implication.

Unlike STT, DTT permits an implementation of implicational networks in which the arrows are represented directly as terms. To illustrate this, recall Norvig's (1989) *meat* network, discussed in Section 1.2.2. According to Norvig, *meat* permits the following senses:

(155)
1. The edible muscle tissue of a mammal, intended for consumption.
2. Allow fowl as well as mammals.
3. Allow fish as well as fowl and mammals.
4. Allow organ meat as well as muscle tissue.
5. Allow skin as well as muscle tissue.
6. The interior edible part of any food (e.g. *coconut meat*)
7. (metaphor) The core or essence of something.

I also proposed the additional sense:

8. Any edible part, interior or otherwise, of any animal.

These eight senses are partially ordered by implication into the following network:

(156)
$$\begin{tikzcd}
& \llbracket \text{meat} \rrbracket_8 & & & \\
\llbracket \text{meat} \rrbracket_3 \arrow[ur] & & & & \\
\llbracket \text{meat} \rrbracket_2 \arrow[u] & \llbracket \text{meat} \rrbracket_4 \arrow[uu] & \llbracket \text{meat} \rrbracket_5 \arrow[uul] & \llbracket \text{meat} \rrbracket_6 & \llbracket \text{meat} \rrbracket_7 \\
& & \llbracket \text{meat} \rrbracket_1 \arrow[ull] \arrow[ul] \arrow[u] \arrow[ur] \arrow[urr] & &
\end{tikzcd}$$

As discussed, the *meet* of two senses represents a common prototype and the *join* of two senses represents a common generalization.

Let us begin by formalizing the senses. All the senses in the network are of type $Ind \to Type$, since they are all noun meanings. The most prototypical sense, $[\![\text{meat}]\!]_1$ could be implemented as follows:

(157) $$[\![\text{meat}]\!]_1 \coloneqq \lambda \text{x} : Ind \;.\; \begin{bmatrix} \text{edible(x)} \\ \text{from-mammal(x)} \\ \text{muscle-tissue(x)} \end{bmatrix}$$

where the predicates edible(x), from-mammal(x) and muscle-tissue(x) should be understood as shorthand for a collection of further conditions. Senses 2-8 are then formulated as weakened versions of $[\![\text{meat}]\!]_1$, as follows:

(158) $$[\![\text{meat}]\!]_2 \coloneqq \lambda \text{x} : Ind \;.\; \begin{bmatrix} \text{edible(x)} \\ \text{from-mammal(x)} + \text{from-fowl(x)} \\ \text{muscle-tissue(x)} \end{bmatrix}$$

$$[\![\text{meat}]\!]_3 \coloneqq \lambda \text{x} : Ind \;.\; \begin{bmatrix} \text{edible(x)} \\ \text{(from-mammal(x)} + \text{from-fowl(x))} + \text{from-fish(x)} \\ \text{muscle-tissue(x)} \end{bmatrix}$$

$$[\![\text{meat}]\!]_4 \coloneqq \lambda \text{x} : Ind \;.\; \begin{bmatrix} \text{edible(x)} \\ \text{from-mammal(x)} \\ \text{muscle-tissue(x)} + \text{organs(x)} \end{bmatrix}$$

$$[\![\text{meat}]\!]_5 \coloneqq \lambda \text{x} : Ind \;.\; \begin{bmatrix} \text{edible(x)} \\ \text{from-mammal(x)} \\ \text{(muscle-tissue(x)} + \text{organs(x))} + \text{skin(x)} \end{bmatrix}$$

$$[\![\text{meat}]\!]_6 \coloneqq \lambda \text{x} : Ind \;.\; \begin{bmatrix} \text{edible(x)} \\ \text{interior-part(x)} \end{bmatrix}$$

$$[\![\text{meat}]\!]_7 \coloneqq \lambda \text{x} : Ind \;.\; \begin{bmatrix} \text{interior-part(x)} + \text{essential-part(x)} \end{bmatrix}$$

$$[\![\text{meat}]\!]_8 \coloneqq \lambda \text{x} : Ind \;.\; \begin{bmatrix} \text{edible(x)} \\ \text{from-animal(x)} \end{bmatrix}$$

Having described the senses, we can now turn to the arrows, which are formalized as dependent functions. For example, consider the arrow from $[\![\text{meat}]\!]_1$ to $[\![\text{meat}]\!]_2$, which I shall call $[\![\text{meat}]\!]_1^2$. This should take a proof that x is *meat* in the sense of being the muscle tissue of a mammal, and show that it belongs to the broader sense of *meat* which includes fowl. Its type is:

(159) $$[\![\text{meat}]\!]_1^2 : \textstyle\prod_{\text{x}:Ind} [\![\text{meat}]\!]_1(\text{x}) \to [\![\text{meat}]\!]_2(\text{x})$$

It is implemented by injecting the proof of from-mammal(x) into the left-hand side of the sum type from-mammal(x) + from-fowl(x), as follows:

(160) $$[\![\text{meat}]\!]_1^2 \coloneqq \lambda \text{x} : \mathit{Ind} \ . \ \lambda \text{t} : \begin{bmatrix} \text{edible(x)} \\ \text{from-mammal(x)} \\ \text{muscle-tissue(x)} \end{bmatrix} . \begin{pmatrix} \text{t.1} \\ \text{inl(t.2)} \\ \text{t.3} \end{pmatrix}$$

Notice that all the components except the second component are simply copied. Most of the other arrows in the network are implemented in a similar way, by dropping components and/or injecting components into sum types. Some of the other arrows are slighly more sophisticated: for instance, the arrow from $[\![\text{meat}]\!]_1$ to $[\![\text{meat}]\!]_8$ requires going from a proof of muscle-tissue(x) to a proof of interior-part(x).

Given two consecutive arrows in a lexical network, we can combine them to get a composite arrow. For example, consider the arrow $[\![\text{meat}]\!]_2^3$ from $[\![\text{meat}]\!]_2$ to $[\![\text{meat}]\!]_3$, which is given by:

(161) $$[\![\text{meat}]\!]_2^3 : \textstyle\prod_{\text{x}:\mathit{Ind}} [\![\text{meat}]\!]_2(\text{x}) \to [\![\text{meat}]\!]_3(\text{x})$$

$$[\![\text{meat}]\!]_2^3 \coloneqq \lambda \text{x} : \mathit{Ind} \ . \ \lambda \text{t} : \begin{bmatrix} \text{edible(x)} \\ \text{from-mammal(x)} + \text{from-fowl(x)} \\ \text{muscle-tissue(x)} \end{bmatrix} . \begin{pmatrix} \text{t.1} \\ \text{inl(t.2)} \\ \text{t.3} \end{pmatrix}$$

We can compose the arrows $[\![\text{meat}]\!]_1^2$ and $[\![\text{meat}]\!]_2^3$ to get an arrow $[\![\text{meat}]\!]_1^3$ which goes from $[\![\text{meat}]\!]_1$ to $[\![\text{meat}]\!]_3$ in a single step:

(162) $$[\![\text{meat}]\!]_1^3 : \textstyle\prod_{\text{x}:\mathit{Ind}} [\![\text{meat}]\!]_1(\text{x}) \to [\![\text{meat}]\!]_3(\text{x})$$

$$\begin{aligned} [\![\text{meat}]\!]_1^3 &\coloneqq \lambda \text{x} : \mathit{Ind} \ . \ [\![\text{meat}]\!]_2^3(\text{x}) \circ [\![\text{meat}]\!]_1^2(\text{x}) \\ &= \lambda \text{x} : \mathit{Ind} \ . \ \lambda \text{t} : \begin{bmatrix} \text{edible(x)} \\ \text{from-mammal(x)} \\ \text{muscle-tissue(x)} \end{bmatrix} . \begin{pmatrix} \text{t.1} \\ \text{inl(inl(t.2))} \\ \text{t.3} \end{pmatrix} \end{aligned}$$

where $\circ$ is ordinary function composition. As shown $[\![\text{meat}]\!]_1^3$ takes a proof of $[\![\text{meat}]\!]_1(\text{x})$ and performs $[\![\text{meat}]\!]_1^2(\text{x})$ followed by $[\![\text{meat}]\!]_2^3(\text{x})$.

Recall that the arrows in an implicational networks should form a partial order, that is they should be reflexive, antisymmetric and transitive. The antisymmetry of the *meat* network is apparent from its structure. Transitivity is given by the composition of arrows which I have just described. The remaining property needed to ensure that we have a partial order is reflexivity. Given a network diagram like (156), each sense should be understood as having a unique implicit arrow to itself, known as the identity arrow. The identity arrow is a function which 'does nothing'. For example, the identity arrow at $[\![\text{meat}]\!]_1$ is given by:

(163) $$[\![\text{meat}]\!]_1^1 : \textstyle\prod_{\text{x}:\mathit{Ind}} [\![\text{meat}]\!]_1(\text{x}) \to [\![\text{meat}]\!]_1(\text{x})$$

$$[\![\text{meat}]\!]_1^1 \coloneqq \lambda \text{x} : \mathit{Ind} \ . \ \lambda \text{t} : \begin{bmatrix} \text{edible(x)} \\ \text{from-mammal(x)} \\ \text{muscle-tissue(x)} \end{bmatrix} . \ \text{t}$$

Note that composing $[\![\text{meat}]\!]_1^1$ with any other arrow yields that arrow back, in other words $[\![\text{meat}]\!]_1^1$ acts as the identity with respect to composition. The presence of identity arrows is what ensures the reflexivity property. To summarize, given a common noun N, an *interpretation network for N* is given by:

(164)
- A set of senses $\{[\![\mathrm{N}]\!]_i : \mathit{Ind} \to \mathit{Type}\}$
- A set of arrows $\{[\![\mathrm{N}]\!]_i^j : \prod_{\mathrm{x}:\mathit{Ind}} [\![\mathrm{N}]\!]_i(\mathrm{x}) \to [\![\mathrm{N}]\!]_j(\mathrm{x})\}$

subject to the following conditions for all i, j, k:

- There is at most one arrow $[\![\mathrm{N}]\!]_i^j$
- $[\![\mathrm{N}]\!]_i^i = \lambda \mathrm{x} : \mathit{Ind} \ . \ \lambda \mathrm{n} : [\![\mathrm{N}]\!]_i \ . \ \mathrm{n}$
- Given arrows $[\![\mathrm{N}]\!]_i^j$ and $[\![\mathrm{N}]\!]_j^i$ then $i = j$
- Given arrows $[\![\mathrm{N}]\!]_i^j$ and $[\![\mathrm{N}]\!]_j^k$ then $[\![\mathrm{N}]\!]_i^k = \lambda \mathrm{x} : \mathit{Ind} \ . \ [\![\mathrm{N}]\!]_j^k(\mathrm{x}) \circ [\![\mathrm{N}]\!]_i^j(\mathrm{x})$

The conditions ensure that the network is a partial order.

I assume that the interpretation network for a noun – including both senses and arrows – is part of its lexical entry. One could imagine an alternate account in which the lexical entry consists of simply a list of senses $[\![\mathrm{N}]\!]_1$, $[\![\mathrm{N}]\!]_2$, $[\![\mathrm{N}]\!]_3$, and so on, with the arrows connecting them left implicit. Arrows between senses would then be recovered by means of subtyping. For instance, the arrow from $[\![\text{meat}]\!]_1$ to $[\![\text{meat}]\!]_2$ would be a consequence of the fact that $[\![\text{meat}]\!]_1(\mathrm{x}) < [\![\text{meat}]\!]_2(\mathrm{x})$ for all x. This approach can be made to work for common noun networks, where each sense is of type $\mathit{Ind} \to \mathit{Type}$, but does not generalize to more complex kinds of networks. In particular, it does not work for networks involving context updates, since unlike elements of *Type*, elements of *Update* are not related by the subtype relation.

2.6.2 Networks with presuppositions

The previous subsection described networks relating different noun senses. These are particular easy to implement, since they involve only one polymorphic variable, namely a generic individual x. However, a major finding in cognitive linguistics is that network structures are found in all parts of the lexicon, not only in nouns. For each of the types introduced in Section 2.5 – intersective adjectives, verbs, determiners, and so on – we would like to introduce a notion of arrow which describes a connection between senses of that type. Some of these are slightly more complicated to implement because of the presence of presuppositions. It is convenient to begin with arrows between context updates, since other kinds of arrows can be seen as parameterized versions of update arrows.

Recall that a sentence meaning is an element of *Update* which is defined as follows:

(165) $\mathit{Update} := \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \mathit{Prop} \end{bmatrix}$

where the first component is the presupposition type and the second component is the assertion. Suppose we have two elements $U, V : \mathit{Update}$ and we would like to formulate an arrow from U to V. This is given by an element of the following type:

(166) $$\mathrm{U} \rightarrow_{\mathit{Update}} \mathrm{V} \coloneqq \begin{bmatrix} \mathrm{f} : \mathrm{U.1} \rightarrow \mathrm{V.1} \\ \prod_{\mathrm{p:U.1}} \mathrm{U.2(p)} \rightarrow \mathrm{V.2(f(p))} \end{bmatrix}$$

Just as an element of *Update* has two components – a collection of presuppositions and an assertion – so an arrow between updates has two components – a presupposition map and an assertion map. The presupposition map f goes from the presuppositions of U to the presuppositions of V. The assertion map specifies, for any element p of U's presupposition type, how to go from the assertion of U at p to the assertion of V at $f(p)$ (note that we cannot take the assertion of V at p directly, because p belongs to U's presupposition type). Arrows between updates can be composed. Given three sentence meanings $U, V, W : \mathit{Update}$, and two arrows $F : U \rightarrow_{\mathit{Update}} V$ and $G : V \rightarrow_{\mathit{Update}} W$, we can compose F and G to get an arrow $G \circ_{\mathit{Update}} F : U \rightarrow_{\mathit{Update}} W$ which goes from U to W in a single step. This is implemented as one would expect:

(167) $$\mathrm{G} \circ_{\mathit{Update}} \mathrm{F} : \mathrm{U} \rightarrow_{\mathit{Update}} \mathrm{W}$$

$$\mathrm{G} \circ_{\mathit{Update}} \mathrm{F} \coloneqq \begin{pmatrix} \mathrm{G.1} \circ \mathrm{F.1} \\ \lambda \mathrm{p} : \mathrm{U.1}\ .\ \mathrm{G.2(G.1(p))} \circ \mathrm{F.2(p)} \end{pmatrix}$$

Moreover, given some update U : *Update*, this has an automatic identity arrow to itself, which is given by:

(168) $$\mathrm{id}_{\mathrm{U}} : \mathrm{U} \rightarrow_{\mathit{Update}} \mathrm{U}$$

$$\mathrm{id}_{\mathrm{U}} \coloneqq \begin{pmatrix} \lambda \mathrm{p} : \mathrm{U.1}\ .\ \mathrm{p} \\ \lambda \mathrm{p} : \mathrm{U.1}\ .\ \lambda \mathrm{q} : \mathrm{U.2(p)}\ .\ \mathrm{q} \end{pmatrix}$$

It follows that networks of context updates form partial orders.

To illustrate what a network of context updates looks like, it is useful to consider an example. Recall Jackendoff's analysis of the verb *climb* as consisting of three senses:

(169) $[\![\mathrm{climb}]\!]_{\mathrm{rise,clamber}}$ rising and clambering, e.g. *the man climbed the tree*

$[\![\mathrm{climb}]\!]_{\mathrm{rise}}$ rising only, e.g. *the snake climbed the tree*

$[\![\mathrm{climb}]\!]_{\mathrm{clamber}}$ clambering only, e.g. *the monkey climbed down the tree*

$\mathrm{Climb}_{\mathrm{rise,,\ clamber}}$ is the prototypical sense, whereas $\mathrm{climb}_{\mathrm{rise}}$ and $\mathrm{climb}_{\mathrm{clamber}}$ are less typical. Note that ascending and clambering have different presuppositions: roughly speaking, rising presupposes that the subject is a physical object, whereas clambering presupposes both that the subject is animate and that it has limbs which can be used to grasp (an animal such as a snake or worm cannot clamber because it lacks the necessary anatomy). Consider the simple statement:

(170) John climbed.

This will have a network of interpretations corresponding to the different senses of *climb*. Without specifying how these senses are derived, we might represent them as follows:

(171) $[\![\text{John climbed}]\!]_{\text{rise,clamber}}, [\![\text{John climbed}]\!]_{\text{rise}}, [\![\text{John climbed}]\!]_{\text{clamber}} : \textit{Update}$

$$[\![\text{John climbed}]\!]_{\text{rise,clamber}} \coloneqq \begin{bmatrix} \text{P} \coloneqq \begin{bmatrix} \text{x} : \textit{Ind} \\ \text{John(x)} \\ \text{physical(x)} \\ \text{has-limbs(x)} \end{bmatrix} \\ \lambda\text{p} : \text{P} \;.\; \left| \begin{bmatrix} \text{e} : \textit{Event} \\ \text{t} : \textit{Time} \\ \text{clambering(e)} \\ \text{rising(e)} \\ \text{past(t)} \\ \text{agent(p.x, e)} \\ \text{theme(p.x, e)} \\ \text{happen-at(e, t)} \end{bmatrix} \right| \end{bmatrix}$$

$$[\![\text{John climbed}]\!]_{\text{rise}} \coloneqq \begin{bmatrix} \text{P} \coloneqq \begin{bmatrix} \text{x} : \textit{Ind} \\ \text{John(x)} \\ \text{physical(x)} \end{bmatrix} \\ \lambda\text{p} : \text{P} \;.\; \left| \begin{bmatrix} \text{e} : \textit{Event} \\ \text{t} : \textit{Time} \\ \text{rising(e)} \\ \text{past(t)} \\ \text{theme(p.x, e)} \\ \text{happen-at(e, t)} \end{bmatrix} \right| \end{bmatrix}$$

$$
[\![\text{John climbed}]\!]_{\text{clamber}} \coloneqq \left[\begin{array}{l}
\text{P} \coloneqq \left[\begin{array}{l}\text{x} : \textit{Ind}\\ \text{John(x)}\\ \text{physical(x)}\\ \text{has-limbs(x)}\end{array}\right]\\[2ex]
\lambda\text{p} : \text{P} \;.\; \left|\left[\begin{array}{l}\text{e} : \textit{Event}\\ \text{t} : \textit{Time}\\ \text{clambering(e)}\\ \text{past(t)}\\ \text{agent(p.x, e)}\\ \text{happen-at(e, t)}\end{array}\right]\right|
\end{array}\right]
$$

As shown, $[\![\text{John climbed}]\!]_{\text{rise,clamber}}$ presupposes that John is a physical object with limbs and asserts that he rose and clambered; $[\![\text{John climbed}]\!]_{\text{rise}}$ presupposes that John is a physical object and asserts that he rose; and $[\![\text{John climbed}]\!]_{\text{clamber}}$ presupposes that John is a physical object with limbs and asserts that he clambered. The connections between the three senses are derived from the arrows in the *climb* network:

(172)
$$
\begin{tikzcd}
{[\![\text{John climbed}]\!]_{\text{rise}}} & & {[\![\text{John climbed}]\!]_{\text{clamber}}} \\
& {[\![\text{John climbed}]\!]_{\text{rise,clamber}}} \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

Consider the arrow from $[\![\text{John climbed}]\!]_{\text{rise,clamber}}$ to $[\![\text{John climbed}]\!]_{\text{rise}}$. This is an arrow between context updates, so it consists of a presupposition map and an assertion map. It is implemented as follows:

(173) $[\![\text{John climbed}]\!]^{\text{rise}}_{\text{rise,clamber}} : [\![\text{John climbed}]\!]_{\text{rise,clamber}} \to_{\textit{Update}} [\![\text{John climbed}]\!]_{\text{rise}}$

$$
[\![\text{John climbed}]\!]^{\text{rise}}_{\text{rise,clamber}} \coloneqq
$$

$$
\left(\begin{array}{l}
\lambda\text{p} : \left[\begin{array}{l}\text{x} : \textit{Ind}\\ \text{John(x)}\\ \text{physical(x)}\\ \text{has-limbs(x)}\end{array}\right] . \begin{pmatrix}\text{p.1}\\ \text{p.2}\\ \text{p.3}\end{pmatrix}\\[4ex]
\lambda\text{p} : \left[\begin{array}{l}\text{x} : \textit{Ind}\\ \text{John(x)}\\ \text{physical(x)}\\ \text{has-limbs(x)}\end{array}\right] . \left|\lambda\text{q} : \left[\begin{array}{l}\text{e} : \textit{Event}\\ \text{t} : \textit{Time}\\ \text{clambering(e)}\\ \text{past(t)}\\ \text{agent(p.x, e)}\\ \text{happen-at(e, t)}\end{array}\right] . \left|\begin{pmatrix}\text{p.1}\\ \text{p.2}\\ \text{p.4}\\ \text{p.5}\\ \text{p.7}\\ \text{p.8}\end{pmatrix}\right|\right|
\end{array}\right)
$$

The presupposition map takes a context in which John is known to be a physical object with limbs and drops the limbs component, leaving only the proof that he is a physical object. The assertion map takes a context in which John is a physical object with limbs and transforms the assertion that John rose and clambered into an assertion that he rose, by dropping the components associated with clambering. Note the use of propositional truncation, which ensures that the assertion map relates two propositions. The other arrow, from $[\![\text{John climbed}]\!]_{\text{rise,clamber}}$ to $[\![\text{John climbed}]\!]_{\text{clamber}}$, works in a similar way. In a compositional account, both of these arrows would be derived from the *climb* network.

given two...	an arrow is of type...
contexts $A, B : Type$	$A \to B$
propositions $P, Q : Prop$	$P \to Q$
common noun meanings $N_1, N_2 : Ind \to Type$	$\prod_{x:Ind} N_1(x) \to N_2(x)$
sentence meanings $U, V : Update$	$U \to_{Update} V := \begin{bmatrix} \text{f} : \text{U.1} \to \text{V.1} \\ \prod_{\text{p:U.1}} \text{U.2(p)} \to \text{V.2(f(p))} \end{bmatrix}$
sentential modifiers $M_1, M_2 : Update \to Update$	$\prod_{U:Update} M_1(U) \to_{Update} M_2(U)$
intersective adjective meanings $A_1, A_2 : Ind \to Update$	$\prod_{x:Ind} A_1(x) \to_{Update} A_2(x)$
determiner meanings $D_1, D_2 : (Ind \to Type) \to (Ind \to Update) \to Update$	$\prod_{\text{N}:Ind \to Type} \prod_{\text{P}:Ind \to Update} D_1(N)(P) \to_{Update} D_2(N)(P)$
generalized quantifiers $Q_1, Q_2 : (Ind \to Update) \to Update$	$\prod_{P:Ind \to Update} Q_1(P) \to_{Update} Q_2(P)$

Table 2.5: Semantic types and their corresponding arrow types.

Having formalized arrows between context updates, we can generalize this to parameterized context updates. For example, given two intersective adjective meanings:

(174) $\quad \text{A}_1, \text{A}_2 : Ind \to Update$

An arrow from A_1 to A_2 will be an indexed family of update maps, one for every individual:

(175) $\quad \prod_{\text{x}:Ind} \text{A}_1(\text{x}) \to_{Update} \text{A}_2(\text{x})$

Similarly, given two quantificational determiners:

(176) $\mathrm{D}_1, \mathrm{D}_2 : (\mathit{Ind} \to \mathit{Type}) \to (\mathit{Ind} \to \mathit{Update}) \to \mathit{Update}$

An arrow from D_1 to D_2 consists of a family of update maps, one for every possible noun and property argument:

(177) $\prod_{\mathrm{N}:\mathit{Ind}\to\mathit{Type}} \prod_{\mathrm{P}:\mathit{Ind}\to\mathit{Update}} \mathrm{D}_1(\mathrm{N})(\mathrm{P}) \to_{\mathit{Update}} \mathrm{D}_2(\mathrm{N})(\mathrm{P})$

We can now associate an arrow type to every class of semantic object introduced in Section 2.5. Table 2.5 gives, for any pair of objects belonging to the same semantic type, the corresponding type for an arrow between them. In the next chapter, this table will be extended to include gradable adjectives. One could also introduce networks based on other semantic types, such as verbs, prepositions, tense/aspect morphology, modals, and so on.

2.6.3 Basic compositionality

We have seen how various kinds of lexical networks can be implemented in DTT Semantics, but it remains to be shown that these networks can behave compositionally. As discussed in Section 1.2.4, the arrows in a lexical network can be preserved by composition, lifted to the level of entire phrases or sentences. The simplest kind of network composition is when a network is applied to a non-network argument. This operation is automatically monotonic, meaning it preserves arrows in their original direction. To illustrate, suppose we take the *meat* network discussed previously and apply it to some individual $m : \mathit{Ind}$. This is done by taking every sense and arrow in the *meat* network and applying it to x, as follows:

(178)
$$\left[\begin{tikzcd}
 & [\![\text{meat}]\!]_8 & & & \\
{[\![\text{meat}]\!]_3} \arrow[ur] & & & & \\
{[\![\text{meat}]\!]_2} \arrow[u] & {[\![\text{meat}]\!]_4} \arrow[uu] & {[\![\text{meat}]\!]_5} \arrow[uul] & {[\![\text{meat}]\!]_6} & {[\![\text{meat}]\!]_7} \\
 & & {[\![\text{meat}]\!]_1} \arrow[ull] \arrow[ul] \arrow[u] \arrow[ur] \arrow[urr] & &
\end{tikzcd}\right](x)$$

$$= \begin{tikzcd}
 & [\![\text{meat}]\!]_8(x) & & & \\
{[\![\text{meat}]\!]_3(x)} \arrow[ur] & & & & \\
{[\![\text{meat}]\!]_2(x)} \arrow[u] & {[\![\text{meat}]\!]_4(x)} \arrow[uu] & {[\![\text{meat}]\!]_5(x)} \arrow[uul] & {[\![\text{meat}]\!]_6(x)} & {[\![\text{meat}]\!]_7(x)} \\
 & & {[\![\text{meat}]\!]_1(x)} \arrow[ull] \arrow[ul] \arrow[u] \arrow[ur] \arrow[urr] & &
\end{tikzcd}$$

This application makes sense because every sense and arrow in a noun network contains a lambda abstraction over individuals. In general, given a network which is parameterized by values of some type A, one can always reduce the network by taking its senses and arrows at some value $a : A$.

Another kind of network composition is when, rather than a network acting on an argument, some function acts on a network. For example, suppose we want to apply the definite article *the* to the meat network, as shown below:

(179) $$[\![\text{the}]\!]\left(\begin{tikzcd} & {[\![\text{meat}]\!]_8} & & & \\ {[\![\text{meat}]\!]_3} \arrow[ur] & & & & \\ {[\![\text{meat}]\!]_2} \arrow[u] & {[\![\text{meat}]\!]_4} \arrow[uu] & {[\![\text{meat}]\!]_5} \arrow[uul] & {[\![\text{meat}]\!]_6} & {[\![\text{meat}]\!]_7} \\ & & {[\![\text{meat}]\!]_1} \arrow[ull] \arrow[ul] \arrow[u] \arrow[ur] \arrow[urr] & & \end{tikzcd}\right)$$

In order to evaluate this application, we must know how $[\![\text{the}]\!]$ acts *both on senses and on arrows.* In Section 2.5.5, I proposed the following implementation for $[\![\text{the}]\!]$:

(180) $$[\![\text{the}]\!] : (\mathit{Ind} \to \mathit{Type}) \to (\mathit{Ind} \to \mathit{Update}) \to \mathit{Update}$$

$$\begin{aligned} [\![\text{the}]\!] \coloneqq \lambda \mathrm{N} : \mathit{Ind} \to \mathit{Type}\ . \\ \lambda \mathrm{P} : \mathit{Ind} \to \mathit{Update}\ . \\ \left(\begin{array}{l} \mathrm{C} \coloneqq \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{N(x)} \\ \mathrm{P(x).1} \end{bmatrix} \\ \lambda \mathrm{c} : \mathrm{C}\ .\ \mathrm{P(c.1).2(c.3)} \end{array}\right) \end{aligned}$$

This describes the action of $[\![\text{the}]\!]$ on the senses of a noun network, but we have yet to specify its action on arrows. Suppose we have two arbitrary noun meanings $N_1, N_2 : \mathit{Ind} \to \mathit{Type}$ which are connected by an arrow:

(181) $$\alpha : \textstyle\prod_{\mathrm{x}:\mathit{Ind}} \mathrm{N}_1(x) \to \mathrm{N}_2(x)$$

For $[\![\text{the}]\!]$ to act on an entire lexical network, we must be able to lift this to an arrow of the form:

(182) $$[\![\text{the}]\!](\alpha) : \textstyle\prod_{\mathrm{P}:\mathit{Ind}\to\mathit{Update}} [\![\text{the}]\!](\mathrm{N}_1)(\mathrm{P}) \to_{\mathit{Update}} [\![\text{the}]\!](\mathrm{N}_2)(\mathrm{P})$$

In other words, the lifted arrow $[\![\text{the}]\!](\alpha)$ must tell us how to go from $[\![\text{the}]\!]$ at N_1 to $[\![\text{the}]\!]$ at N_2, for any possible choice of property argument. This is implemented as follows:

(183) $[\![\text{the}]\!](\alpha) \coloneqq \lambda \mathrm{P} : \mathit{Ind} \to \mathit{Update}\ .$

$$\begin{pmatrix} \lambda \mathrm{c} : \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{N}_1(\mathrm{x}) \\ \mathrm{P}(\mathrm{x}).1 \end{bmatrix} . \begin{pmatrix} \mathrm{c}.1 \\ \alpha(\mathrm{c}.1)(\mathrm{c}.2) \\ \mathrm{c}.3 \end{pmatrix} \\ \lambda \mathrm{c} : \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{N}_1(\mathrm{x}) \\ \mathrm{P}(\mathrm{x}).1 \end{bmatrix} . \ \lambda \mathrm{d} : \mathrm{P}(\mathrm{c}.1).2(\mathrm{c}.3)\ .\ \mathrm{d} \end{pmatrix}$$

As shown, the lifted arrow works by using the original arrow α to convert the presupposition that there exists some individual satisfying N_1 into the presupposition that there exists some individual satisfying N_2. The assertion map is simply the identity, because the assertion does not contain information pertaining to the noun argument. Now that we know how $[\![\text{the}]\!]$ acts on both senses and arrows, we can evaluate the application in (179) to get the following network:

(184)
$$\begin{tikzcd}
& {[\![\text{the}]\!]([\![\text{meat}]\!]_8)} & & & \\
{[\![\text{the}]\!]([\![\text{meat}]\!]_3)} \arrow[ur] & & & & \\
{[\![\text{the}]\!]([\![\text{meat}]\!]_2)} \arrow[u] & {[\![\text{the}]\!]([\![\text{meat}]\!]_4)} \arrow[uu] & {[\![\text{the}]\!]([\![\text{meat}]\!]_5)} \arrow[uul] & {[\![\text{the}]\!]([\![\text{meat}]\!]_6)} & {[\![\text{the}]\!]([\![\text{meat}]\!]_7)} \\
& & {[\![\text{the}]\!]([\![\text{meat}]\!]_1)} \arrow[ull] \arrow[ul] \arrow[u] \arrow[ur] \arrow[urr] & &
\end{tikzcd}$$

in which $[\![\text{the}]\!]$ is applied to all the senses and arrows.

To generalize, any word which is capable of acting on a lexical network must specify, as part of its lexical entry, both an action on senses and an action on arrows. Given a function $f : A \to B$, where A and B are semantic types (noun meanings, context updates, intersective adjectives, etc.), I shall say that f 'lifts arrows' iff given two arbitrary elements $a, b : A$ connected by an arrow $a \to_A b$ – that is, an arrow of the appropriate type for elements of A – this can be converted to an arrow $f(a) \to_B f(b)$, of the appropriate type for elements of B. A function may be able to lift arrows in more than one argument position, in which case a separate rule is required for each argument. For example, in addition to lifting arrows between noun senses, $[\![\text{the}]\!]$ can also lift arrows between property senses. Given two properties $P_1, P_2 : \mathit{Ind} \to \mathit{Update}$ connected by an arrow:

(185) $\beta : \prod_{\mathrm{x}:\mathit{Ind}} \mathrm{P}_1(x) \to \mathrm{P}_2(x)$

this is lifted to an arrow:

(186) $[\![\text{the}]\!](-)(\beta) : \prod_{\mathrm{N}:\mathit{Ind}\to\mathit{Type}} [\![\text{the}]\!](\mathrm{N})(\mathrm{P}_1) \to_{\mathit{Update}} [\![\text{the}]\!](\mathrm{N})(\mathrm{P}_2)$

$[\![\text{the}]\!](-)(\beta) \coloneqq \lambda \mathrm{N} : \mathit{Ind} \to \mathit{Type}\ .$

$$
\begin{pmatrix}
\lambda \mathrm{c} : \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{N}(\mathrm{x}) \\ \mathrm{P}_1(\mathrm{x}).1 \end{bmatrix} . \begin{pmatrix} \mathrm{c}.1 \\ \mathrm{c}.2 \\ \beta(\mathrm{c}.1).1(\mathrm{c}.3) \end{pmatrix} \\
\lambda \mathrm{c} : \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{N}(\mathrm{x}) \\ \mathrm{P}_1(\mathrm{x}).1 \end{bmatrix} . \ \lambda \mathrm{d} : \mathrm{P}_1(\mathrm{c}.1).2(\mathrm{c}.3) \ . \ \beta(\mathrm{c}.1).2(\mathrm{d})
\end{pmatrix}
$$

which goes from $[\![\text{the}]\!](N)(P_1)$ to $[\![\text{the}]\!](N)(P_2)$, for arbitrary N.

As discussed previously, some functions have the effect of reversing a network's arrows. I shall say that a function $f : A \to B$ is 'anti-monotone' or 'downwards monotone' iff it lifts arrow $a \to_A b$ to an arrow in the opposite direction $f(b) \to_B f(a)$. For instance, suppose that instead of applying $[\![\text{the}]\!]$ to the meat network, we instead apply $[\![\text{all}]\!]$. The action of $[\![\text{all}]\!]$ on senses was given in Section 2.5.5:

(187) $[\![\text{all}]\!] : (\mathit{Ind} \to \mathit{Type}) \to (\mathit{Ind} \to \mathit{Update}) \to \mathit{Update}$

$$
\begin{aligned}
[\![\text{all}]\!] \coloneqq \ & \lambda \mathrm{N} : \mathit{Ind} \to \mathit{Type} \ . \\
& \quad \lambda \mathrm{P} : \mathit{Ind} \to \mathit{Update} \ . \\
& \qquad \begin{pmatrix} \mathrm{C} \coloneqq \prod_{\mathrm{x}:\mathit{Ind}} \mathrm{N}(\mathrm{x}) \to \mathrm{P}(\mathrm{x}).1 \\ \lambda \mathrm{c} : \mathrm{C} \ . \ \prod_{\mathrm{x}:\mathit{Ind}} \prod_{\mathrm{n}:\mathrm{N}(\mathrm{x})} \mathrm{P}(\mathrm{x}).2(\mathrm{c}(\mathrm{x})(\mathrm{n})) \end{pmatrix}
\end{aligned}
$$

Given two noun meanings $N_1, N_2 : \mathit{Ind} \to \mathit{Type}$ and an arrow $\alpha : \prod_{x:\mathit{Ind}} N_1(x) \to N_2(x)$, this is lifted to an arrow:

(188) $[\![\text{all}]\!](\alpha) : [\![\text{all}]\!](N_2) \to_{\mathit{Update}} [\![\text{all}]\!](N_1)$

Notice that the direction gets reversed: the original arrow went from N_1 to N_2 but the lifted arrow goes from $[\![\text{all}]\!](N_2)$ to $[\![\text{all}]\!](N_1)$. It is implemented as follows:

(189) $[\![\text{all}]\!](\alpha) \coloneqq \lambda \mathrm{P} : \mathit{Ind} \to \mathit{Update}$.

$$
\begin{aligned}
& \text{let } \mathrm{E} \coloneqq [\![\text{all}]\!](N_2) \text{ in} \\
& \quad \begin{pmatrix} \lambda \mathrm{c} : \mathrm{E}.1 \ . \ (\lambda \mathrm{x} : \mathit{Ind} \ . \ \lambda \mathrm{r} : \mathrm{N}_1(\mathrm{x}) \ . \ \mathrm{c}(\mathrm{x})(\alpha(\mathrm{x})(\mathrm{r}))) \\ \lambda \mathrm{c} : \mathrm{E}.1 \ . \ \lambda \mathrm{d} : \mathrm{E}.2(\mathrm{d}) \ . \ (\lambda \mathrm{x} : \mathit{Ind} \ . \ \lambda \mathrm{r} : N_1(\mathrm{x}) \ . \ \mathrm{d}(\mathrm{x})(\alpha(\mathrm{x})(\mathrm{r}))) \end{pmatrix}
\end{aligned}
$$

Applying $[\![\text{all}]\!]$ to the meat network therefore yields a network in which all the arrows are reversed:

(190)
$$
[\![\text{all}]\!] \left(
\begin{tikzcd}
 & {[\![\text{meat}]\!]_8} & & & \\
{[\![\text{meat}]\!]_3} \arrow[ur] & & & & \\
{[\![\text{meat}]\!]_2} \arrow[u] & {[\![\text{meat}]\!]_4} \arrow[uu] & {[\![\text{meat}]\!]_5} \arrow[uul] & {[\![\text{meat}]\!]_6} & {[\![\text{meat}]\!]_7} \\
 & & {[\![\text{meat}]\!]_1} \arrow[ull] \arrow[ul] \arrow[u] \arrow[ur] \arrow[urr] & &
\end{tikzcd}
\right)
$$

$$
= \begin{tikzcd}
& \llbracket \text{all} \rrbracket(\llbracket \text{meat} \rrbracket_8) \arrow[dl] \arrow[dd] \arrow[ddr] & & & \\
\llbracket \text{all} \rrbracket(\llbracket \text{meat} \rrbracket_3) \arrow[d] & & & & \\
\llbracket \text{all} \rrbracket(\llbracket \text{meat} \rrbracket_2) \arrow[drr] & \llbracket \text{all} \rrbracket(\llbracket \text{meat} \rrbracket_4) \arrow[dr] & \llbracket \text{all} \rrbracket(\llbracket \text{meat} \rrbracket_5) \arrow[d] & \llbracket \text{all} \rrbracket(\llbracket \text{meat} \rrbracket_6) \arrow[dl] & \llbracket \text{all} \rrbracket(\llbracket \text{meat} \rrbracket_7) \arrow[dll] \\
& & \llbracket \text{all} \rrbracket(\llbracket \text{meat} \rrbracket_1) & &
\end{tikzcd}
$$

As a result, the weakest interpretations of *meat* give rise to the strongest interpretations of *all meat*, and vice versa.

When it comes to determiners like *the* and *all*, the importance of monotonicity is widely recognised. For example, it is a well-known observation that downwards monotone determiners can license negative polarity items like *ever*, *any*, *at all*, and so on (Fauconnier 1975, Ladusaw 1979). Facts like these suggest that the monotonicity of a determiner ought to be accessible to the grammar. In a Montague-style semantics, a determiner's monotonicity is difficult to ascertain from its lexical entry, unless hard-coded in the form of a syntactic marker or meaning postulate. By contrast, in a compositional theory of lexical networks based on DTT, the monotonicity of a determiner is simply its action on arrows, which is part of its lexical entry. In the framework I am proposing, the notion of monotonicity is extended far beyond determiners to encompass all words which are capable of acting on a network.

It is worth pointing out that not all words which act on the senses of a lexical network will also lift arrows. A function $f : A \to B$ which has has no effect on arrows $a \to_A b$ is called 'non-monotone'. For example, suppose that we alter the interpretation of *the* by adding a uniqueness presupposition, as follows:

(191) $\llbracket \text{the} \rrbracket' : (\mathit{Ind} \to \mathit{Type}) \to (\mathit{Ind} \to \mathit{Update}) \to \mathit{Update}$

$$
\begin{aligned}
\llbracket \text{the} \rrbracket' \coloneqq \lambda \text{N} : \mathit{Ind} \to \mathit{Type}\ .& \\
\lambda \text{P} : \mathit{Ind} \to \mathit{Update}\ .& \\
& \left(\begin{array}{l} \text{C} \coloneqq \begin{bmatrix} \text{x} : \mathit{Ind} \\ \text{N(x)} \\ \text{P(x).1} \\ \prod_{\text{y}:\mathit{Ind}} \text{N(y)} \to \text{y} =_{\mathit{Ind}} \text{x} \end{bmatrix} \\ \lambda \text{c} : \text{C}\ .\ \text{P(c.1).2(c.3)} \end{array} \right)
\end{aligned}
$$

The added presupposition requires that x should be the only individual satisfying the noun denotation N. Now it is no longer possible to write a lifting rule for arrows between noun senses. Roughly speaking, this is because the fact that there is a unique referent of *the N* for some strong interpretation of *N* does not imply that there is a unique referent for some weaker interpretation of *N*, nor the reverse.

2.6.4 Applying networks to networks

Thus far, we have covered applying a network to a single argument and applying a single function to a network. In the general case, one applies an entire network of functions to an entire network of arguments. I shall now describe how this works abstractly. Suppose that we have a network of arguments, in which all the senses are of type A. Consider two senses $\mathrm{a}, \mathrm{b} : A$ linked by an arrow $\mathrm{f} : \mathrm{a} \rightarrow_A \mathrm{b}$:

(192) $$\boxed{\mathrm{a} \xrightarrow{\mathrm{f}} \mathrm{b} \quad _{A}}$$

Now suppose we have a network of senses of type $A \rightarrow B$, each of which can take an element of A as an argument. Consider two senses $F, G : A \rightarrow B$ which are connected by an arrow α:

(193) $$\boxed{\mathrm{F} \xrightarrow{\alpha} \mathrm{G} \quad _{A \rightarrow B}}$$

Since both F and G have type $A \rightarrow B$, the type of α must be $\prod_{x:A} F(x) \rightarrow_B G(x)$. Finally, suppose that F and G are monotone, so they not only act on senses of type A, but are also capable of lifting $\rightarrow_A$ arrows. We can therefore apply (193) to (192), as follows:

(194) $$\boxed{\mathrm{F} \xrightarrow{\alpha} \mathrm{G} \quad _{A \rightarrow B}}\left(\boxed{\mathrm{a} \xrightarrow{\mathrm{f}} \mathrm{b} \quad _{A}}\right)$$

Notice that F can act on either a or b, and likewise G can act on either a or b. The result is four senses of type B forming a square:

(195) $$= \boxed{\begin{tikzcd} \mathrm{F(a)} \arrow[r, "\mathrm{F(f)}"] \arrow[d, "\alpha(\mathrm{a})"'] \arrow[dr, phantom, "\checkmark"] & \mathrm{F(b)} \arrow[d, "\alpha(\mathrm{b})"] \\ \mathrm{G(a)} \arrow[r, "\mathrm{G(f)}"'] & \mathrm{G(b)} \end{tikzcd} \quad _{B}}$$

The horizontal arrows correspond to lifting the arrow f under F or G respectively, whereas the vertical arrows correspond to taking the component of α at a and b respectively. Notice that there are two ways to get from $F(a)$ to $G(b)$, depending on which path one takes around the square. This square *commutes* (indicated by ✓), meaning that the two paths give the same result, in other words $\alpha(b) \circ F(f) = G(f) \circ \alpha(a)$.. We can view the entire commuting square as the composition of the arrow f with the arrow α.

In general, when one network acts on another in a composition $N(M)$, the number of senses multiplies, giving us one sense for every pair $n(m)$, where n

is a sense in N and m is a sense in M. Each arrow in N combines with each arrow in M to give a commuting square, in the manner just described. It follows that the number of potential interpretations for an expression grows exponentially in the number of words. This is a potential problem from the point of view of real-time semantic interpretation, since the number of possible interpretations will quickly overwhelm the resources of the interpreter. For example, a sentence with 10 words, each of which has 5 possible senses, will have $5^{10} = 9765625$ possible interpretations. This problem is discussed further in Section 4.7. In brief, my solution is that the interpreter aims to find the strongest possible assertion which is compatible with the context. This is done by building the interpretation network in a gradual breadth-first manner, beginning with the weakest or most general senses and gradually strengthening those which are found to be felicitous. When a sense is found to be inconsistent or infelicitous, the interpreter removes it from consideration, along with all the senses with arrows into it. This has the effect of dramatically reducing the number of senses which the interpreter needs to consider.

2.6.5 Skew monotonicity

Thus far, we have seen examples of monotone functions like $[\![\text{the}]\!]$ and anti-monotone functions like $[\![\text{all}]\!]$. However, the usual story about monotonicity is complicated by the fact that we are explicitly representing presuppositions. Some functions which return context updates act monotonically with respect to presuppositions but anti-monotonically with respect to assertions. The paradigmatic example is *not*, whose action on senses is repeated below:

(196) $[\![\text{not}]\!] : \mathit{Update} \to \mathit{Update}$

$$[\![\text{not}]\!] \coloneqq \lambda \text{U} : \mathit{Update}\ .\ \begin{pmatrix}\text{U.1} \\ \lambda \text{p} : \text{U.1}\ .\ \neg \text{U.2(p)}\end{pmatrix}$$

As shown, $[\![\text{not}]\!]$ takes an update and copies its presuppositions whilst negating its assertion. It follows that, given a map of context updates $\alpha : \text{U} \to_{\mathit{Update}} \text{V}$, $[\![\text{not}]\!]$ preserves the direction of the presupposition map whilst reversing the direction of the assertion map. I shall write the type of the lifted arrow $[\![\text{not}]\!](\alpha)$ as follows:

(197) $[\![\text{not}]\!](\alpha) : [\![\text{not}]\!](\text{U}) \rightleftarrows_{\mathit{Update}} [\![\text{not}]\!](\text{V})$

As the notation $\rightleftarrows_{\mathit{Update}}$ suggests, the presupposition map goes from the update on the left to the update on the right, whereas the assertion map goes in the opposite direction, from the update on the right to the update on the left. I refer to this as a 'skewed' update arrow, in contrast to the ordinary 'parallel' kind of update arrow, signified by $\to_{\mathit{Update}}$. The type of a skew update arrow is defined as follows, for any updates A and B:

(198) $$\text{A} \rightleftarrows_{\mathit{Update}} \text{B} \coloneqq \begin{bmatrix}\text{f} :\ \text{A.1} \to \text{B.1} \\ \prod_{\text{p:A.1}} \text{B.2(f(p))} \to \text{A.2(p)}\end{bmatrix}$$

Given this definition, the implementation of $[\![\text{not}]\!](\alpha)$ is:

(199) $[\![\text{not}]\!](\alpha) \coloneqq \text{let } \mathrm{U}' \coloneqq [\![\text{not}]\!](\mathrm{U}) \text{ in}$

$$\text{let } \mathrm{V}' \coloneqq [\![\text{not}]\!](\mathrm{V}) \text{ in}$$
$$\begin{pmatrix} \lambda \mathrm{p} : \mathrm{U}'.1 \;.\; \alpha.1(\mathrm{p}) \\ \lambda \mathrm{p} : \mathrm{U}'.1 \;.\; \lambda \mathrm{q} : \mathrm{V}'.2(\alpha.1(\mathrm{p})) \;.\; (\lambda \mathrm{r} : \mathrm{U}'.2(\mathrm{p}) \;.\; \mathrm{q}(\alpha.2(\mathrm{p})(\mathrm{r}))) \end{pmatrix}$$

To understand this implementation, recall the intuitionistic definition of negation as a function type: $\neg A \equiv A \to 0$.

Like parallel networks, skew networks form partial orders. Given three context updates $A, B, C : \mathit{Update}$ and two skew arrows $\alpha : \mathrm{A} \rightleftarrows_{\mathit{Update}} \mathrm{B}$ and $\beta : \mathrm{B} \rightleftarrows_{\mathit{Update}} \mathrm{C}$, the composition is defined as follows:

(200) $\beta \circ_{\mathit{Update}} \alpha : \mathrm{A} \rightleftarrows_{\mathit{Updt}} \mathrm{C}$

$$\beta \circ_{\mathit{Update}} \alpha \coloneqq \begin{pmatrix} \beta.1 \circ \alpha.1 \\ \lambda \mathrm{p} : \mathrm{A}.1 \;.\; \alpha.2(\mathrm{p}) \circ \beta.2(\alpha.1(\mathrm{p})) \end{pmatrix}$$

As one would expect, the two presupposition maps are composed in the reverse order to the two assertion maps. Finally, any context update $\mathrm{U} : \mathit{Update}$, is associated with a unique skew identity arrow id_{U}, which is the same as the parallel version defined in (168). The concept of a skew arrow generalizes from context updates to parameterized context updates. Table 2.6 shows some semantic types which can have skew arrows.

given two...	a skew arrow is of type...
sentence meanings $U, V : \mathit{Update}$	$U \rightleftarrows_{\mathit{Update}} V \coloneqq \begin{bmatrix} \mathrm{f} : \mathrm{U}.1 \to \mathrm{V}.1 \\ \prod_{\mathrm{p}:\mathrm{U}.1} \mathrm{V}.2(\mathrm{f}(\mathrm{p})) \to \mathrm{U}.2(\mathrm{p}) \end{bmatrix}$
sentential modifiers $M_1, M_2 : \mathit{Update} \to \mathit{Update}$	$\prod_{U:\mathit{Update}} M_1(U) \rightleftarrows_{\mathit{Update}} M_2(U)$
intersective adjective meanings $A_1, A_2 : \mathit{Ind} \to \mathit{Update}$	$\prod_{x:\mathit{Ind}} A_1(x) \rightleftarrows_{\mathit{Update}} A_2(x)$
determiner meanings $D_1, D_2 : (\mathit{Ind} \to \mathit{Type}) \to (\mathit{Ind} \to \mathit{Update}) \to \mathit{Update}$	$\prod_{\mathrm{N}:\mathit{Ind}\to\mathit{Type}} \prod_{\mathrm{P}:\mathit{Ind}\to\mathit{Update}} D_1(N)(P) \rightleftarrows_{\mathit{Update}} D_2(N)(P)$
generalized quantifiers $Q_1, Q_2 : (\mathit{Ind} \to \mathit{Update}) \to \mathit{Update}$	$\prod_{P:\mathit{Ind}\to\mathit{Update}} Q_1(P) \rightleftarrows_{\mathit{Update}} Q_2(P)$

Table 2.6: Semantic types which can have skew arrows.

Chapter 3

The Lexical Semantics of Spatial Adjectives

Contents

3.1 Motivation

The previous chapter explained how the lexical networks found in cognitive linguistics can be made compatible with compositional semantics using DTT. The aim of this chapter is to apply these ideas to the lexical semantics of spatial adjectives such as *long*, *short*, *tall*, *wide*, *narrow*, *thick*, *thin*, *deep* and *shallow*. These adjectives are highly polysemous, and hence provide good examples of lexical networks – for example, the adjective *wide* can refer to a secondary axis (e.g. *wide ruler*), the axis orthogonal to an observer (e.g. *wide bus*), or the magnitude of an area (e.g. *wide courtyard*). The structure of the network reflects prototypicality and abstract similarity: two senses with a common meet are perceived as deriving from the same prototype, whereas two senses with a common join are perceived as having a common generalization. This chapter draws heavily on the work of cognitive linguists, particularly Clark (1973), Vandeloise (1988, 1993), Dirven & Taylor (1986), Herskovits (1987) and Vogel (2004).

I have decided to focus primarily on the spatial senses of these adjectives rather than non-spatial or metaphorical senses, such as the use of *long* to describe time (*long meeting*), or the use of *high* to describe pitch (*high melody*). The reason for this is that the spatial senses are themselves highly complex and in need of detailed explanation, a task which is complicated significantly by the inclusion of metaphorical senses. Spatial senses can be described by combining a set of primitive notions such as 'axis', 'position', 'path', and so on, whereas the internal structure of non-spatial senses is more difficult to formalize, since the primitives underlying non-geometric domains are less well understood. Following Vogel (2004), the main criterion for distinguishing spatial from non-spatial senses is the ability to combine with distance-denoting measure phrases – for instance, *long table* is a spatial sense since one can say *10m long table*, but *long meeting* is not, since one cannot say ?*10m long meeting* in reference to its temporal duration.

In order to formalize spatial senses, I make two additions to the usual semantic ontology: degrees and vectors. *Degrees* are abstract representations of the extent to which an individual possesses a certain property (length, weight, intelligence, loudness, etc.). As we saw in Chapter 1, degrees organised into scales can explain a wide range of gradable phenomena, including comparison, polarity and cross-scalar incommensurability (Cresswell 1976, von Stechow 1984, Klein 1980, Bierwisch 1989, Kennedy 1999, Kennedy & McNally 2005). *Vectors* are geometric entities with a direction and magnitude. Originally introduced into semantics to describe locative prepositions (Zwarts 1997, Zwarts & Winter 2000), they have been fruitfully applied to a number of other phenomena, including directional prepositions and telicity (Zwarts 2005), gradability and measure phrase combination (Faller 2000, Winter 2001*a*), and general spatial vocabulary relating to place, size, orientation, shape and parts (Zwarts 2003).

This chapter is organised as follows. Section 3.2 explains the representation of degrees, gradable adjectives and gradable adjective networks. Section 3.3 is an introduction to Vector Space Semantics and introduces the semantic primitives involved in spatial adjectives. With these theoretical tools in hand, Sections 3.4-3.9 are dedicated to the adjectives *high/low*, *tall*, *long/short*, *wide/narrow*, *deep/shallow* and *thick/thin* respectively. Each of these sections gives a brief descriptive introduction to the different senses of that adjective before presenting my analysis. Having formalized the different spatial senses, I show how they connect together into a lexical network. The relationship between antonymous pairs of adjectives is explained in Section 3.4.3 which discusses the relationship between *high* and *low*. From that point on, I discuss only those negative polarity adjectives which are not perfect antonyms of their positive counterparts. For example, *short* has its own section since it is not a perfect antonym of *long*, whereas *low* does not, being a perfect antonym of *high*.

3.2 Implementing gradable adjectives

3.2.1 Typing degrees

Recall from the discussion of degree semantics in Section 1.4, that degrees are organised into *scales*. In a traditional degree-based approach, a scale is though of as something like a triple $\langle D, \leq, \delta \rangle$, where D is a set of degrees, $\leq$ is an order relation which is at least partial, and δ is a dimension parameter. Scales are built from degrees, in the sense that each scale has an underlying set of degrees D. This has some unintuitive consequences. For instance, it suggests that it makes sense to think of an abstract degree $d \in D$, separate from any particular scale, but it is not clear what kind of thing d would be. Moreover, the intention is that the set of degrees for two different scales should be disjoint, but nothing in the definition forces this to be the case. For instance, nothing seems to prevent the denotations of *John's height*, *Jane's average walking speed* and *150kg* from being identical as elements of D. On this account, degrees do not possess their dimension and polarity inherently, but only when considered as belonging to a particular scale.

From the perspective of DTT, it makes more sense to think of the type of degrees as being dependent on a scale parameter, so that degrees on distinct scales are inherently disjoint. Let us assume a type *Scale*, elements of which are labels for scales:

(201) $\mathrm{dist}, \mathrm{weight}, \mathrm{temp}, \mathrm{prob}, \mathrm{durtn}, \ldots : \mathit{Scale}$

For every scale, there is a corresponding type of negative degrees and positive degrees for that scale, formed as follows:

(202) $\mathrm{s} : \mathit{Scale} \vdash \mathit{Degree}(\mathrm{s}, +) : \mathit{Type}$

$\mathrm{s} : \mathit{Scale} \vdash \mathit{Degree}(\mathrm{s}, -) : \mathit{Type}$

For example, the type $\mathit{Degree}(\mathrm{dist}, +)$ would consist of all positive degrees on the distance scale, whereas the type $\mathit{Degree}(\mathrm{weight}, -)$ would consist of all negative degrees on the weight scale. Given two degrees on the same scale with the same polarity, we can compare their magnitudes via a comparison predicate which is unique to that scale and polarity, as follows:

(203) $\mathrm{s} : \mathit{Scale}, \mathrm{d}_1, \mathrm{d}_2 : \mathit{Degree}(\mathrm{s}, +) \vdash \mathrm{d}_1 \leq_{(\mathrm{s},+)} \mathrm{d}_2 : \mathit{Prop}$

$\mathrm{s} : \mathit{Scale}, \mathrm{d}_1, \mathrm{d}_2 : \mathit{Degree}(\mathrm{s}, -) \vdash \mathrm{d}_1 \leq_{(\mathrm{s},-)} \mathrm{d}_2 : \mathit{Prop}$

where for any scale s, the predicates $\leq_{(\mathrm{s},+)}$ and $\leq_{(\mathrm{s},-)}$ both satisfy the axioms for a total linear order[1]. Two degrees on different scales or with different polarities cannot be compared because it is impossible to form the proposition that one is greater than the other.

To express the isomorphism between positive and negative degrees on the same scale, we need a means for converting between them. This is expressed using two polymorphic functions:

[1] See `https://mathworld.wolfram.com/TotallyOrderedSet.html`

(204) $\mathrm{s} : \mathit{Scale} \vdash \mathrm{negate_s} : \mathit{Degree}(\mathrm{s}, +) \to \mathit{Degree}(\mathrm{s}, -)$

$\mathrm{s} : \mathit{Scale} \vdash \mathrm{negate_s^{-1}} : \mathit{Degree}(\mathrm{s}, -) \to \mathit{Degree}(\mathrm{s}, +)$

where $\mathrm{negate_s}$ goes from a positive s-degree to a negative s-degree, and $\mathrm{negate_s^{-1}}$ goes in the opposite direction. These are defined to be mutually inverse, as follows:

(205) $\mathrm{s} : \mathit{Scale}, \mathrm{d} : \mathit{Degree}(\mathrm{s}, +) \vdash \mathrm{d} \equiv \mathrm{negate_s^{-1}}(\mathrm{negate_s}(\mathrm{d})) : \mathit{Degree}(\mathrm{s}, +)$

$\mathrm{s} : \mathit{Scale}, \mathrm{d} : \mathit{Degree}(\mathrm{s}, -) \vdash \mathrm{d} \equiv \mathrm{negate_s}(\mathrm{negate_s^{-1}}(\mathrm{d})) : \mathit{Degree}(\mathrm{s}, -)$

That is, doing $\mathrm{negate_s}$ followed by $\mathrm{negate_s^{-1}}$ is the same as doing nothing, and vice versa. Furthermore, we must ensure that the order of negative degrees is reversed compared to positive degrees. In other words, there are two functions:

(206)
- $\mathrm{s} : \mathit{Scale}, \mathrm{d_1}, \mathrm{d_2} : \mathit{Degree}(\mathrm{s}, +) \vdash$
  $\mathrm{reverse_s} : \mathrm{d_1} \leq_{(\mathrm{s},+)} \mathrm{d_2} \to \mathrm{negate_s}(\mathrm{d_2}) \leq_{(\mathrm{s},-)} \mathrm{negate_s}(\mathrm{d_1})$
- $\mathrm{s} : \mathit{Scale}, \mathrm{d_1}, \mathrm{d_2} : \mathit{Degree}(\mathrm{s}, -) \vdash$
  $\mathrm{reverse_s^{-1}} : \mathrm{d_1} \leq_{(\mathrm{s},-)} \mathrm{d_2} \to \mathrm{negate_s^{-1}}(\mathrm{d_2}) \leq_{(\mathrm{s},+)} \mathrm{negate_s^{-1}}(\mathrm{d_1})$

That is, if $\mathrm{d_1}$ is a larger positive degree than $\mathrm{d_2}$, then one can show that $\mathrm{negate}(\mathrm{d_2})$ is a larger negative degree than $\mathrm{negate}(\mathrm{d_1})$. Likewise, if $\mathrm{d_1}$ is a larger negative degree than $\mathrm{d_2}$, one can show that $\mathrm{negate}^{-1}(\mathrm{d_2})$ is a larger positive degree than $\mathrm{negate}^{-1}(\mathrm{d_1})$.

We can treat all measure phrases such as *2 meters*, *10kg*, *50 degrees Celsius*, *3.5 seconds*, and so on, as belonging to the following type, which contains all positive degrees on all scales:

(207) $$\mathit{Degree}(+) \coloneqq \begin{bmatrix} \mathrm{s} : \mathit{Scale} \\ \mathit{Degree}(\mathrm{s}, +) \end{bmatrix}$$

The reason for thinking that measure phrases have positive polarity is that they cannot combine with negative polarity adjectives, e.g. ?*2m short*, ?*10kg light*, ?*3°C cold*, and so on. To give an example, the measure phrase *2m* would have the following interpretation:

(208) $[\![\text{2 meters}]\!] : \mathit{Degree}(+)$

$$[\![\text{2 meters}]\!] \coloneqq \begin{pmatrix} \mathrm{dist} \\ 2 \cdot m \end{pmatrix}$$

where m is some positive non-zero constant which converts the inherent units associated with the distance scale to units of meters. A speaker may be unsure as to the precise value of m, in which case they can represent the value of the resulting degree only approximately. The fact that *2 meters* has dimensions of distance explains why it cannot combine with an adjective of different dimensions as in ?*the car is 2 meters old*.

In addition to the type of all positive degrees, we can also define the type of all negative degrees:

(209) $$Degree(-) \coloneqq \begin{bmatrix} \mathrm{s} : \mathit{Scale} \\ \mathit{Degree}(\mathrm{s}, -) \end{bmatrix}$$

It is also useful to have a type for all degrees, whether positive or negative:

(210) $$Degree \coloneqq \begin{bmatrix} \mathrm{p} : \mathit{Pol} \\ \mathrm{s} : \mathit{Scale} \\ \mathit{Degree}(\mathrm{s}, \mathrm{p}) \end{bmatrix}$$

Given any positive or negative degree, we can always consider this as a general *Degree*, by inserting a + or − label. This is handled by the following functions:

(211) $$\text{as\_pos} : \mathit{Degree}(+) \to \mathit{Degree}$$

$$\text{as\_neg} : \mathit{Degree}(-) \to \mathit{Degree}$$

$$\text{as\_pos} \coloneqq \lambda \mathrm{d} : \mathit{Degree}(+) \ . \begin{pmatrix} + \\ \mathrm{d.1} \\ \mathrm{d.2} \end{pmatrix}$$

$$\text{as\_neg} \coloneqq \lambda \mathrm{d} : \mathit{Degree}(-) \ . \begin{pmatrix} - \\ \mathrm{d.1} \\ \mathrm{d.2} \end{pmatrix}$$

3.2.2 Typing gradable adjectives

This thesis will follow Kennedy's (1999) analysis of gradable adjectives as measure functions. In importing Kennedy's analysis into DTT, we must deal with the fact that a measure function like **tall** is only a partial function, applying only to those individuals which meet certain presuppositions. Recall from the previous chapter that an intersective adjective is represented by an element of the following type:

(212) $$\mathit{Ind} \to \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \mathit{Prop} \end{bmatrix}$$

A gradable adjective is represented similarly, except that instead of returning a proposition, it returns a degree which measures some aspect of the individual. Positive and negative gradable adjectives are typed as follows:

(213) $$\mathit{Gradable}(+) \coloneqq \mathit{Ind} \to \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \mathit{Degree}(+) \end{bmatrix}$$

$$\mathit{Gradable}(-) \coloneqq \mathit{Ind} \to \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \mathit{Degree}(-) \end{bmatrix}$$

As shown, a positive gradable adjective is a function from an individual x and some collection of presuppositions P(x) to a positive degree; likewise a negative gradable adjective takes individual x and some presuppositions P(x) to a negative degree. The following is an example of a positive polarity adjective:

(214) $[\![\text{tall}]\!]_{\text{up, 1st, large}} : \mathit{Gradable}(+)$

$$[\![\text{tall}]\!]_{\text{up, 1st, large}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\ \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ \textsc{up}(\mathrm{v}) \\ \textsc{1st}(\mathrm{x}, \mathrm{v}) \\ \textsc{large}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \begin{pmatrix} \text{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

This sense will be discussed in more detail in Section 3.5.2. For now, notice that the presuppositions involve a vector v, which corresponds to the vertical primary axis of the individual. The measure function takes an element of the presupposition type and returns a positive degree whose value is the magnitude of v.

It makes sense to distinguish positive from negative polarity adjectives because they are associated with a range of syntactic differences. For example, positive adjectives can occur with *un-*, e.g. *unhappy*, *unkind*, *unclear*, *uninteresting*, whereas negative adjectives cannot, e.g. **unsad*, **uncruel*, **unvague*, **unboring*. At the same time, to write expressions for degree morphemes like *more/-er*, it is also necessary to have a type for all gradable adjectives, whether positive or negative. This is defined as follows:

(215) $\mathit{Gradable} \coloneqq \mathit{Ind} \rightarrow \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \rightarrow \mathit{Degree} \end{bmatrix}$

As one would expect, a general gradable adjective is a function from an individual x and a collection of presuppositions P(x) to an arbitrary degree, which can be positive or negative. Any element of *Gradable*(+) or *Gradable*(−) can be considered as an element of *Gradable*, by extending the as_pos and as_neg functions defined in (211):

(216) $\text{as\_pos}' : \mathit{Gradable}(+) \rightarrow \mathit{Gradable}$

$$\text{as\_neg}' : \mathit{Gradable}(-) \rightarrow \mathit{Gradable}$$

$$\text{as\_pos}' \coloneqq \lambda \mathrm{G} : \mathit{Gradable}(+)\ .$$
$$\left(\lambda \mathrm{x} : \mathit{Ind}\ .\ \begin{pmatrix} \mathrm{G}(\mathrm{x}).1 \\ \lambda \mathrm{p} : \mathrm{G}(\mathrm{x}).1(\text{as\_pos}(\mathrm{G}(\mathrm{x}).2(\mathrm{p}))) \end{pmatrix}\right)$$

$$\text{as\_neg}' \coloneqq \lambda \mathrm{G} : \mathit{Gradable}(-)\ .$$
$$\left(\lambda \mathrm{x} : \mathit{Ind}\ .\ \begin{pmatrix} \mathrm{G}(\mathrm{x}).1 \\ \lambda \mathrm{p} : \mathrm{G}(\mathrm{x}).1(\text{as\_neg}(\mathrm{G}(\mathrm{x}).2(\mathrm{p}))) \end{pmatrix}\right)$$

That is, one explicitly labels the output degree as a positive degree in the case of $Gradable(+)$ or a negative degree in the case of $Gradable(-)$.

Like other kinds of semantic objects, gradable adjectives form networks. Suppose we have two terms $G, H : Gradable(+)$, that is two positive gradable adjectives with different presupposition types. Given some arbitrary individual x, we can visualize the content of $G(x)$ and $H(x)$ as follows:

(217)
$$\begin{tikzcd} \mathrm{G(x).1} \arrow[dr, "\mathrm{G(x).2}"'] & & \mathrm{H(x).1} \arrow[dl, "\mathrm{H(x).2}"] \\ & \mathit{Degree}(+) & \end{tikzcd}$$

That is, both $G(x)$ and $H(x)$ provide a presupposition type and a function from this type to positive degrees. As this diagram suggests, an arrow from $G(x)$ to $H(x)$ will consist of a function $f(x) : G.1 \to H.1$ which completes the triangle and makes it commute, as follows:

(218)
$$\begin{tikzcd} \mathrm{G(x).1} \arrow[rr, "\mathrm{f(x)}"] \arrow[dr, "\mathrm{G(x).2}"'] & \checkmark & \mathrm{H(x).1} \arrow[dl, "\mathrm{H(x).2}"] \\ & \mathit{Degree}(+) & \end{tikzcd}$$

The data in (218) can be packaged into the following type, abstracting over the individual x:

(219) $$\mathrm{G} \to_{Gradable(+)} \mathrm{H} \coloneqq \prod_{\mathrm{x}:Ind} \begin{bmatrix} \mathrm{f} : \mathrm{G(x).1} \to \mathrm{H(x).1} \\ \prod_{\mathrm{p:G(x).1}} \mathrm{G(x).2(p)} =_{Degree(+)} \mathrm{H(x).2(f(p))} \end{bmatrix}$$

where the first component contains the function between presupposition types and the second component is the commuting condition. Arrows between negative gradable adjectives and general gradable adjectives are defined analogously:

(220) $$\mathrm{G} \to_{Gradable(-)} \mathrm{H} \coloneqq \prod_{\mathrm{x}:Ind} \begin{bmatrix} \mathrm{f} : \mathrm{G(x).1} \to \mathrm{H(x).1} \\ \prod_{\mathrm{p:G(x).1}} \mathrm{G(x).2(p)} =_{Degree(-)} \mathrm{H(x).2(f(p))} \end{bmatrix}$$

$$\mathrm{G} \to_{Gradable} \mathrm{H} \coloneqq \prod_{\mathrm{x}:Ind} \begin{bmatrix} \mathrm{f} : \mathrm{G(x).1} \to \mathrm{H(x).1} \\ \prod_{\mathrm{p:G(x).1}} \mathrm{G(x).2(p)} =_{Degree} \mathrm{H(x).2(f(p))} \end{bmatrix}$$

The following example is intended to illustrate what an arrow between gradable adjectives looks like:

(221) $$[\![\mathrm{tall}]\!]^{\mathrm{vert,\,1st,\,large}}_{\mathrm{up,\,1st,\,large}} : [\![\mathrm{tall}]\!]_{\mathrm{up,\,1st,\,large}} \to_{Gradable(+)} [\![\mathrm{tall}]\!]_{\mathrm{vert,\,up,\,large}}$$

$$[\![\text{tall}]\!]^{\text{vert, 1st, large}}_{\text{up, 1st, large}} \coloneqq \lambda \text{x} : \mathit{Ind} \ . \begin{pmatrix} \lambda \text{p} : [\![\text{tall}]\!]_{\text{up, 1st, large}}.1 \ . \begin{pmatrix} \text{p.1} \\ \text{inl(p.2)} \\ \text{p.3} \\ \text{p.4} \end{pmatrix} \\ \lambda \text{p} : [\![\text{tall}]\!]_{\text{up, 1st, large}}.1 \ . \ \text{refl}_{\begin{pmatrix} \text{dist} \\ \|\text{p.1}\| \end{pmatrix}} \end{pmatrix}$$

As shown, $[\![\text{tall}]\!]^{\text{vert, 1st, large}}_{\text{up, 1st, large}}$ goes from the sense $[\![\text{tall}]\!]_{\text{up, 1st, large}}$ which was given in (214) to a weaker sense $[\![\text{tall}]\!]_{\text{vert, 1st, large}}$. The presuppositions are altered by injecting one of the components on the left. Since both $[\![\text{tall}]\!]_{\text{up, 1st, large}}$ and $[\![\text{tall}]\!]_{\text{vert, 1st, large}}$ return the degree:

(222) $\begin{pmatrix} \text{dist} \\ \|\text{p.1}\| \end{pmatrix}$

the commuting condition is witnessed by the reflexivity proof associated with this term.

Given an arrow connecting two elements of *Gradable*(+) or *Gradable*(−), this can be converted into an arrow between two elements of *Gradable*. For instance, suppose we have two positive gradable adjectives G, H : *Gradable*(+) connected by an arrow:

(223) $\alpha : \text{G} \rightarrow_{\mathit{Gradable}(+)} \text{H}$

This can be lifted to an arrow:

(224) $\text{as\_pos}'(\alpha) : \text{as\_pos}'(\text{G}) \rightarrow_{\mathit{Gradable}} \text{as\_pos}'(\text{H})$

as follows:

(225) $\text{as\_pos}'(\alpha) \coloneqq \lambda \text{x} : \mathit{Ind} \ .$

$$\text{let G}' \coloneqq \text{as\_pos}'(\text{G})(\text{x}) \text{ in}$$
$$\begin{pmatrix} \lambda \text{p} : \text{G}'.1 \ . \ \alpha(\text{x}).1(\text{p}) \\ \lambda \text{p} : \text{G}'.1 \ . \ \text{ap}_{\text{as\_pos}}(\alpha(\text{x}).2(\text{p}))) \end{pmatrix}$$

As shown, the first component of the arrow remains unchanged, and the second component is altered using the function ap described in Section 2.4.3, which lifts identities under functions.

As one would expect, arrows between (positive, negative or general) gradable adjectives can be composed. Given three gradable adjectives G, H, I : *Gradable* and two arrows:

(226) $\alpha : G \rightarrow_{\mathit{Gradable}} H$

$\beta : H \rightarrow_{\mathit{Gradable}} I$

The composition is given by:

(227) $\beta \circ_{Gradable} \alpha : G \to_{Gradable} I$

$$\beta \circ_{Gradable} \alpha \coloneqq \lambda \mathrm{x} : Ind \ . \begin{pmatrix} \mathrm{f} \coloneqq \beta(\mathrm{x}).1 \circ \alpha(\mathrm{x}).1 \\ \lambda \mathrm{p} : \mathrm{G}.1 \ . \ \alpha(\mathrm{x}).2(\mathrm{p}) \bullet \beta(\mathrm{x}).2(\mathrm{f}(\mathrm{p})) \end{pmatrix}$$

As shown, the presupposition maps are composed using ordinary function composition ($\circ$) and the commutativity proofs are composed using the composition of identities operation ($\bullet$) which was described in Section 2.4.3.

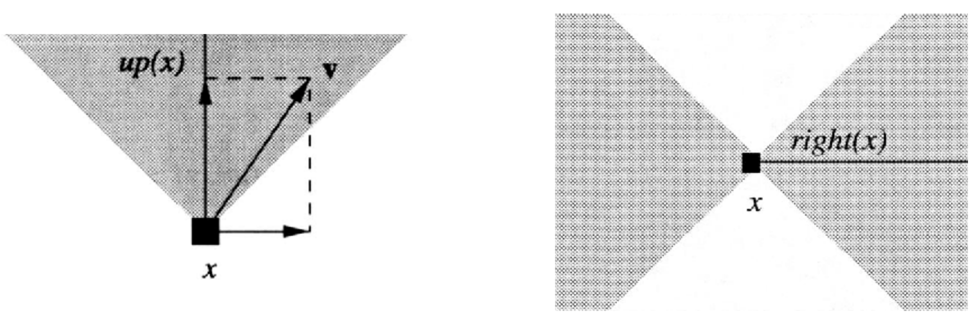
3.3 Vector Space Semantics

3.3.1 Introduction to vector space semantics

The idea that spatial language can be analysed in terms of vectors was first proposed by John O'Keefe in his description of English spatial prepositions (O'Keefe & Nadel 1978, O'Keefe 1996). He argued that the role of prepositions is to locate objects with respect to other objects within a *cognitive map* – an allocentric cognitive representation of the environment. The relative position of an object is given by a vector specifying its distance and direction from some reference object. Spatial prepositions denote sets of position vectors. For example, *above* denotes the set of vectors based at the reference object which point upwards, *behind* denotes the set of vectors based at the reference object which point backwards; and so on. O'Keefe links his vector analysis of prepositions to the representation of space in the hippocampus, proposing that vectors are implemented in terms of *place cells* – cells which fire when an animal visits a certain region in its environment. Each preposition would be associated with a pattern of place cell activation which defines a spatial field surrounding the reference object.

Zwarts (1997) and Zwarts & Winter (2000) have developed a model-theoretic implementation of O'Keefe's ideas. They propose an analysis of locative prepositions in which the position of the figure (located object) relative to the ground (reference object) is represented by a 'located vector'. A located vector is defined as a pair of ordinary vectors $(\mathbf{u}, \mathbf{v})$, where $\mathbf{u}$ specifies the location of the base point and $\mathbf{v}$ specifies the vector's direction and magnitude. A prepositional phrase such as *above the house* or *beside the car* denotes a set of located vectors based at the reference object defining a 'search region' in which the figure might be found (see Figure 3.1). The correct regions for topological prepositions (*in*, *on*, *at*, etc.) can be defined solely based on the region occupied by the object, whereas the correct regions for projective prepositions (*above/below*, *in front of/behind* and *beside*) require the use of special 'perspective functions' called up, $front$ and $right$, which associate individuals with unit vectors in their axial directions.

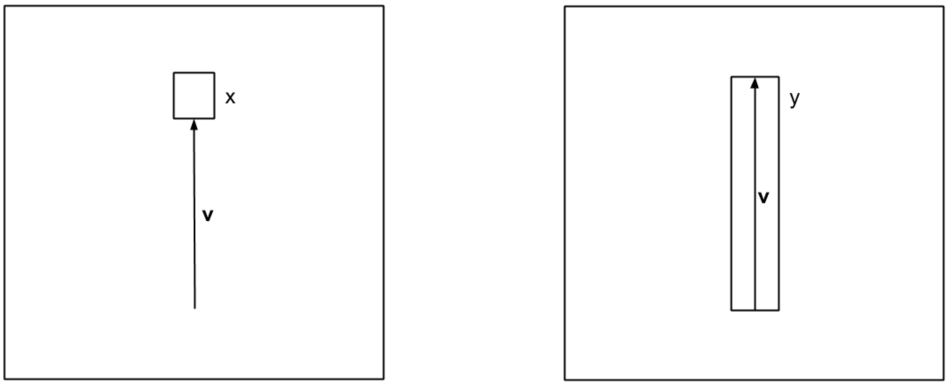
Zwarts (2003) argues that vectors should be used not only to describe the relative position of objects, but also to represent parts and axes. This idea was already



(a) *above x* **(b)** *beside x*

Figure 3.1: Locative prepositions can be analyzed as denoting sets of vectors based at the ground object, from Zwarts & Winter (2000).

present in theories of 3D object representation such as Marr's (1982) 3D model, which uses vectors to represent the orientation and size of both whole objects and parts. The rational for treating axes as vectors is that, like relative positions, they are characterised by the two pieces of information that define a vector: magnitude and direction. As shown in Figure 2, one and the same vector **v** can be used to represent either the position of some object with respect to a contextually chosen origin, or the major axis of an object. One advantage of using vectors to describe both positions and axes is that it allows us to explain parallelisms between the domain of place and the domain of size or dimension. For instance, the English spatial adjectives *high*, *low* and *deep* can measure either a position vector or an axis vector, depending on context. The vector-based analysis correctly predicts that both senses should be able to combine with distance measure phrases, since they both involve degrees on the scale of distance.



(a) Position vector. **(b)** Axis vector.

Figure 3.2: based on Zwarts (2003).

In addition to positions and axes, vectors can also be used to represent *paths*. A path is a linearly ordered sequence of vectors, with a defined start point and endpoint. Directional prepositional phrases such as *to the house*, *from the supermarket*, *across the field*, and so on, can be analysed as denoting sets of paths, with constraints on their source, goal or route. See Zwarts (2005) for a fully worked-out version of this idea, as well as a path-based explanation of prepositional aspect (e.g. the difference between *to the house* and *towards the house*). In addition to describing directional prepositions, paths are also useful for modelling the axes of objects which curve or bend and so cannot be described by a single vector, such as the axis of a river or piece of string (Zwarts 2003). This will be useful for the analysis of length, which can describe a curve as well as a straight line. Once we have paths, it also makes sense to think about other geometric objects, such as *ribbons*, which are like paths but with an additional width at every point in the path. Ribbons turn out to be important to the meaning of *wide* and *narrow*, as I shall show in Section 3.7.

3.3.2 Formalizing vectors

In order to introduce vectors, one must first introduce a field of numbers. Henceforth, I shall assume a type $Real$, elements of which are real numbers, together with the usual operations of addition $\_ + \_ : Real \times Real \to Real$, and multiplication $\_ \cdot \_ : Real \times Real \to Real$. Moreover, for any two real numbers $x, y : Real$, there is a type $x \leq y$ corresponding to the proposition 'x is less than or equal to y'. The structure of the real numbers is then given by axioms involving $+$, $\cdot$ and $\leq$. In brief, $(Real, +, \cdot)$ must satisfy the *field axioms*[1]; $(Real, \leq)$ must be a total order; and $+$ and $\cdot$ must be compatible with $\leq$, meaning that for all $x, y, z : Real$:

(228)
- if $x \leq y$ then $x + z \leq y + z$
- if $0 \leq x$ and $0 \leq y$ then $0 \leq x \cdot y$

Finally, there is the completeness axiom which distinguishes the real numbers from the rational numbers. This axiom states that every non-empty subset of $Real$ which is bounded above has a least upper bound in $Real$. For example, the subset $\sum_{r:Real} r^2 > 2$ has the least upper bound $\sqrt{2} : Real$. The conjunction of all these axioms is sufficient to define the real number system up to isomorphism.

All the vectors in this chapter belong to the same vector space, which I shall simply call $Vector$. The preferred interpretation of vectors is as arrows in ordinary three-dimensional Euclidean space, where each arrow has a direction and magnitude but no fixed location. In the literature on spatial language, it is sometimes assumed that the concept of vector is subordinate to the concept of reference frame or coordinate system. For instance, in their discussion of spatial language, Carlson et al. (2003) write that spatial templates, by which they mean configurations of vectors, are "tied to, and perhaps defined by, reference frames" (2003, p. 7). However, mathematically speaking it is only the components or coordinates of a vector that are dependent on a choice of coordinates, the vector itself being an invariant object which survives a change of coordinates. This is most apparent in physics, where vectors are used to represent physical quantities like velocity and acceleration. The velocity of a car may be described differently in different coordinate systems, but all these descriptions correspond to the same physical fact.

The type $Vector$ comes equipped with two operations, addition $\_ + \_ : Vector \times Vector \to Vector$, and scalar multiplication $\_ \cdot \_ : Real \times Vector \to Vector$. There is a unique vector $\mathbf{0} : Vector$ which plays the role of additive identity. Finally, a function $-\_ : Vector \to Vector$ sends each vector to its additive inverse. These operations are required to satisfy the following axioms, for all $\mathbf{u}, \mathbf{v}, \mathbf{w} : Vector$ and $a, b : Real$:

(229)
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- $\mathbf{v} + \mathbf{0} = \mathbf{v}$
- $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$

[1] `https://mathworld.wolfram.com/FieldAxioms.html`

- $a \cdot (b \cdot \mathbf{v}) = (a \cdot b) \cdot \mathbf{v}$
- $1 \cdot \mathbf{v} = \mathbf{v}$
- $a \cdot (\mathbf{u} + \mathbf{v}) = a \cdot \mathbf{u} + a \cdot \mathbf{v}$
- $(a + b) \cdot \mathbf{v} = a \cdot \mathbf{v} + b \cdot \mathbf{v}$

In addition, I shall assume an inner product between vectors $\langle \_, \_ \rangle : Vector \to Real$. The inner product takes two vectors and returns a scalar measuring their 'degree of separation'. It is defined through the following axioms for all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w} : Vector$ and $a : Real$:

(230)
- $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle$
- $\langle a \cdot \mathbf{u}, \mathbf{v} \rangle = a \cdot \langle \mathbf{u}, \mathbf{v} \rangle$
- $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$
- $\langle \mathbf{v}, \mathbf{v} \rangle \geq 0$
- $\langle \mathbf{v}, \mathbf{v} \rangle = 0$ iff $\mathbf{v} = \mathbf{0}$

Geometrically, the inner product between two vectors $\langle \mathbf{u}, \mathbf{v} \rangle$ can be thought of as the quantity:

(231) $\|\mathbf{u}\| \|\mathbf{v}\| \cos \theta$

that is, the product of the length of $\mathbf{u}$, the length of $\mathbf{v}$, and the cosine of the angle θ between the two vectors. Strictly speaking, however, the notions of length and angle are defined in terms of the inner product, not vice versa. The *length* of a vector $\mathbf{v}$ is defined as:

(232) $\|\mathbf{v}\| \coloneqq \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$

Because $\langle \mathbf{v}, \mathbf{v} \rangle$ is always positive, this is guaranteed to be a positive real number, as we would expect from a notion of length. The angle between two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined through its cosine:

(233) $\cos \theta \coloneqq \dfrac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\| \|\mathbf{v}\|}$

It follows that $\mathbf{u}$ and $\mathbf{v}$ are orthogonal iff $\langle \mathbf{u}, \mathbf{v} \rangle = 0$.

3.3.3 Spatial primitives

Throughout this chapter, I will make use of a collection of primitive predicates relating vectors and individuals. The intention is that these primitives should not only be useful for spatial adjectives, but also locative and directional prepositions, axial part terms, postural verbs, and so on: they correspond to fundamental concepts in the spatial domain. One example is the predicate $\textsc{axis}(\mathrm{x}, \mathrm{v})$, which encodes the idea 'v is an axis of x'. The proofs of primitive predicates provide the interface between type theory and perception. For example, a proof of $\textsc{axis}(\mathrm{x}, \mathrm{v})$ would be a piece of visual information showing that v is an axis of x. Such a proof would consist of whatever data is used to encode the position or axis of an object in the brain. The role of primitive predicates like $\textsc{axis}(x, v)$ is similar to that of *image schemas* in cognitive semantics: they encode basic concepts which are meant to be directly grounded in sensory/motor experiences. I adopt the convention that primitive predicates are written in SMALL CAPS.

Spatial primitives are assumed to be part of spatial cognition, and hence universal across different languages. Every language somehow expresses basic notions of axis, path, surface, observer, and so on, although the particular way in which these primitives get assembled into lexical and grammatical items can differ cross-linguistically. For example, English *thick* covers both the minimal dimension of a surface (e.g. *thick plate*) and the minimal dimension of a cylinder (e.g. *thick stick*), whereas Japanese distinguishes lexically between surface thickness and tubular thickness, which are encoded by *atsui* and *futoi* respectively (Shimotori 2013). To give another example, English has two words for the vertical axis – *high* and *tall* – whereas Italian combines both into a single lexical item, *alto* (Goy 2002). The primitives which appear in English *high* and *tall* also appear in *alto*, but the particular combination is different.

The following is a list of the primitive predicates assumed in this chapter, together with their intended interpretations:

Predicate	Intended Interpretation
$v : Vector$ $\vdash \textsc{up}(v) : Prop$	'v is pointed upwards'
$v : Vector$ $\vdash \textsc{down}(v) : Prop$	'v is pointed downwards'
$v : Vector$ $\vdash \textsc{horz}(v) : Prop$	'v is horizontal'
$v : Vector$ $\vdash \textsc{grnd}(v) : Prop$	'v describes a point on the ground'
$v : Vector$ $\vdash \textsc{large}(v) : Prop$	'v is at least as large as the average human height'
$x : Ind,\ v : Vector$ $\vdash \textsc{axis}(x, v) : Prop$	'v is an axis of x'
$x : Ind,\ b : Bivector$ $\vdash \textsc{axis}(x, b) : Prop$	'b is a 2D integrated axis of x'
$x : Ind,\ u, v : Vector$ $\vdash \textsc{axis}(x, u, v) : Prop$	'v is an axis of x, based at the point described by u'
$x : Ind,\ v : Vector$ $\vdash \textsc{top}(x, v) : Prop$	'v is the inherently vertical axis of x'

$x : Ind,\ v : Vector$ $\vdash \textsc{1st}(x, v) : Prop$	'v is the primary axis of x'
$x : Ind,\ v, : Vector$ $\vdash \textsc{2nd}(x, v) : Prop$	'v is a secondary axis of x'
$x : Ind,\ v : Vector$ $\vdash \textsc{intrnl}(x, v) : Prop$	'v is an internal axis of x'
$x : Ind,\ u, v : Vector$ $\vdash \textsc{posn}(x, u, v) : Prop$	'v is the position vector of x with respect to the point described by u'
$x : Ind,\ p : Path$ $\vdash \textsc{path}(x, p) : Prop$	'the shape of x is approximated by the path p'
$x : Ind,\ r : Vector,\ p : Path$ $\vdash \textsc{path}(x,\ r,\ p) : Prop$	'the shape of x is approximated by a path of vectors p, centered at the point r'
$x : Ind,\ r : Ribbon,\ w : (0, \infty)$ $\vdash \textsc{ribbon}(x, r, w) : Prop$	'the shape of x is approximated by the ribbon r, with average width w'
$x : Ind,\ p, t, f, r : Vector$ $\vdash \textsc{obs}(x, p,\ t, f, r) : Prop$	'x has a canonical observer based at p with coordinate vectors (t, f, r)'
$x : Ind,\ p : Path,\ t : (0, \infty)$ $\vdash \textsc{cyldr}(x, p, t) : Prop$	'x is approximated by a cylinder with axis p and average thickness t'
$x : Ind,\ s : Surface,\ t : (0, \infty)$ $\vdash \textsc{surf}(x, s, t) : Prop$	'x is approximated by the surface s with average thickness t'
$x : Ind,\ S : Vector{\to}Prop,\ t : (0, \infty)$ $\vdash \textsc{skel}(x, S, t) : Prop_1$	'the shape of x is approximated by a skeleton S, with average thickness t'

Table 3.1

Further description of each of these predicate types is given in the discussion of particular adjectives. This includes the definitions of the types *Bivector*, *Path*, *Ribbon* and *Surface*. The type (0, ∞) contains all positive real numbers excluding 0 and is an abbreviation for $\sum_{r:Real} r > 0$.

I shall not attempt to give a detailed justification for this particular set of primitives, since I do not regard it as definitive. The list in Table 3.1 is not an *a priori* deduction, but consists of the basic notions which I have found necessary in order to describe the different senses of spatial adjectives. Without a basic choice of primitives, it is impossible for a decompositional analysis to get off the ground, but it is important to remember that the fundamental vocabulary might need to be altered or updated in light of new observations or theoretical developments. An important point is that – unlike the featural approaches to gradable adjectives discussed in Section 1.3 – there is not a straightforward correspondence between primitives and spatial adjective meanings. There is not single primitive which encodes the meaning of an adjective like *wide*, rather *wide* involves many different primitives, some of which are also involved in other adjectives. The list in Table 3.1, although designed with spatial adjectives in mind, is not circular in the sense that it assumes from the beginning the very concepts it attempts to explain.

I do not mean to rule out a deeper analysis in which some of the predicates in the table are given a more thorough analysis in terms of other, more basic primitives. For example, UP(v), DOWN(v) and HORZ(v) could be described in terms of a primitive unit vector **up**, together with the inner product. The predicates AXIS(x, u, v), VERT(x, v), 1ST(x, v), 2ND(x, u, v) and INTRNL(x, v) could all be defined in terms of AXIS(x, v) together with some additional conditions. RIBBON(x, r, w) could be defined in terms of PATH(x, p) and some additional information. In each case, I have found that attempting to decompose further yields an analysis which is more complicated and less intuitive. Moreover, further decomposition introduces a greater degree of speculation, since we do not currently understand how the brain encodes spatial information.

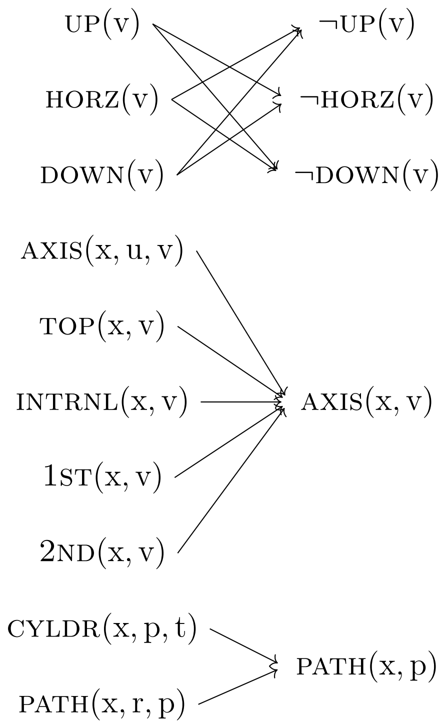


Figure 3.3: Some implications between predicates in Table 3.1

Although I shall not attempt to further decompose the primitives in Table 3.1, it is important to acknowledge some basic connections between them. Some of these connections are illustrated in Figure 3.3. This can be considered a kind of 'sub-lexical' network, since it does not correspond to the meaning of a word. Like the links in a lexical network, these implications are internalized as terms of the type system. For example, the implication from 1ST(x, v) to AXIS(x, v) corresponds to a polymorphic function of type $\prod_{\mathrm{x}:Ind,\mathrm{v}:Vector}$ 1ST(x, v) → AXIS(x, v). In addition to the simple implications shown in Figure 3.3 there are also some more complex inferences which

I have not drawn. For example, if an individual x : *Ind* has a canonical observer, then an axis v of x which is parallel to the observer's intrinsic vertical axis will be an intrinsic vertical axis of the object, in other words it will satisfy $\textsc{top}(x, v)$.

Now that we have the theoretical tools of degrees and vectors, and understand how these are implemented in the type system, we can turn to an analysis of spatial adjectives themselves. Each section henceforth is dedicated to a particular spatial adjective network. I begin each section with a brief introduction to the various senses of the adjective, together with some previous attempts to describe its meaning. I then present a detailed analysis of the spatial senses, using the primitives given in Table 3.1. As discussed in Section 1.2.2, the following criteria are used for distinguishing senses:

(234) a. For each sense in the network, one must be able to find a situation or sentence which is restricted to that sense.
b. If two senses are mutually compatible and typically occur together, then they have a common meet.
c. If two senses can be coordinated without giving rise to zeugma, then they have a common join.

3.4 Height

3.4.1 Introduction to height

A basic observation about the adjectives *high* and *low* is that they are ambiguous between a *positional* sense, which refers to vertical distance from the ground, and a *dimensional* sense, which refers to vertical extension (Clark 1973, Lyons 1977, Dirven & Taylor 1986). For example, a *high bird's nest* can mean either a bird's nest which is a significant distance from the ground or a bird's nest which has a larger than usual vertical extent. The same alternation occurs in other languages. For example, Lafrenz (1983) writes that German *hoch* 'high/tall' can describe either the distance of an object from the ground, as in *hohe Decke* 'high ceiling', or the distance of an object from base to tip, as in *hoher Mast* 'high mast'. The same alternation has been described for Swedish *hög* 'high/tall' (Vogel 2004), Italian *alto* 'high/tall' (Goy 2002), Polish *wysoki* 'high/tall' (Linde-Usiekniewicz 2002), Japanese *takai* 'high/tall' (Shimotori 2013), and Yucatec Maya *ka'nal* 'high/tall' (Stolz 1996).

English is unusual in that it has two positive adjectives which involve the vertical axis – *high* and *tall*. In most other languages, these are covered by the same lexical item. Whereas *high* is ambiguous between position and dimension, *tall* has only a dimensional sense (e.g. *tall bird's nest* has only a dimensional interpretation). Another difference is that the dimension referred to by *high* must be vertical in the actual environment, whereas *tall* can describe an axis which is vertical in the object's canonical orientation. For example, a toppled lamppost can still be described as *tall*, but the use of *high* in this situation sounds strange. What is more, *tall* typically applies to an axis which is salient in some way, being either the primary axis of an object (e.g. *tall wine glass*), or large in comparison to a human being (e.g. *tall fence*), whereas *high* has no such restriction. For these reasons, *high* and *tall* are described in separate sections – see Section 3.9 for a description of *tall*.

High and *low* are usually described as measuring vertical displacement with respect to some reference point or plane (Clark 1973, Fillmore 1997, Lyons 1977, Dirven & Taylor 1986). In the dimensional sense, the reference point is the base (lowermost part) of the object, whereas in the positional sense the reference point is a location on the ground directly below the object. What counts as ground level can vary depending on context – for example, the positional height of an object in a room is evaluated with respect to the floor, not with respect to the surface of the Earth. What counts as an upwards direction can also be contextually determined – for instance, a page or other 2D surface sets up a local coordinate system, with its own vertical and horizontal axes, so a figure can be described as *high/low on the page* even when the sheet of paper itself is horizontal (Vogel 2004). To have a positional height, an object must be located above the contextually determined ground. An object which is located *below* the ground is thought of as having a *depth* rather than a height – for example, the distance from the ground to an ore deposit is a depth not a height, since it is directed downwards.

An interesting observation which has consequences for the structure of the height network is that positional and dimensional height can be conceptualized as the same. For example, the most typical interpretation of the sentence *the* *bird is higher than the tree*, is one in which the positional height of the bird is compared to the dimensional height of the tree. For this to be possible, the tree must be based on the ground, so that its dimensional height is evaluated with respect to the same reference plane as the bird's positional height. In this way, the concept of being based on the ground establishes a kind of semantic bridge between positional and dimensional height, which I refer to as 'elevational' height. Elevetional height is the join of positional height and a form of dimensional height in which the object is based on the ground.

In addition to their various spatial senses, *high* and *low* also have a wide range of non-spatial senses. For instance, they can be used to describe pitch (*high/low note*), social status (*high/low rank*), emotion (*to feel high/low*), and quantity (*high number*). What all these senses seem to have in common is the idea of elevation above some baseline level. In the pitch domain, the baseline is some very low reference pitch; in the domain of social status, the baseline is the bottom of the social hierarchy; in emotion, the baseline is 'rock bottom', and so on. Nevertheless, none of these senses can be coordinated with spatial senses without giving rise to zeugma: for instance, it sounds strange to say ?*John's balcony is high and so is his voice*, or ?*John's balcony is as high as his social standing.* I therefore take the non-spatial senses of *high* to be disconnected from the spatial senses.

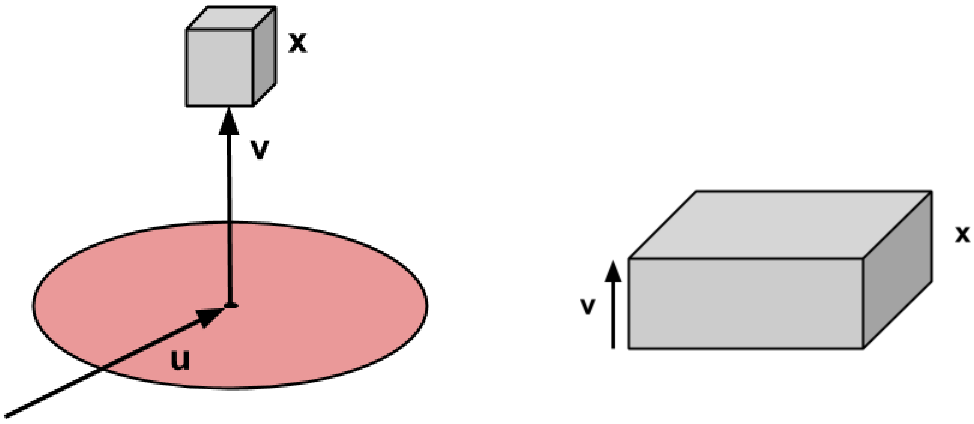
3.4.2 The high network

Let us begin with the positional sense of *high*. Recall that a positive polarity gradable adjective is formalized as an element of type *Gradable*(+), which was defined in (213). The positional sense can be written as follows:

(235) $$[\![\mathrm{high}]\!]_{\mathrm{posn}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\ \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{u} : \mathit{Vector} \\ \mathrm{v} : \mathit{Vector} \\ \textsc{grnd}(\mathrm{u}) \\ \textsc{posn}(\mathrm{x}, \mathrm{u}, \mathrm{v}) \\ \textsc{up}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

As required, it takes an individual and returns a presupposition type together with a map from that type to positive degrees. The presuppositions are listed below, in the order in which they appear:

(236)
1. there exists some vector u
2. there exists some vector v
3. u describes a point on the ground
4. the position of x with respect to u is v
5. v is directed upwards



(a) Positional height: u describes a point on the ground, v is the position of x with respect to u.

(b) Dimensional height: v is an upwards axis of x.

Figure 3.4

Given a context in which these presuppositions are satisfied, the measure function extracts the magnitude of the v vector, which is a positive degree on the scale of distance. Notice that the vector u specifies the start point of v, but there is no information about where u itself originates. The origin of u is intentionally left open – it might be based at the speaker, at some other observer, or at some contextually understood location in the environment. The choice of origin for u can be safely ignored, because nothing depends on it, the relevant quantity being the magnitude of v. For an illustration of a positional height situation, see Figure 3.4a.

Dimensional height is similar to positional height, except that the vector v represents an axis of x rather than its position. This can be formalized as follows:

(237) $$[\![\text{high}]\!]_{\text{dim}} := \lambda \text{x} : \mathit{Ind} \;.\; \begin{pmatrix} \text{P} := \begin{bmatrix} \text{v} : \mathit{Vector} \\ \textsc{axis}(\text{x}, \text{v}) \\ \textsc{up}(\text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P} \;.\; \begin{pmatrix} \text{dist} \\ \|\text{p}.1\| \end{pmatrix} \end{pmatrix}$$

As shown, dimensional height presupposes that:

(238)
1. there exists some vector v
2. v is an axis of x
3. v is directed upwards

Given a context which meets these requirements, the measure function extracts the magnitude of the v axis as before. Note that condition (2) rules out axes which are vertically oriented but directed downwards instead of upwards. This seems intuitively correct – for instance, the major axis of a stalactite cannot be described as a height, since it is pointed down rather than up. For an illustration of a dimensional height situation, see Figure 3.4b.

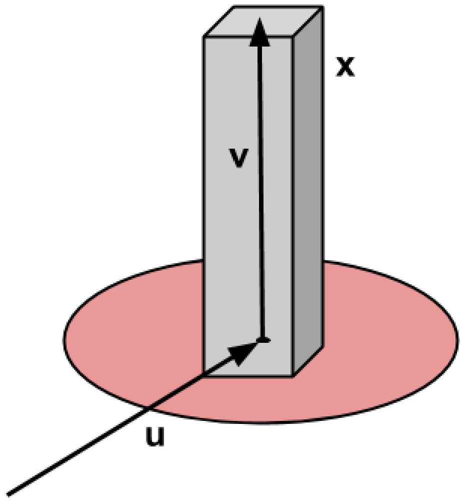


Figure 3.5: The sense $[\![\text{high}]\!]_{\text{dim, grnd}}$: u points to a location on the ground, and v is an upwards-directed axis of x based at u.

As discussed in the previous subsection, positional and dimensional height are connected through the idea of being based on the ground. In a sentence like *the bird is higher than the tree*, the positional height of the bird is compared to the dimensional height of the tree, with the understanding that the tree is based at ground level. We can therefore identify a stronger, more prototypical version of $[\![\text{high}]\!]_{\text{dim}}$ in which the vector v is not only an axis of the object, but is also based on the ground. This is encoded using the predicate type $\textsc{axis}(\mathrm{x}, \mathrm{u}, \mathrm{v})$, meaning 'v is an axis of x, based at u', in combination with $\textsc{grnd}(\mathrm{u})$, meaning 'u describes a position on the ground':

$$(239) \quad [\![\text{high}]\!]_{\text{dim, grnd}} \coloneqq \lambda \mathrm{x} : \mathit{Ind} \; . \left(\begin{array}{l} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{u} : \mathit{Vector} \\ \mathrm{v} : \mathit{Vector} \\ \textsc{grnd}(\mathrm{u}) \\ \textsc{axis}(\mathrm{x}, \mathrm{u}, \mathrm{v}) \\ \textsc{up}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \; . \begin{pmatrix} \text{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{array} \right)$$

As shown, $[\![\text{high}]\!]_{\text{dim, grnd}}$ is almost identical to $[\![\text{high}]\!]_{\text{posn}}$, the only difference being that v now represents an axis of x based at u, rather than the position of x with respect to u. See Figure 3.5 for an illustration.

Now that we have the two senses $[\![\text{high}]\!]_{\text{posn}}$ and $[\![\text{high}]\!]_{\text{dim, grnd}}$, it is easy to see that they are related through the common notion of ground level. This is what allows them to be coordinated in a sentence like *the bird is higher than the tree*. The join of these two senses, which I refer to as the 'elevation' sense, is given below:

$$(240) \quad [\![\text{high}]\!]_{\text{elev}} \coloneqq \lambda \text{x} : \mathit{Ind}\ .\ \left(\begin{array}{l} \text{P} \coloneqq \begin{bmatrix} \text{u} : \mathit{Vector} \\ \text{v} : \mathit{Vector} \\ \textsc{grnd}(\text{u}) \\ \textsc{posn}(\text{x}, \text{u}, \text{v}) + \textsc{axis}(\text{x}, \text{u}, \text{v}) \\ \textsc{up}(\text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P}\ .\ \begin{pmatrix} \text{dist} \\ \|\text{p}.1\| \end{pmatrix} \end{array} \right)$$

As shown, $[\![\text{high}]\!]_{\text{elev}}$ requires either that v represents the position of x with respect to u, as in $[\![\text{high}]\!]_{\text{posn}}$, or that v represents the axis of x based as u, as in $[\![\text{high}]\!]_{\text{dim, grnd}}$. It therefore covers both situations. In *the bird is higher than the tree*, the height of the bird satisfies $\textsc{posn}(\text{x}, \text{u}, \text{v})$, whereas the height of the tree satisfies $\textsc{axis}(\text{x}, \text{u}, \text{v})$.

The four senses of *high* defined above form the simple lexical network shown in Figure 3.6. It is important to bear in mind that this is not the complete lexical network for *high*, since it omits non-spatial senses. Notice that the *high* network is not a straightforward radial network of the sort often seen in cognitive linguistics, since it has more than one prototype and contains joins in addition to meets. The senses $[\![\text{high}]\!]_{\text{posn}}$ and $[\![\text{high}]\!]_{\text{dim, grnd}}$ have an abstract similarity, which is represented by their join; whereas the senses $[\![\text{high}]\!]_{\text{elev}}$ and $[\![\text{high}]\!]_{\text{dim}}$ are derived from the same prototype, represented by their meet.

$$\begin{tikzcd}
{[\![\text{high}]\!]_{\text{posn}}} \arrow[dr] & \\
& {[\![\text{high}]\!]_{\text{elev}}} \\
{[\![\text{high}]\!]_{\text{dim, grnd}}} \arrow[ur] \arrow[dr] & \\
& {[\![\text{high}]\!]_{\text{dim}}}
\end{tikzcd}$$

Figure 3.6: The lexical network for *high*, excluding non-spatial senses.

Given knowledge of how the various senses of *high* are implemented, it should be easy to see how the arrows are implemented. Recall that given two positive gradable adjectives $\text{G}, \text{H} : \mathit{Gradable}(+)$, an arrow between them is of type $H \rightarrow_{\mathit{Gradable}(+)} G$, which was defined in (219). Consider the arrow $[\![\text{high}]\!]^{\text{elev}}_{\text{posn}}$, which goes from $[\![\text{high}]\!]_{\text{posn}}$ to $[\![\text{high}]\!]_{\text{elev}}$. This is implemented as follows:

$$(241) \quad [\![\text{high}]\!]^{\text{elev}}_{\text{posn}} : [\![\text{high}]\!]_{\text{posn}} \rightarrow_{\mathit{Gradable}(+)} [\![\text{high}]\!]_{\text{elev}}(\text{x})$$

$$[\![\text{high}]\!]^{\text{elev}}_{\text{posn}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\ \begin{pmatrix} \lambda \mathrm{p} : [\![\text{high}]\!]_{\text{posn}}(\mathrm{x}).1\ .\ \begin{pmatrix} \mathrm{p}.1 \\ \mathrm{p}.2 \\ \mathrm{p}.3 \\ \mathrm{inl}(\mathrm{p}.4) \\ \mathrm{p}.5 \end{pmatrix} \\ \lambda \mathrm{p} : [\![\text{high}]\!]_{\text{posn}}(\mathrm{x}).1\ .\ \mathrm{refl}_{\begin{pmatrix} \text{dist} \\ \|\mathrm{p.v}\| \end{pmatrix}} \end{pmatrix}$$

As shown, the presuppositions of $[\![\text{high}]\!]_{\text{posn}}$ are sent to the presuppositions of $[\![\text{high}]\!]_{\text{elev}}$ by injecting the proof of $\textsc{posn}(\mathrm{x}, \mathrm{u}, \mathrm{v})$ as a proof of $\textsc{posn}(\mathrm{x}, \mathrm{u}, \mathrm{v}) + \textsc{axis}(\mathrm{x}, \mathrm{u}, \mathrm{v})$. As before, the fact that both $[\![\text{high}]\!]_{\text{posn}}$ and $[\![\text{high}]\!]_{\text{elev}}$ return the same degree is witnessed by the appropriate reflexivity proof. Henceforth, I shall generally omit discussion of how arrows are implemented where this can be easily deduced from the implementation of senses.

As discussed above, I take the non-spatial senses of *high* to be disconnected from the spatial senses. For example, consider the 'social status' sense $[\![\text{high}]\!]_{\text{social}}$, as in *high office*, *high position*, *high standing*, and so on. On the one hand, it is clear that this sense cannot be coordinated with any of the spatial senses without giving rise to zeugma, e.g. ?*John's balcony is as high as his social standing.* By our criteria, this suggests that $[\![\text{high}]\!]_{\text{social}}$ does not share a common join with any of the senses in Figure 3.6. Moreover, although $[\![\text{high}]\!]_{\text{social}}$ is mutually compatible with the spatial senses, the combination does not form a unified prototype: someone who is high in both a positional and social sense is *high* in two completely different ways, not in a single more typical way. This suggests that $[\![\text{high}]\!]_{\text{social}}$ does not share a common meet with any of the spatial senses. Similar observations could be made with respect to the other non-spatial senses.

However, despite being disconnected from the spatial senses, the non-spatial senses are still part of the *high* network. It is important to distinguish between two disconnected parts of the same lexical network and two distinct lexical networks which happen to share a common pronunciation. The latter are perceived as unrelated in meaning, such as the two interpretations of the word *bank*, 'financial institution' and 'side of a river'. In contrast, two disconnected pieces of the same lexical network are perceived as related 'metaphorically' or 'loosely', like the spatial and social senses of *high*.

I would tentatively suggest that the perception of a loose metaphorical connection between disconnected senses stems from the human ability to quickly construct meets and joins in lexical networks. For example, given $[\![\text{high}]\!]_{\text{social}}$ and $[\![\text{high}]\!]_{\text{elev}}$, one can easily imagine a common join in which ground level is identified with the bottom of the social hierarchy, and distance above ground is identified with social standing. At the same time, one can also imagine a common meet in which physical elevation and social status coincide, so that both senses are simultaneously satisfied. On this account, the difference between a metaphorical sense connection and an ordinary sense connection is that a metaphorical sense connection is computed 'on the fly', whereas an ordinary connection is lexically available. In order for a metaphorical connection to be constructed, the two senses must belong to the same network.

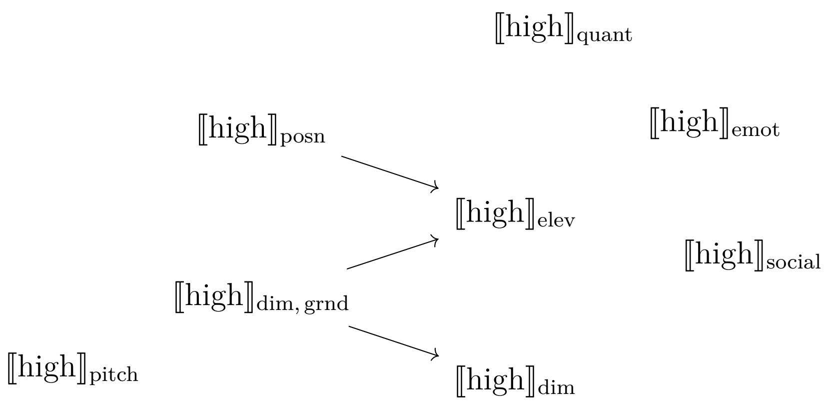


Figure 3.7: The non-spatial senses of *high* are disconnected from the spatial senses, but still part of the same network.

3.4.3 Representing antonymy: the low network

The negative adjective *low* is a perfect antonym to *high*. Like *high*, *low* has both a positional interpretation, as in *low airplane*, and a dimensional interpretation, as in *low tower*. It also shows a general 'elevation' interpretation which unites the two, as in *the plane is lower than the tower*. What is more, *low* is parallel to *high* in most if not all of its non-spatial applications, including emotion (*to feel low*), social status (*low rank*), pitch (*low note*), quantity (*low number*), and so on. In other words, the structure of the *low* network is the same as the *high* network. The same goes for other pairs of perfect antonyms, such as *thick/thin*, *big/small*, *light/heavy*, and so on, though not for imperfect antonyms like *long/short*.

How should the relationship between antonymous adjectives be represented? Recall that arrows within a lexical network are idiosyncratic – they capture unique relationships between senses of a particular lexical item – whereas the relationship between a positive polarity adjective and its negative polarity counterpart is always the same: *high* is to *low*, as *big* is to *small*, as *light* is to *heavy*, as *remarkable* is to *unremarkable*, and so on. This suggests that antonymy should be described by a function which applies to an entire lexical network. I shall call the function which implements antonymous opposition UN, after the *un-* morpheme. UN converts a network of positive adjective meanings into a network of negative adjective meanings. Its action on senses is shown below:

(242) $\textsc{un} : \mathit{Gradable}(+) \rightarrow \mathit{Gradable}(-)$

$$\textsc{un} \coloneqq \lambda \mathrm{G} : \mathit{Ind} \rightarrow \mathit{Gradable}(+)\ .$$
$$\lambda \mathrm{x} : \mathit{Ind}\ . \begin{pmatrix} \mathrm{P} \coloneqq \mathrm{G(x).1} \\ \lambda \mathrm{p} : \mathrm{P}\ . \begin{pmatrix} \mathrm{G(x).2(p).1} \\ \mathrm{negate(G(x).2(p))} \end{pmatrix} \end{pmatrix}$$

That is, *un-* acts on a sense by negating the degree returned by its measure function. To give an example, applying UN to the sense $[\![\text{high}]\!]_{\text{dim, grnd}}$ gives us the antonymous sense $[\![\text{low}]\!]_{\text{dim, grnd}}$, as follows:

(243) $[\![\text{low}]\!]_{\text{dim, grnd}} \coloneqq \textsc{un}([\![\text{high}]\!]_{\text{dim, grnd}})$

$$= \lambda \mathrm{x} : \mathit{Ind} \ . \begin{pmatrix} \mathrm{P} \coloneqq [\![\text{high}]\!]_{\text{dim, grnd}}(\mathrm{x}).1 \\ \lambda \mathrm{p} : \mathrm{P} \ . \begin{pmatrix} [\![\text{high}]\!]_{\text{dim, grnd}}(\mathrm{x}).2(\mathrm{p}).1 \\ \text{negate}([\![\text{high}]\!]_{\text{dim, grnd}}(\mathrm{x}).2(\mathrm{p})) \end{pmatrix} \end{pmatrix}$$

$$= \lambda \mathrm{x} : \mathit{Ind} \ . \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{u} : \mathit{Vector} \\ \mathrm{v} : \mathit{Vector} \\ \textsc{grnd}(\mathrm{u}) \\ \textsc{axis}(\mathrm{x}, \mathrm{u}, \mathrm{v}) \\ \textsc{up}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \begin{pmatrix} \text{dist} \\ -\|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

As shown, $[\![\text{low}]\!]_{\text{dim, grnd}}$ is exactly the same as $[\![\text{high}]\!]_{\text{dim, grnd}}$, except that instead of returning a degree with positive polarity, it instead returns a degree with negative polarity. Since positive degrees on the distance scale are positive real numbers, it makes sense to represent negative degrees on the distance scale are negative real numbers. (Note that this does not necessarily work for every scale, since not all scales are isomorphic to the positive real numbers. Some scales are not really 'numerical' at all, in the sense that they do not support operations like addition, subtraction, multiplication, and so on.)

UN acts not only on senses but also on sense connections. Suppose we have two positive gradable adjectives $\mathrm{G}, \mathrm{H} : \mathit{Gradable}(+)$ linked by an arrow:

(244) $\alpha : \mathrm{G} \to_{\mathit{Gradable}(+)} \mathrm{H}$

This can be lifted to an arrow $\textsc{un}(\alpha)$ between their antonyms as follows:

(245) $\textsc{un}(\alpha) : \textsc{un}(\mathrm{G}) \to_{\mathit{Gradable}(-)} \textsc{un}(\mathrm{H})$

$$\textsc{un}(\alpha) \coloneqq \lambda \mathrm{x} : \mathit{Ind} \ . \begin{pmatrix} \lambda \mathrm{p} : \textsc{un}(\mathrm{G})(\mathrm{x}).1 \ . \ \alpha(\mathrm{x}).1(\mathrm{p}) \\ \lambda \mathrm{p} : \textsc{un}(\mathrm{G})(\mathrm{x}).1 \ . \begin{pmatrix} \alpha(\mathrm{x}).2(\mathrm{p}).1 \\ \mathrm{ap}_{\text{negate}}(\alpha(\mathrm{x}).2(\mathrm{p}).2) \end{pmatrix} \end{pmatrix}$$

Note that the only aspect of the original map which needs to be altered is the second component of the commuting condition, which must be lifted from an equality between positive degrees to an equality between negative degrees using the equality-lifting function ap. Knowing how UN acts on senses and arrows amounts to knowing how it acts on an entire network. Figure 3.8 shows how the *low* network is derived from the *high* network through the application of UN.

The function UN is intended both as the interpretation of the morpheme $[\![\text{un-}]\!]$ and as a kind of lexical redundancy rule (Jackendoff 1975, Bresnan et al. 2015): it can be applied either within the morphosyntax, as in *unkind*, or within the lexicon, as in *low*. In this respect, it resembles other derivational affixes such as *-ness*, *-(i)fy*, *-able*, *-ish*, *-er*, etc., which in addition to being affixes function as derivational processes inside the lexicon relating different lexical items. As with other derivational morphology, lexical application of UN can block its morphosyntactic application. For example, **unbig*, **unlong* and **unhot* strike us as unacceptable because of the presence of the negative polarity adjectives *small*, *short* and *cold* in the lexicon.

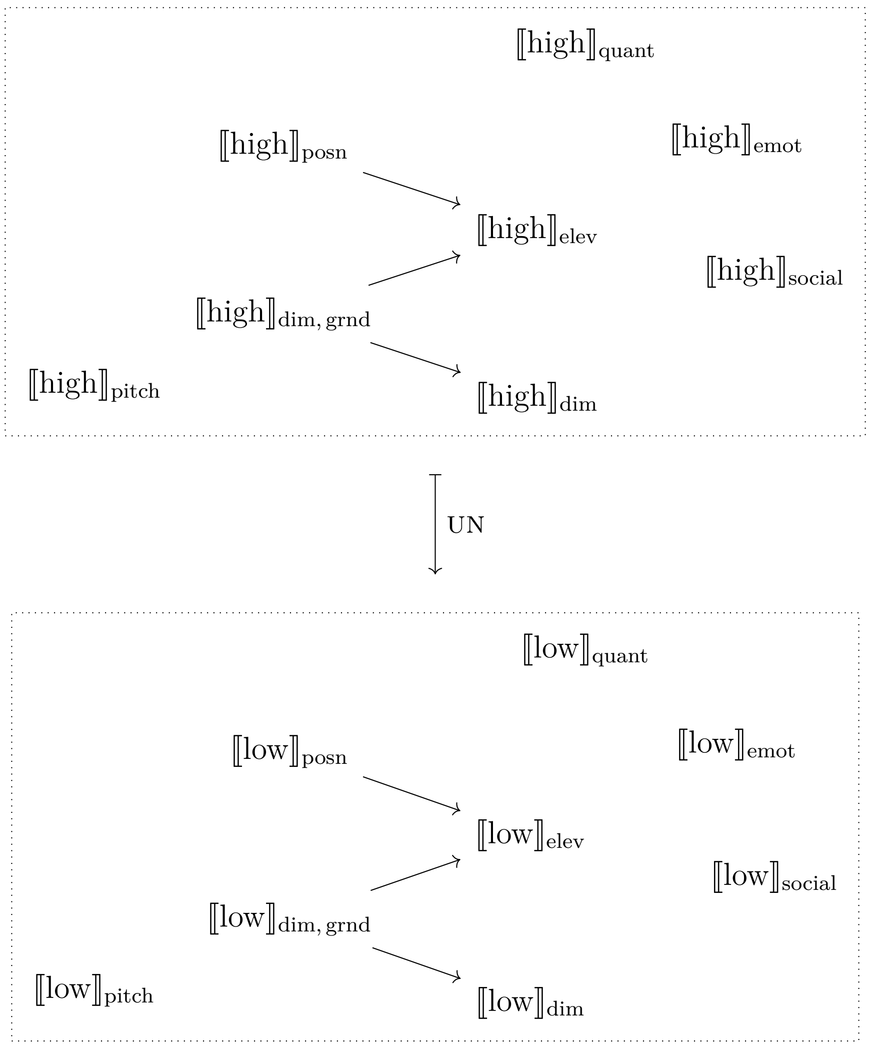


Figure 3.8: UN acts on the *high* network to derive the antonymous *low* network.

3.5 Tall

3.5.1 Introduction to tall

The most detailed description of the English adjective *tall* comes from Dirven & Taylor (1986), whose work was discussed in Section 1.3.4. Based on a questionnaire study in which participants were asked to rate the acceptability of various *tall* + noun combinations, they propose that *tall* is based on the following cognitive prototype:

(246) *Tall* prototypically applies to objects (Dirven & Taylor 1986):
   a. with a canonical vertical orientation (*tall person* is better than *tall infant*)
   b. whose vertical dimension is primary (*tall mountain* is better than *tall hill*)
   c. whose vertical dimension is sufficiently large (*tall fence* is better than *tall ribbon*)
   d. formed of solid material, not hollow (*tall tree* is better than *tall wardrobe*)
   e. profiled against a background (*tall bookcase* is better than *tall door*)
   f. which have acquired their height through a process of growth or construction (*tall building* is better than *tall window*)

A situation in which the object meets all of these criteria is judged as highly typical (e.g. *tall person*), whereas a situation in which only some criteria are met is judged as less typical (e.g. *tall lorry*). The authors also mention that *tall* can block the application of *high* in very prototypical situations – for instance, ?*high person* is unacceptable since people combine very prototypically with *tall*.

One issue with Dirven & Taylor's analysis is that it does not explain why certain combinations of criteria are sufficient, whereas others are not. For example, an object which satisfies both (a) and (b) (e.g. a wine bottle) can always be described as *tall* regardless of whether it satisfies the other criteria. Similarly, an object which satisfies both (a) and (c) (e.g. a fence) can always be described as *tall*. On the other hand, an object which satisfies only (d) and (e) (e.g. a planet) cannot be described as *tall*. The sufficiency of certain combinations of conditions but not others suggests a lexical network analysis in which arrows are used to express how the central prototype can be weakened. This supports the view that prototypes by themselves cannot play the role of concepts. Rather, one needs prototypes together with ways of acceptably weakening or transforming them.

Most languages do not distinguish *tall* and *high*. Rather, something like the prototype in (246) seems to be associated with the dimensional sense of height. For example, Goy (2002) conducted a study of Italian *alto* 'high/tall' inspired by Dirven and Taylor's methodology, whereby participants were presented with *alto* + noun combinations and asked to rate their acceptability. Nouns with high average scores (e.g. *torre* 'tower', *piramide* 'pyramid', *lampione* 'lamppost') had a vertical axis which was primary, larger than a human being, and pointed upwards. Nouns with medium scores (e.g. *automobile* 'car', *bichiere* 'glass', *uccello* 'bird') had only some of these properties, and nouns with low scores (e.g. *serpente* 'snake', *sigaretta* ‘cigarette’, *biro* ‘pen’) had none. Vogel (2004) found similar results for the Swedish adjective *hög* ‘high/tall’: she gives *hög träd* ‘tall tree’ as a prototypical combination, *hög tillbringare* ‘tall jug’ as a less typical combination, and ?*hög banderoll* ‘tall banner’ as an unacceptable combination.

An object described as *tall* must be conceptualized as directed upwards rather than downwards. For instance, we can describe a tree or a tower as *tall*, but not a stalactite or chandelier, since these are conceptualized as pointing downwards. Instead, objects which are directed downwards are usually described as *long*, provided that their downwards axis is sufficiently salient. The same holds for analogs of *tall* in other languages. For instance, Geckeler (1997) writes that French *haut* ‘tall/high’ must describe an upwards-directed object; it cannot describe a thread hanging from the ceiling, which is instead described as *long* ‘long’. Similarly, Linde-Usiekniewicz (2002) writes that Polish *wysoki* ‘high/tall’ cannot be used to describe objects which hang from above, like curtains or hanging lights. The upwards constraint may explain the tendency, often noted in the literature, for *tall* to prefer rigid over flexible objects. A flexible object like a piece of string does not ‘stand up’, but must be hung from above in order to have an identifiable vertical axis. This situation excludes the use of *tall*, an object hung from above is perceived as directed downwards.

Does *tall* have non-spatial senses? Many dictionaries list a sense meaning ‘considerable’ or ‘difficult’, as in *tall order* or *tall price*. However, this sense seems highly collocational, not occurring outside of a few fixed combinations. Novel combinations, such as ?*tall question*, ?*tall situation*, ?*tall issue*, and so on, seem bizarre. The same goes for other non-spatial senses, such as the use of *tall* to mean ‘difficult to believe’, which occurs in combinations like *tall tale* and *tall story* but for most speakers cannot be used productively in new combinations, e.g. ?*tall article*, ?*tall description*, ?*tall accusation*. For this reason, I have chosen not to include these senses in the *tall* network.

3.5.2 The tall network

The spatial senses of *tall* are all derived from a single prototype. Following the suggestions of Dirven & Taylor (1986), Goy (2002) and Vogel (2004), I take the prototypical sense of *tall* to refer to an axis which is:

(247) a. directed upwards in the environment
b. the primary axis of the object
c. larger than or comparable to human height

Examples of objects which satisfy all the conditions are trees, towers, lampposts, pillars and mountains.

Regarding the primary axis condition, I am assuming a distinction between the primary and secondary axes of an object, following work on human and computer vision (e.g. Marr & Nishihara 1978, Marr 1982, Biederman 1987). An object’s primary axis, if it has one, is distinguished from its other axes, either by being significantly larger or by symmetry. For example, the primary axis of a sofa is its side-side axis, since this is typically larger than the other axes. For an example of an axis which is distinguished by symmetry, consider a short bowl whose diameter is larger than its height. The primary axis of the bowl still refers to its height, despite the fact that it is not maximal, because this is the bowl's axis of rotational symmetry. The primary axis can be found by trying to find a cylinder which best approximates the shape of the object, where the cylinder is allowed to have an arbitrarily shaped cross-section (Marr 1982); the axis of the cylinder is then taken to be the primary axis of the object. Some objects have no primary axis, either because they have a highly irregular shape, like a crumpled newspaper, or because they have perfect rotational symmetry, like a football.

I have chosen not to incorporate the rigidity of the object into the *tall* prototype, as other authors have proposed. Flexible objects can in fact occur perfectly acceptably with *tall*, as in *tall grass*, *tall reed*, *tall stem*, and so on. In so far as there is a preference for *tall* to occur with rigid objects, this can be explained by the fact that flexible objects do not tend to point upwards, as explained in the previous subsection. Hence, it is not necessary to include a rigidity requirement in addition to the upwards direction requirement. Dirven & Taylor's (1986) condition that the object has acquired its height through a process of growth or construction is also unnecessary, as shown by combinations like *tall pole*, *tall hat*, *tall vase*, etc., which neither grow nor are built upwards incrementally. One could argue that such objects are conceptualized as exhibiting a kind of fictive motion in the upwards direction, but the same could be said for any upwards-extended object.

The conditions in (247) can be formalized as follows:

(248) $$[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st,\,large}} \coloneqq \lambda \mathrm{x} : Ind \;.\; \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : Vector \\ \textsc{up}(\mathrm{v}) \\ \textsc{1st}(\mathrm{x}, \mathrm{v}) \\ \textsc{large}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \;.\; \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

where components 2-4 of the presupposition type encode conditions (247a-c) respectively. Recall that the proofs of primitive predicates are intended to be pieces of perceptual information:

(249)
- A proof of $\textsc{up}(\mathrm{v})$ would involve evidence that the angle between v and the environmental direction vector **up** is greater than some threshold.
- A proof of $\textsc{1st}(\mathrm{x}, \mathrm{v})$ would consist of information showing that v is the primary axis of x.
- A proof of $\textsc{large}(\mathrm{v})$ would involve evidence that the magnitude of v is greater than some threshold corresponding to the average human height.

The prototype $[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st,\,large}}$ can be weakened in various directions. To begin with, one can drop the $\textsc{1st}(\mathrm{x}, \mathrm{v})$ or $\textsc{large}(\mathrm{v})$ requirements. For example:

(250)
- *Tall wine bottle*: the height of a wine bottle satisfies $\textsc{top}(\mathrm{x}, \mathrm{v})$, $\textsc{up}(\mathrm{v})$ and $\textsc{1st}(\mathrm{x}, \mathrm{v})$ but does not satisfy $\textsc{large}(\mathrm{v})$, since it is not comparable to a human height.
- *Tall wall*: the height of a wall satisfies $\textsc{top}(\mathrm{x}, \mathrm{v})$, $\textsc{up}(\mathrm{v})$ and $\textsc{large}(\mathrm{v})$ but fails to satisfy $\textsc{1st}(\mathrm{x}, \mathrm{v})$, since it is not primary.

Since they both fail to satisfy one of the tallness conditions, these senses should be perceived as slightly less typical than $[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st,\,large}}$. They are formalized as follows:

(251)
$$[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st}} := \lambda \mathrm{x} : Ind\ .\ \begin{pmatrix} \mathrm{P} := \begin{bmatrix} \mathrm{v} : Vector \\ \textsc{up}(\mathrm{v}) \\ \textsc{1st}(\mathrm{x}, \mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

$$[\![\mathrm{tall}]\!]_{\mathrm{up,\,large}} := \lambda \mathrm{x} : Ind\ .\ \begin{pmatrix} \mathrm{P} := \begin{bmatrix} \mathrm{v} : Vector \\ \textsc{up}(\mathrm{v}) \times \textsc{axis}(\mathrm{x}, \mathrm{v}) \\ \textsc{large}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

Notice that in $[\![\mathrm{tall}]\!]_{\mathrm{up,\,large}}$, because we have dropped the condition $\textsc{1st}(\mathrm{x}, \mathrm{v})$, we must add the condition $\textsc{axis}(\mathrm{x}, \mathrm{v})$ to ensure that v is an axis of x.

The senses $[\![\mathrm{tall}]\!]_{\mathrm{up,\,large}}$ and $[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st}}$ are subsumed by a common join. For instance, consider the following sentence:

(252) The wall is taller than the bollard.

The height of a wall satisfies $\textsc{large}(\mathrm{v})$ but not $\textsc{1st}(\mathrm{x}, \mathrm{v})$, since it is not primary, whereas the height of a bollard satisfies $\textsc{1st}(\mathrm{x}, \mathrm{v})$ but not $\textsc{large}(\mathrm{v})$, since it is not of comparable size to a human being. The fact that (252) is interpretable suggests that there is a sense which collapses the distinction between $\textsc{1st}(\mathrm{x}, \mathrm{v})$ and $\textsc{large}(\mathrm{v})$ into a single condition, which we can think of as requiring the axis to be 'significant'. This is formalized as follows:

(253)
$$[\![\mathrm{tall}]\!]_{\mathrm{up,\,sgfnt}} := \lambda \mathrm{x} : Ind\ .\ \begin{pmatrix} \mathrm{P} := \begin{bmatrix} \mathrm{v} : Vector \\ \textsc{up}(\mathrm{v}) \times \textsc{axis}(\mathrm{x}, \mathrm{v}) \\ |\textsc{1st}(\mathrm{x}, \mathrm{v}) + \textsc{large}(\mathrm{v})| \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

Note the use of propositional truncation. This is needed to ensure that there is a single composite arrow from $[\![\mathrm{tall}]\!]_{\mathrm{up,1st,large}}$ to $[\![\mathrm{tall}]\!]_{\mathrm{up,sgfnt}}$, which disregards whether one uses the proof of $\textsc{1st}(\mathrm{x}, \mathrm{v})$ or $\textsc{large}(\mathrm{v})$ to show significance. The requirement that there is only a single composite arrow is part of the conditions for a partial order.

In addition to describing an axis which is *actually* vertical, *tall* can also describe an axis which is *canonically* vertical. Consider the sentence:

(254) The wine bottle [lying on its side in storage] is taller than the pencil [held in an upright position].

This suggests that there is a more abstract sense of *tall* in which the axis is allowed to be either pointing upwards either in the environment or in the object's canonical orientation. We might refer to this as a 'general' vertical axis. Each of the senses defined up to this point – $[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st,\,large}}$, $[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st}}$, $[\![\mathrm{tall}]\!]_{\mathrm{up,\,large}}$ and $[\![\mathrm{tall}]\!]_{\mathrm{up,\,sgnft}}$ – can be weakened to a general vertical axis:

(255)
$$[\![\mathrm{tall}]\!]_{\mathrm{vert,\,1st,\,large}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\left(\begin{array}{l} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ \textsc{up}(\mathrm{v}) + \textsc{top}(\mathrm{x},\mathrm{v}) \\ \textsc{1st}(\mathrm{x},\mathrm{v}) \\ \textsc{large}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{array}\right)$$

$$[\![\mathrm{tall}]\!]_{\mathrm{vert,\,1st}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\left(\begin{array}{l} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ \textsc{up}(\mathrm{v}) + \textsc{top}(\mathrm{x},\mathrm{v}) \\ \textsc{1st}(\mathrm{x},\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{array}\right)$$

$$[\![\mathrm{tall}]\!]_{\mathrm{vert,\,large}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\left(\begin{array}{l} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ (\textsc{up}(\mathrm{v}) \times \textsc{axis}(\mathrm{x},\mathrm{v})) + \textsc{top}(\mathrm{x},\mathrm{v}) \\ \textsc{large}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{array}\right)$$

$$[\![\mathrm{tall}]\!]_{\mathrm{vert,\,sgfnt}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\left(\begin{array}{l} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ (\textsc{up}(\mathrm{v}) \times \textsc{axis}(\mathrm{x},\mathrm{v})) + \textsc{top}(\mathrm{x},\mathrm{v}) \\ |\textsc{1st}(\mathrm{x},\mathrm{v}) + \textsc{large}(\mathrm{v})| \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{array}\right)$$

where the predicate $\textsc{top}(\mathrm{x}, \mathrm{v})$ means 'v is the canonical vertical axis of x'. Note that the senses $[\![\mathrm{tall}]\!]_{\mathrm{vert,\,1st,\,large}}$, $[\![\mathrm{tall}]\!]_{\mathrm{vert,\,1st}}$, $[\![\mathrm{tall}]\!]_{\mathrm{vert,\,large}}$ and $[\![\mathrm{tall}]\!]_{\mathrm{vert,\,sgnft}}$ refer to an *actual or canonical vertical axis*, whereas the senses $[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st,\,large}}$, $[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st}}$, $[\![\mathrm{tall}]\!]_{\mathrm{up,\,large}}$ and $[\![\mathrm{tall}]\!]_{\mathrm{up,\,sgnft}}$ refer to an *actual vertical axis only.* There are no senses which refer to a canonical vertical axis only, to the exclusion of an actual vertical axis. I take this to be because the concept of an actual vertical axis is presupposed by, and prior to, the concept of a canonical vertical axis.

The various senses of *tall* form the lexical network shown in Figure 3.9. Notice how the senses form a lattice in which any two senses have a common meet and a common join. There is a unique initial sense because all the senses are derived from a common prototype; and there is a unique final sense because all the senses are understood to be abstractly similar, allowing them to be coordinated without zeugma. The more specific a sense – i.e. the fewer arrows are required to reach it, starting from the prototype – the more typical it is judged to be. Hence, $[\![\text{tall}]\!]_{\text{up, 1st, large}}$ is predicted to be more typical than $[\![\text{tall}]\!]_{\text{up, 1st}}$, which is predicted to be more typical than $[\![\text{tall}]\!]_{\text{vert, 1st}}$, and so on. If these predictions are found to be incorrect, the structure of the network would need to be changed accordingly. Note that unlike Dirven & Taylor's (1986) analysis, the network not only accounts for typicality judgements, but also explains which features of the prototype can be acceptably weakened and which cannot.

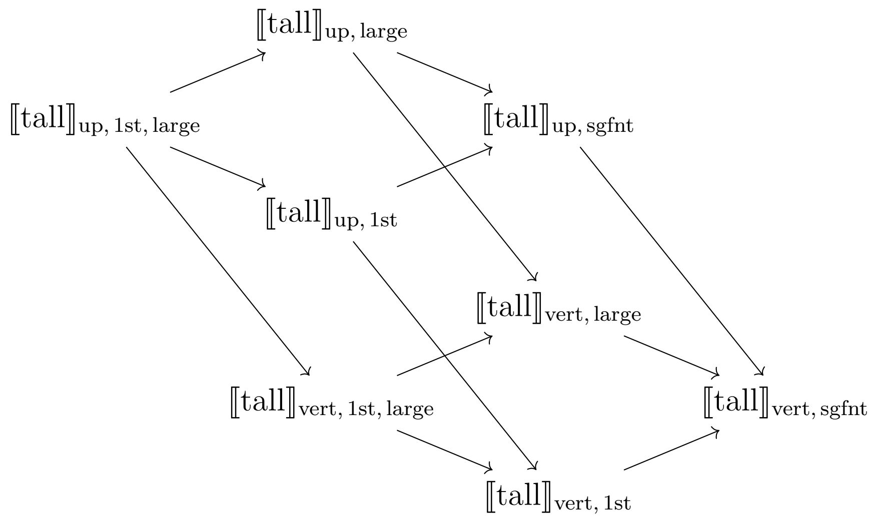


Figure 3.9: The *tall* network.

3.6 Length

3.6.1 Introduction to length

The most common description of length is that it refers to an object's maximal dimension. For example, in his analysis of German dimensional adjectives, Bierwisch (1967) assigns *lang* 'long' and *kurz* 'short' the feature +`max`, indicating a maximal dimension. Similarly, Lyons (1977) writes that if an object has a maximal extension, then this is identified as its length, unless canonical orientation causes it to be labelled as the width – as in the maximal dimension of a sofa – in which case the object will not have a length. However, other authors such as Lang (1989) and Fillmore (1997), have criticised the maximal theory. Lang points out that we can easily interpret sentences like the following, where length refers to the dimension which is secondary in extent:

(256) Examples from Lang (1989):

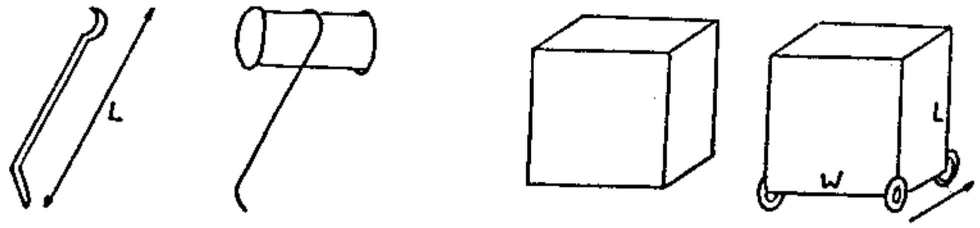
a. Die Drillemaschine ist breiter als lang.
'The drill is wider than it is long.'

b. Unser neues Doppelbett ist 2m lang und 3m breit.
'Our new double bed is 2m long and 3m wide.'

c. Der Samtrest ist 1.3m in der Breite, aber nur 0.5m lang.
'The velvet strip is 1.3m in width but only 0.5m long'

On the maximal theory, each of these examples ought to be contradictory. Lang's proposed solution is that length refers to whichever dimension is maximal in a prototypical version of the object. For instance, the front-back dimension of a drill is prototypically greater than its side-side dimension, so the front-back dimension can be described as a length even in cases where it is actually smaller. However, an issue with this analysis is that there may be types of objects whose length is prototypically shorter than their width. For example, the wingspan of a small aircraft is usually longer than the length of its fuselage, but the fuselage axis is nevertheless described as *length* and the wingspan as *width*.

Another aspect of length which is widely acknowledged in the literature is its preference for a horizontal dimension (Greimas 1966, Bierwisch 1967, Lyons 1977, Lafrenz 1983, Spang-Hanssen 1990). *Long* and *short* do not combine well with objects that are both canonically vertical and pointed upwards in the environment, as in ?*long tower*, ?*long person*, ?*long tree*, and so on – such expressions tend to give rise to the implicature that the object is in a horizontal or toppled orientation. However, *long* and *short* do seem to be acceptable with objects which are either canonically vertical but not actually vertical (e.g. the primary axis of a toppled tree), or actually vertical but not canonically vertical (e.g. a stick held in an upwards position). They are also acceptable with a inherently downwards-directed object, as in *long stalactite*, *long necktie*, *long curtains*, and so on. Because the acceptability conditions of length are in complementary distribution with those of *tall*, some authors (e.g. Vogel 2004), have argued that length has no orientation requirements of its own – its preference for a non-vertical axis is simply due to blocking by *tall*.



(a) The length of a poker is measured in a straight line, whereas the length of a thread is measured along a curve.

(b) According to Vandeloise, the length of a mobile entity is evaluated in its canonical direction of motion.

Figure 10, from Vandeloise (1988).

The most detailed analysis of length comes from Vandeloise's (1988) study of length and width, which was discussed in Section 1.3.4. Vandeloise points out that length is not always evaluated in a straight line. Rather, the length of a flexible or winding object, such as a thread or a river, is evaluated along the path of the object itself. Contrast this with a rigid pointed object, like a poker or a fork, whose length is evaluated in a straight line from base to tip (see Figure 9a). Vandeloise also argued for a connection between length and motion, arguing that the length of a mobile entity is evaluated along its canonical direction of motion. For instance, he argued that the cube shown in Figure 9b has no length until we conceptualize it as mobile, in which case its length is given by its frontal direction. This is also intended to explain why the frontal direction of a vehicle such as an aircraft can be referred to as a length even when it is less than the object's width.

A more recent analysis of length which is also based on the concept of path has been put forward by Zwarts (2003). Like Vandeloise, he notes that length may refer either to straight or curved objects. In the case of straight objects, the primary axis is simply given by a vector. In the case of a curved object, the primary axis is given by a *path*, which Zwarts analyses as a chain of vectors connected end-to-end. The number of vectors in its path corresponds to the scale or grain-size – a path containing only a few vectors is a very coarse-grained representation of the object, whereas a path containing many vectors is more fine-grained. (Paths of this kind are needed not only to describe length, but also for directional prepositions like *around* and *across* – for example, the sentence *the lake was two miles around* would involve a path that encircles the lake.) In Zwarts' analysis, the adjectives *long* and *short* each have two versions – one applying to vector axes and one applying to paths. The length of a path is given by summing together the length of all its component vectors.

In addition to its spatial senses, *long* also has a temporal sense, as in *long meeting*, *long day*, *four hours long*, and so on. Whereas the spatial sense has dimensions of distance, the temporal sense has dimensions of duration. The same parallelism between spatial length and temporal duration appears in many other languages – for instance, Japanese also uses *nagai* 'long' to describe duration, as in *nagai jinsei* 'long life' (Shimotori 2013). Evans (2013) argues that this parallelism is due to a conceptual metaphor DURATION IS LENGTH, a cross-cultural universal which arises because of common perceptual and motor experience. There are natural connections between DURATION IS LENGTH and other conceptual metaphors that have been proposed in cognitive linguistics, such as TIME IS MOTION (Lakoff & Johnson 1980), and TEMPORAL SEQUENCE IS LINEAR POSITION (Moore 2006).

3.6.2 The long network

My analysis of length is comparatively simple. To begin with, I do not agree with Vandeloise (1988) that length and width are mutually dependent concepts. In fact, there are many cases in which the duality between length and width breaks down, either because they refer to the same dimension (e.g. the primary axis of a sofa), or because the object has a width without having a length (e.g. a circle). At the same time, it is clear there is some kind of interaction between the two concepts. Indeed, when an object has two horizontal dimensions, it is often the case that one is labelled as the length and the other as the width. Rather than encoding this interaction explicitly – i.e. by directly referring to *long* in the description of *wide*, or vice versa – I prefer to explain interactions between spatial adjectives as a consequence of the primitives which they contain. *Long* contains the primitive $1\text{ST}(x, v)$, and therefore interacts in a complementary way with *wide*, which, as we shall see in Section 3.7, contains the primitive $2\text{ND}(x, v)$. *Long* also interacts in a competitive way with *tall*, since they both contain $1\text{ST}(x, v)$.

I also disagree with Vandeloise's claim that length has to do with the object's inherent direction of motion and the perspective from which it is viewed. For example, the frontal axis of a siege tower is its canonical direction of motion, but this cannot be described as *long*. The frontal axis of a desk is the direction in which it is canonically viewed, but this cannot be described as *long* either. Rather, the only condition which an axis is required to satisfy in order to count as a length is that it is primary – that is, it should satisfy $1\text{ST}(x, v)$. For instance, the primary axis of a stick can be described as *long*, despite the fact that it typically satisfies neither L_2 nor L_3, because it satisfies $1\text{ST}(x, v)$. The reason why the wheeled box in Figure 10 is perceived to have a length whereas the wheel-less box has no length is not because the wheeled box has a frontal axis and the wheel-less box does not; but rather because the wheeled box has a unique choice of primary axis due to its unique plane of symmetry, whereas the wheel-less box has multiple planes of symmetry and so cannot be assigned a primary axis.

I agree with other authors that *long* prefers a horizontal axis. However, this is not a necessary condition, since *long* can also apply to a diagonal or downwards-directed axis, as in *long curtain*, *long chandelier*, *long stalactite*, *long nose*, and so on. The unacceptability of *long* with an upwards-directed axis does not need to be encoded directly, since it can be explained as the blocking of *long* by *tall*. For example, consider the height of a tower block, which satisfies both $[\![\text{tall}]\!]_{\text{up, 1st, large}}$ and the 'primary axis' sense of length, $[\![\text{long}]\!]_{\text{1st}}$. Because $[\![\text{tall}]\!]_{\text{up, 1st, large}}$ is much more specific than $[\![\text{long}]\!]_{\text{1st}}$, the application of *long* in this situation is perceived as anomalous. In contrast to this, consider the primary axis of a wine bottle stored in a horizontal position, which satisfies both $[\![\text{long}]\!]_{\text{1st, horz}}$ and $[\![\text{tall}]\!]_{\text{vert, sgfnt}}$. Because these senses are roughly equally specific, the primary axis can be described as either *long* or *tall*.

To summarize, the most prototypical case of straight-line length refers to an axis which is both primary and horizontal. This is encoded as follows:

$$(257)\quad [\![\text{tall}]\!]_{\text{1st, horz}} := \lambda \text{x} : \mathit{Ind}\ .\ \begin{pmatrix} \text{P} := \begin{bmatrix} \text{v} : \mathit{Vector} \\ \textsc{1st}(\text{x}, \text{v}) \\ \textsc{horz}(\text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P}\ .\ \begin{pmatrix} \text{dist} \\ \|\text{p.1}\| \end{pmatrix} \end{pmatrix}$$

This sense can be weakened by dropping the horizontal condition, as when *long* is used to describe a downwards-directed object:

$$(258)\quad [\![\text{tall}]\!]_{\text{1st}} := \lambda \text{x} : \mathit{Ind}\ .\ \begin{pmatrix} \text{P} := \begin{bmatrix} \text{v} : \mathit{Vector} \\ \textsc{1st}(\text{x}, \text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P}\ .\ \begin{pmatrix} \text{dist} \\ \|\text{p.1}\| \end{pmatrix} \end{pmatrix}$$

As shown, $[\![\text{tall}]\!]_{\text{1st}}$ requires only that v is the primary axis of the object being described. A primary axis need not be maximal but can be determined by other constraints. For example, the primary axis of a small aircraft is the axis which runs through the fuselage, although the wingspan axis may in fact be larger. This is because the fuselage axis lies on the aircraft's plane of symmetry, which is a stronger requirement than the maximality requirement.

The senses $[\![\text{long}]\!]_{\text{1st, horz}}$ and $[\![\text{long}]\!]_{\text{1st}}$ involve only a straight-line axis, and so do not cover the length of pathlike objects such as rivers, threads, ropes or pieces of string. To express path length we need to introduce paths into the ontology, which can be done in various ways. One approach would be to treat them as primitives, introducing basic facts about paths as axioms, in the same way we have described vectors (see e.g. Piñón 1993, Krifka 1998, Eschenbach et al. 2000). This approach has the advantage that it stays close to human intuition and avoids introducing complex mathematical machinery, which is generally seen as implausible from a cognitive point of view. The alternative is to construct paths from more basic objects, such as sequences of places (Verkuyl & Zwarts 1992), or functions from some ordered domain to places (Zwarts & Winter 2000, Zwarts 2005). The advantage of this approach is its ontological parsimony and formal explicitness. Other considerations being equal, we prefer to build new types from pre-existing types rather than introducing them as primitives.

Following Zwarts (2005), I shall represent paths as smooth functions from the unit interval $[0,1]$ to vectors. That is, a path will be considered an element of something like the following type:

$$(259)\quad \mathit{Path} := \begin{bmatrix} \text{p} : [0,1] \to \mathit{Vector} \\ \text{smooth}(\text{p}) \end{bmatrix}$$

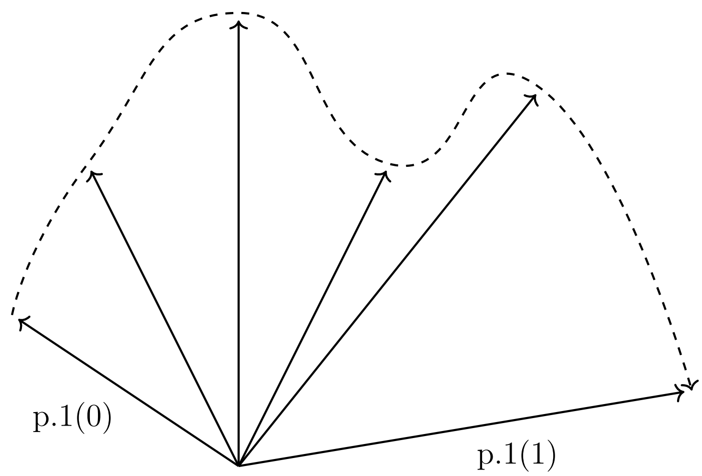


Figure 3.11: A path p is represented as a continuous sequence of vectors emanating from the same point. The length of a path can be estimated using a piecewise approximation such as that shown here.

where smooth(p) stands for some constraint encoding the idea that p is smooth. All the vectors in the path should be thought of as emanating from the same point as shown in Figure 3.11. As before, I take the origin of vectors to be arbitrary. Given a path p : *Path*, its start point is described by p.1(0) and its endpoint by p.1(1).

In addition to paths, we also need a function for computing the length of a path:

(260) length : $Path \rightarrow Real$

This is defined to be the usual notion of arc length. Given a path p : *Path*, consider any regular partition of $[0, 1]$ consisting of points $0 < t_1 < \cdots < t_{n-1} < 1$. This gives a piecewise approximation to p, whose length is $\sum_{i=0}^{n} \|\mathrm{p.1}(t_i) - \mathrm{p.1}(t_i - 1)\|$. The arc length is the upper bound of this value over any possible partition:

(261) $\mathrm{length(p)} \coloneqq \sup \sum_{i=1}^{n} \|\mathrm{p.1}(t_i) - \mathrm{p.1}(t_{i-1})\|$

In practice, arc length cannot be computed exactly unless one knows a parametric equation for the curve which has a closed-form solution. The representation of a curve as a parametric equation seems implausible from a cognitive point of view. Nevertheless, even people with little mathematical training understand that the length of a curve can be estimated by means of a piecewise approximation, with more fine-grained approximations yielding more precise estimates of length, eventually converging to the correct value. I therefore take it for granted that speakers have some notion of the 'true length' of a curve, despite being unable to precisely compute this quantity.

Having introduced paths and a way of understanding their length, we can now formalize the path sense of *long*:

(262) $$[\![\text{tall}]\!]_{\text{1st}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\ \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{p} : \mathit{Path} \\ \textsc{path}(\mathrm{x}, \mathrm{p}) \end{bmatrix} \\ \lambda \mathrm{r} : \mathrm{P}\ .\ \begin{pmatrix} \text{dist} \\ \text{length}(\mathrm{p}) \end{pmatrix} \end{pmatrix}$$

where the predicate $\textsc{path}(\mathrm{x}, \mathrm{p})$ means 'the shape of x is approximated by p'. Like the other primitives discussed in this chapter, a proof of $\textsc{path}(\mathrm{x}, \mathrm{p})$ is intended to be a piece of perceptual information. Given a path p satisfying $\textsc{path}(\mathrm{x}, \mathrm{p})$, the measure function returns the length of p, as defined in (261).

Path length and straight-line length can be sensibly brought together in a sentence like *the stick is long and so is the rope*, which in our system suggests that they ought to have a common join. One possibility would be to consider straight-line length as a special case of path length, in which the path consists only of a single vector. The arrow from straight-line length to path length would take a primary axis and consider it as a path with a single element. However, this strikes me as quite unintuitive, since many objects with a primary axis are not naturally approximated by paths. For instance, a typical desk has a primary axis in the side-side direction, but its shape cannot be sensibly approximated by a path. Moreover, there are objects whose straight-line length can be considered separate from their path length. For example, a corkscrew has a straight-line length from base to tip, but also a path length which winds around the helix.

Rather than representing straight-line length as a special case of path length, I take the two kinds of length to share a common generalization $[\![\text{long}]\!]_{\text{spatial}}$:

(263) $$[\![\text{long}]\!]_{\text{spatial}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\ .\ \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ \textsc{1st}(\mathrm{x}, \mathrm{v}) \end{bmatrix} + \begin{bmatrix} \mathrm{p} : \mathit{Path} \\ \textsc{path}(\mathrm{x}, \mathrm{p}) \end{bmatrix} \\ \lambda \mathrm{r} : \mathrm{P}\ .\ \begin{pmatrix} \text{dist} \\ \text{case r of} \\ \quad \text{inl}(\mathrm{r}') \text{ then } \|\mathrm{r}'.1\| \\ \quad \mid \text{inr}(\mathrm{r}') \text{ then length}(\mathrm{r}'.1) \end{pmatrix} \end{pmatrix}$$

As indicated, $[\![\text{long}]\!]_{\text{spatial}}$ requires either that x has a primary axis v, or that its shape can be approximated by a path p. In the former case, the measure function returns the magnitude of v; in the latter case, it returns the length of p. Both cases have dimensions of distance. The various senses of length form the lexical network shown in Figure 3.12. Notice how $[\![\text{long}]\!]_{\text{spatial}}$ subsumes both $[\![\text{long}]\!]_{\text{1st}}$ and $[\![\text{long}]\!]_{\text{path}}$.

As mentioned, *long* also has a temporal sense, as in *long meeting*, *long holiday*, *long wait*, and so on, which I refer to as $[\![\text{long}]\!]_{\text{durtn}}$. A common claim in cognitive linguistics is that spatial conceptualizations get transferred into the temporal domain by a conceptual metaphor TIME IS SPACE (e.g. Lakoff & Johnson 1980, Radden 2003, Evans 2013). Evans explicitly extends the TIME IS SPACE metaphor to include DURATION IS LENGTH. However, expressions like *long meeting* are generally not understood by speakers as being metaphorical (Jackendoff & Aaron 1991). Moreover, people frequently represent and reason about time and duration without recruiting

$$
\begin{tikzcd}
 & & \llbracket \text{long} \rrbracket_{\text{durtn}} \\
\llbracket \text{long} \rrbracket_{\text{1st, horz}} \arrow[r] & \llbracket \text{long} \rrbracket_{\text{1st}} \arrow[dr] & \\
 & & \llbracket \text{long} \rrbracket_{\text{spatial}} \\
 & \llbracket \text{long} \rrbracket_{\text{path}} \arrow[ur] &
\end{tikzcd}
$$

Figure 3.12: The *long* network.

spatial information, such as when deciding on the appropriate tense and aspect of a verb. These facts suggest that the concept of duration is not somehow derived from spatial length, but that the two have an equal status in the *long* lexical network. Moreover, the temporal sense does not have a common join with any of the spatial senses, since one cannot say something like:

(264) ? The street was long and so was the party.

which ought to be possible if spatial and temporal length were subsumed under a common generalization.

3.6.3 The short network

The English adjective *short* is not a perfect antonym of *long*. Rather, it appears to act as an antonym for both *long* and *tall*. For instance, a river is described as *short* in opposition to *long*, but a person is described as *short* in opposition to *tall*. It follows that the semantics of *short* is not given simply by applying the UN function to either the long network or the tall network, but by somehow combining their features. Like *long*, *short* can apply to curved and winding objects such as rivers, roads and ropes. It can also describe a time in the same way as *long*, as in *short meeting*, *short delay*, *short wait*, and so on. However, like *tall*, *short* can describe a vertical axis which is not the primary axis of the object, as in *short wall*, *short bike*, *short dog*, etc. Like *tall*, a canonical vertical axis can be described as *short* even when the object is not in its canonical orientation – for example, a bike lying on its side can be described as *short*. Because it combines the senses relating to both *long* and *tall*, the short network turns out to be rather complicated.

The structure of the short network is shown in Figure 3.13. As expected, it looks like a combination of the *tall* network in Figure 3.9 and the *long* network in Figure 3.12. Every sense in the network is antonymous to a sense from the *tall* network or a sense from the *long* network. There are four initial senses: $\llbracket\text{short}\rrbracket_{\text{up, 1st, large}}$, $\llbracket\text{short}\rrbracket_{\text{1st, horz}}$, $\llbracket\text{short}\rrbracket_{\text{path}}$, and $\llbracket\text{short}\rrbracket_{\text{durtn}}$ – which are the antonyms of $\llbracket\text{tall}\rrbracket_{\text{up, 1st, large}}$, $\llbracket\text{long}\rrbracket_{\text{1st, horz}}$, $\llbracket\text{long}\rrbracket_{\text{path}}$ and $\llbracket\text{long}\rrbracket_{\text{durtn}}$ respectively. There are three final senses: $\llbracket\text{short}\rrbracket_{\text{vert, sgfnt}}$, which subsumes all the antonyms of *tall*; $\llbracket\text{short}\rrbracket_{\text{1st/path}}$, which subsumes all the spatial antonyms of *long*; and $\llbracket\text{short}\rrbracket_{\text{durtn}}$, which is both final and initial. What connects the antonyms of *tall* and *long*

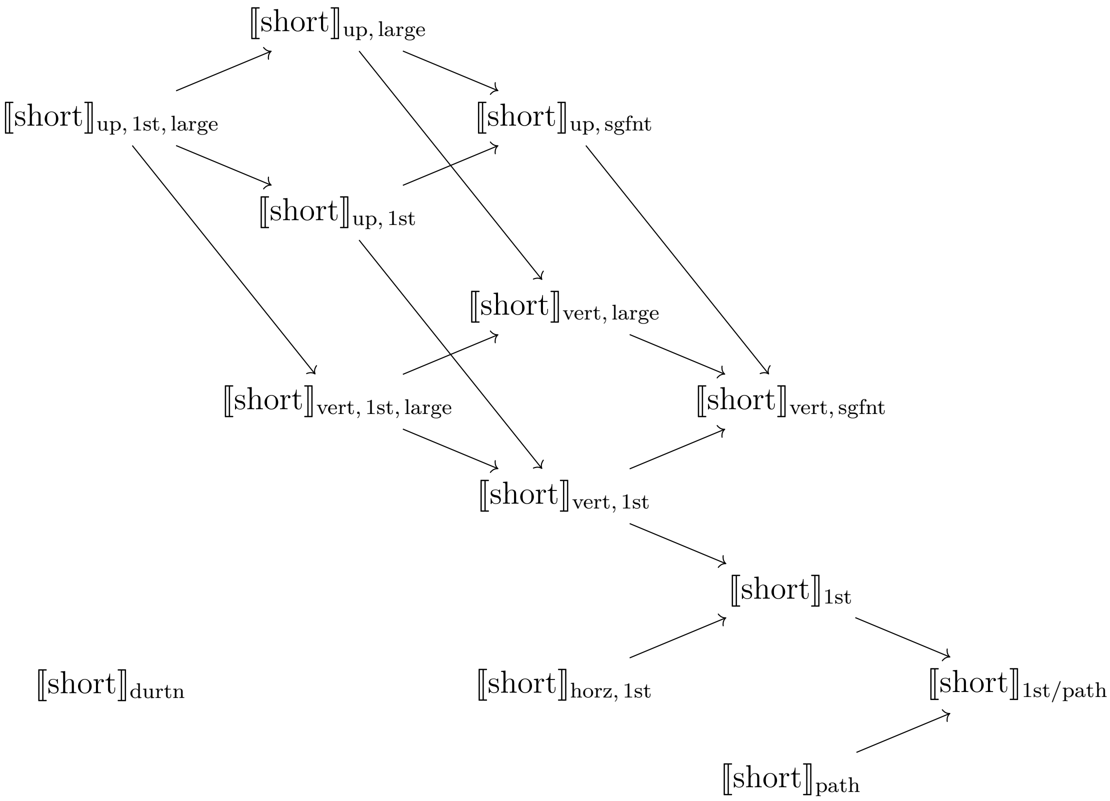


Figure 3.13: The *short* network.

together is the presence of the $\textsc{1st}(\mathrm{x},\mathrm{v})$ constraint in $[\![\text{short}]\!]_{\text{vert, 1st}}$ (the antonym of $[\![\text{tall}]\!]_{\text{vert, 1st}}$), which permits an arrow to $[\![\text{short}]\!]_{\text{1st}}$ (the antonym of $[\![\text{long}]\!]_{\text{1st}}$).

The structure of the network makes predictions about which senses can be acceptably compared. For example, consider the sentence *the wall is shorter than the car.* This has a tallness-based interpretation supported by $[\![\text{short}]\!]_{\text{up, large}}$, whereby the vertical dimension of the wall is compared with the vertical axis of the car, and a length-based interpretation supported by $[\![\text{short}]\!]_{\text{1st/path}}$, whereby the primary axis of the car is compared with the path length of the wall. Mixed interpretations are not possible, however – one does not consider comparing the height of the wall to the length of the car – because the network contains no sense that would support this interpretation. Contrast this to the sentence *the wall is shorter than the tower*, which has both a height-based interpretation supported by $[\![\text{short}]\!]_{\text{up, large}}$, whereby the height of the wall is compared to the height of the tower, and a (less typical) interpretation supported by $[\![\text{short}]\!]_{\text{1st/path}}$, whereby the length of the wall is compared to the height of the tower. This is possible because the height of a tower satisfies $\textsc{1st}(\mathrm{x},\mathrm{v})$, and can therefore be compared with a length.

I shall not go into the details of how the various senses and arrows in the *short* network are implemented, since this should be obvious given knowledge of how the senses of *tall* and *long* are implemented. It is worth noting, however, that since *short* is a negative polarity adjective, its senses are of type *Gradable*$(-)$ rather than *Gradable*$(+)$.

3.7 Width

3.7.1 Introduction to width

Wide is a complex category which is arguably the most polysemous of all the English spatial adjectives. A number of studies have attempted to describe various aspects of the concept. Perhaps the most general point of agreement is that the spatial meaning of *wide* is ambiguous between a sense involving the observer's line of sight, and a sense involving the inherent proportions of the object (Bierwisch 1989, Lyons 1977, Lang 1989, 2001, Vandeloise 1988, Spang-Hanssen 1990, Vogel 2004), henceforth referred to as 'lateral width' and 'secondary width' respectively. The need for this distinction is nicely illustrated by the sofa in Figure 3.14, which is taken from Vogel (2004). The sofa has two horizontal dimensions – x and z – both of which can be described as *wide/narrow*. The x dimension exemplifies lateral width, since it is orthogonal to the line of sight of a canonical observer/user, whereas the z dimension exemplifies secondary width, since it is orthogonal to the sofa's primary axis and horizontal.

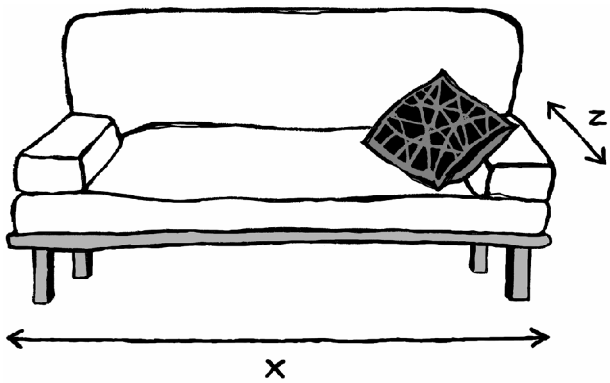


Figure 3.14: The lateral width of a sofa refers to x, whereas the secondary width refers to z, from Vogel (2004).

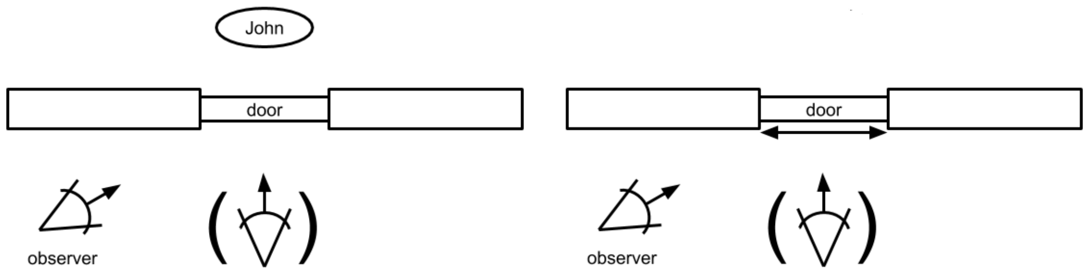
Another indication that lateral and secondary width should be distinguished is that coordinating them gives rise to zeugma. For example, a strip of wood is usually thought of as having a secondary width, whereas a pair of glasses is thought of as having a lateral width. It is therefore unclear what the sentence *the strip is wider than the glasses* is supposed to mean. To interpret the sentence, one must either force both widths to be secondary by comparing the shape of the strip and the shape of the lenses, or force both to be lateral by imagining the strip in some canonical orientation where it can be assigned a lateral axis that can be compared with the width of the glasses. Note, however, that whilst secondary and lateral width cannot be compared, they are not mutually exclusive: rather, a single axis may be both lateral and secondary, such as the side-side axis of a car.

Secondary width refers to an axis which is orthogonal to the object's primary axis, where 'primary axis' is understood in the sense of $\textsc{1st}(\mathrm{x}, \mathrm{v})$. It follows that an object without a primary axis, such as a spherical or disk-like object, cannot have a secondary width. Some objects, such as rivers, ribbons, roads, corridors, and so on, do not have a global secondary width, but rather many different secondary widths at different points along their length. To give an example, the width of the river Nile varies from around 7.5km to around 350m, depending on where it is measured. Such an object can be represented by a 'ribbon' – a sequence of vectors forming a path, each of which has an orthogonal width vector indicating the width of a segment. The total width of the ribbon is given by the average over the width of all of its segments. This explains why we can say *the Nile is wider than the Amazon*, despite the fact that there are choices of points for which this is not the case.

Secondary width prefers an axis which is horizontal (Bierwisch 1967, Lyons 1977, Lafrenz 1983, Spang-Hanssen 1990, Lang 2001, Vogel 2004). For example, imagine a plank of wood which is 300cm × 15cm × 3cm. When the secondary dimension is horizontal (as when the plank is oriented like a shelf) it is a very typical example of width, but if the plank is tilted so that the secondary dimension is vertical the application of width becomes less acceptable. A similar preference can be seen in other languages. Lafrenz (1983) writes that German *breit* 'broad/wide' and *schmal* 'narrow' typically describe the smaller of an object's two horizontal dimensions; for Spang-Hanssen (1990), French *largeur* 'width' is assigned to an object after its primary and vertical dimensions have received a label, with the result that it is always horizontal; Vogel (2004) writes that Swedish *bred* 'broad/wide' refers to "the smaller dimension in the horizontal plane".

Lateral width refers to an axis which is orthogonal to the line of sight of some observer, making it an inherently perspective-dependent concept. There is a natural connection between lateral width and the so-called projective prepositions such as *in front of*, *behind*, *beside*, *to the left of*, and *to the right of*, which also involve an observer. For example, imagine facing a television screen in a canonical viewing situation. The width of the screen is the axis which is orthogonal to your line of sight, the same axis which is involved in the interpretation of *beside*, *to the left of* and *to the right of*. In contrast, the prepositions *in front of* and *behind* involve the axis which is aligned with your line of sight. In this way, the axes of an object inherit their labels from how they are viewed in a canonical situation (Clark 1973, Fillmore 1997, Miller & Johnson-Laird 1976, Herskovits 1987, Levinson 1996, 2003). An object with multiple possible viewpoints will have multiple choices of axis labels. For example, a rectangular table in the centre of an empty room has four possible vantage directions corresponding to its four different sides, each of which is associated with a different interpretation of *wide*, and also a different interpretation of *in front of*, *behind*, *beside*, *to the left of* and *to the right of*.

The 'observer' associated with a preposition may not correspond to the location of an actual entity, but can be entirely virtual. Herskovits (1987) illustrates this using the sentence *John is behind the door*, said by an observer who is facing the door at an oblique angle. As shown in Figure 3.15a, this sentence evokes a virtual observer (shown in parentheses) who is directly in line with the door and facing it; John is understood to be directly behind the door from the perspective of this virtual observer. The same virtual perspective is evoked by the sentence *The door is wide*, where *wide* refers to the axis which is orthogonal to the virtual observer's line of sight, as shown in Figure 3.15b. For some objects, the virtual observer may



(a) *John is behind the door.* **(b)** *The door is wide.*

Figure 3.15: Lateral width can involve a virtual observer, just like projective prepositions. Figure based on Herskovits (1987).

be on or inside the object itself. For example, the axes of a vehicle are labelled from the perspective of someone inside it, and likewise for a bed, sofa or armchair.

In addition to secondary and lateral width, there is also a further sense of width which refers to a two-dimensional area, as in *wide field*, *wide beach*, *wide ocean*, *wide view*. The area need not be associated with an actual surface, but can also be a kind of imaginary area 'swept out' by a curve, as in *wide arc*, *wide circle*, and so on. Some languages – particularly Germanic languages – distinguish lexically between the distance senses of width (secondary and lateral) and the area sense. For example, German uses *breit* for distance width, but *weit* for area width. The claim that *weit* denotes an area is supported by the fact that it does not combine with distance-denoting measure phrases like *10m* (Lafrenz 1983, Durrell 1988). Swedish uses *bred* for distance width, but *vid* for area width, which likewise cannot occur with distance measure phrases (Vogel 2004).

Finally, *wide* also has a non-spatial sense meaning 'of great range or scope', as in *wide experience*, *wide knowledge*, *wide field of study*, and so on. This seems metaphorically related to a spatial sense in which an area is laid out in front of an observer, because an observer surveying a wide area has perceptual access to many different objects contained in that area. An abstract subject is usually described as having two dimensions: *wide* and *deep*, corresponding to the number of topics it contains and the extent or profundity of those topics. This parallels the description of area and depth of large horizontal object such as a lake. However, despite this conceptual connection, the scope sense and the area sense cannot be acceptably coordinated: ?*John's garden is wide and so is his knowledge of linguistics.* I therefore take the scope sense to be disconnected from the various spatial senses of *wide*.

3.7.2 Secondary and ribbon width

A secondary axis is encoded by the predicate $2\textsc{nd}(\mathrm{x}, \mathrm{v})$, meaning 'v is a secondary axis of x'. Given some $\mathrm{x} : Ind$ and $\mathrm{v} : Vector$, $2\textsc{nd}(\mathrm{x}, \mathrm{v})$ can be defined roughly as follows:

(265) 1. v is an axis of x

2. there is some $\mathrm{u} : Vector$, distinct from v, such that $1\textsc{st}(\mathrm{x}, \mathrm{u})$

3. for any other w : $Vector$, distinct from u and v, which satisfies $\textsc{axis}(x, w)$, v is not significantly smaller than w

Examples of axes which satisfy $2nd(x, v)$ include the secondary axis of a plank, blade, leaf, feather or strip. The reason for condition 3 is to block the application of width to a small thickness like the minimal dimension of a ruler, which satisfies all of the other conditions but cannot be described as a width since it is not judged to be significant in comparison to the other axes.

As mentioned previously, the fact that an axis is primary does not imply that it is maximal. In fact, an object's secondary axis can sometimes be larger than its primary axis due to symmetry considerations. For instance, consider the primary and secondary axes of a very wide maple leaf. The primary axis of the leaf is its midvein axis – this is the axis which you would place in a vertical position if told to hold the leaf "pointing up". The secondary axis is the side-side axis which runs orthogonal to the midvein, which may well be larger than the primary axis. The reason why the midvein axis is primary whereas the side-side axis is secondary is that the midvein axis lies on the leaf's unique plane of symmetry.

A prototypical secondary width is horizontal. For instance, the secondary dimension of a business card is more typically described as a width when the card is lying flat on a table, compared to when the card is placed upright on a stand. The horizontal version of secondary width is formalized as follows:

(266) $$[\![\mathrm{wide}]\!]_{\mathrm{2nd,\,horz}} \coloneqq \lambda \mathrm{x} : Ind \ . \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : Vector \\ \textsc{2nd}(\mathrm{x}, \mathrm{v}) \\ \textsc{horz}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

The horizontal requirement, whilst typical, is not necessary for an axis to count as a width. For example, imagine a large collection of planks all stacked in different directions, as one might find in a timber yard. The *widest plank* in the yard is the plank whose secondary dimension is largest, regardless of whether this is oriented vertically or horizontally. This is supported by the following sense, which requires only that the axis is secondary:

(267) $$[\![\mathrm{wide}]\!]_{\mathrm{2nd}} \coloneqq \lambda \mathrm{x} : Ind \ . \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : Vector \\ \textsc{2nd}(\mathrm{x}, \mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix}$$

The senses $[\![\mathrm{wide}]\!]_{\mathrm{2nd,\,horz}}$ and $[\![\mathrm{wide}]\!]_{\mathrm{2nd}}$ require that the object being described is completely straight, so that its primary axis is represented by a single vector. Path-like objects such as ribbons, strips, roads and corridors, do not have a global secondary width, but a different secondary width at every point along their length. Objects like this can be approximated by a configuration called a *ribbon*, which I shall describe by the following type:

$$\text{(268)} \quad \textit{Ribbon} := \begin{bmatrix} \mathrm{s} : \textit{Path} \\ \mathrm{v} : \textit{Path} \\ \prod_{\mathrm{i}:[0,1]} \begin{bmatrix} \|\mathrm{v.1(i)}\| = 1 \\ \langle \mathrm{v.1(i)}, \mathrm{dir(s,i)} \rangle = 0 \end{bmatrix} \end{bmatrix}$$

As shown, a ribbon consists of two paths, s and v. The path s gives the position of points along the object; whereas v describes the direction and magnitude of the width at every point. The constraint is that, at every point $\mathrm{i} : [0, 1]$ along the path, the width vector at i must be of unit length and orthogonal to the tangent vector of s at i (Recall that a path is a smooth function from $[0, 1]$ to vectors, so the notion of tangent vector at a point is well defined.) To express this constraint, I make use of a function $\mathrm{dir} : \textit{Path} \times [0, 1] \to \textit{Vector}$ which takes a path and a point and returns the unit tangent vector (the 'direction' of the path) at that point. An element of *Ribbon* can be visualized as a strip, as shown in Figure 3.16.

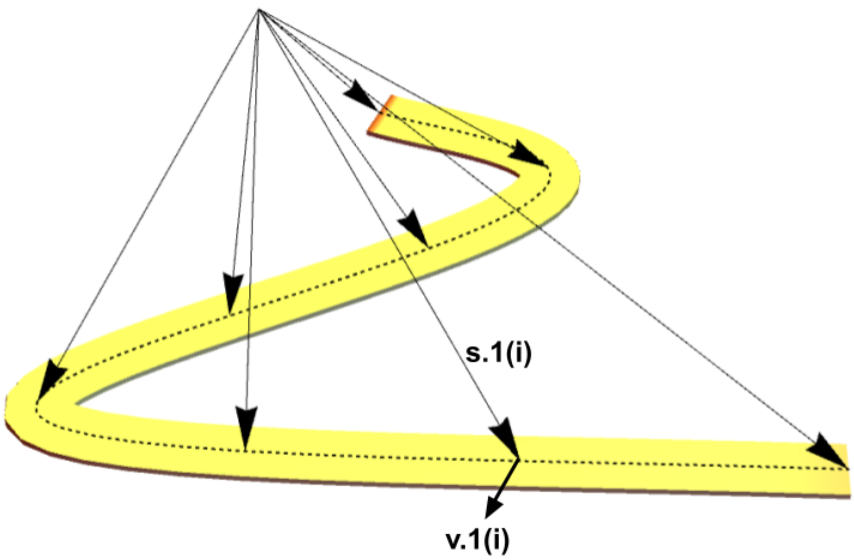


Figure 3.16: Visualizing an element of *Ribbon*. Each point $\mathrm{i} : [0, 1]$ along the strip has a position s.1(i) and normal direction v.1(i).

The connection between a ribbon and an individual is represented by the predicate RIBBON(x, r, w), meaning 'the shape of x is approximated by the ribbon r, with average width w'. The width is a positive real number representing the extension of the ribbon in the orthogonal direction given by the v vectors. Of course, for a real ribbon-like object such as a river or a road, the width can grow or shrink as we move along the path of the object. The width value w is intended as an approximation – it is the average or representative width associated with the object. Hence, the proposition RIBBON(x, r, w) is not intended as 'the width at any point along x is exactly w', but rather 'the width at any point along x is w, plus or minus some noise'. It is therefore a kind of statistical approximation to the underlying shape of x. This kind of approximation is needed if we are to represent the meaning of a sentence like *the Amazon is wider than the Nile.*

Just as the most prototypical case of secondary width is horizontal, the most prototypical case of ribbon width consists of a ribbon in which all the width vectors are horizontal, as in a river, road or staircase. This is encoded as follows:

(269) $$[\![\text{wide}]\!]_{\text{ribbon, horz}} \coloneqq \lambda \mathrm{x} : Ind \ . \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{r} : \textit{Ribbon} \\ \mathrm{w} : (0, \infty) \\ \textsc{ribbon}(\mathrm{x}, \mathrm{r}, \mathrm{w}) \\ \prod_{\mathrm{i}:[0,1]} \textsc{horz}(\mathrm{r}.2.1(\mathrm{i})) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \begin{pmatrix} \mathrm{dist} \\ \mathrm{p}.2 \end{pmatrix} \end{pmatrix}$$

As shown, $[\![\text{wide}]\!]_{\text{ribbon, horz}}$ presupposes: (i) a ribbon r; (ii) a positive real number w; (iii) a proof that x is represented by the ribbon r, with average width w; and (iv) a proof that, for every point along the ribbon r, the normal vector at that point is horizontal. Given a context satisfying these conditions, the measure function returns the value w, labelled as a distance. As in the case of secondary width, the horizontal requirement can be dropped. For example, a strip of cloth can be described as *wide* even when its width vectors are not horizontal. This gives us the most general kind of ribbon width:

(270) $$[\![\text{wide}]\!]_{\text{ribbon}} \coloneqq \lambda \mathrm{x} : Ind \ . \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{r} : \textit{Ribbon} \\ \mathrm{w} : (0, \infty) \\ \textsc{ribbon}(\mathrm{x}, \mathrm{r}, \mathrm{w}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \begin{pmatrix} \mathrm{dist} \\ \mathrm{p}.2 \end{pmatrix} \end{pmatrix}$$

which is identical to $[\![\text{wide}]\!]_{\text{ribbon, horz}}$ except without the horizontal requirement.

A ribbon width can be coordinated with an ordinary secondary width. For example, one can say something like:

(271) The road is wider than the car.

where the width of the road is evaluated in the ribbon sense and the width of the car in the secondary sense. To reflect this, we can introduce a common join of $[\![\text{wide}]\!]_{\text{2nd, horz}}$ and $[\![\text{wide}]\!]_{\text{ribbon, horz}}$ as follows:

(272) $$[\![\text{wide}]\!]_{\text{2nd/ribbon, horz}} \coloneqq \begin{pmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : \textit{Vector} \\ \textsc{2nd}(\mathrm{x}, \mathrm{v}) \\ \textsc{horz}(\mathrm{v}) \end{bmatrix} + \begin{bmatrix} \mathrm{r} : \textit{Ribbon} \\ \mathrm{w} : (0, \infty) \\ \textsc{ribbon}(\mathrm{x}, \mathrm{r}, \mathrm{w}) \\ \prod_{\mathrm{i}:[0,1]} \textsc{horz}(\mathrm{r}.2.1(\mathrm{i})) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \begin{pmatrix} \mathrm{dist} \\ \text{case p of} \\ \quad \mathrm{inl}(\mathrm{p}') \text{ then } \|\mathrm{p}'.1\| \\ \quad \mid \mathrm{inr}(\mathrm{p}') \text{ then } \mathrm{p}'.2 \end{pmatrix} \end{pmatrix}$$

As shown, $[\![\text{wide}]\!]_{\text{2nd/ribbon, horz}}$ presupposes either that x has a secondary horizontal axis v, or that x is approximated by a ribbon of average width w, each of whose normal vectors is horizontal. In the former case, the measure function returns the magnitude of v; in the latter case it returns w. Just as $[\![\text{wide}]\!]_{\text{2nd, horz}}$ and $[\![\text{wide}]\!]_{\text{ribbon, horz}}$ have a common join, the more general senses $[\![\text{wide}]\!]_{\text{2nd}}$ and $[\![\text{wide}]\!]_{\text{ribbon}}$ also have a common join, which is exemplified by the following kind of sentence:

(273) The shoelace is wider than the paperclip.

where the width of a shoelace is evaluated in the ribbon sense and the width of a paperclip in the secondary sense. This is represented by the common join $[\![\text{wide}]\!]_{\text{2nd/ribbon}}$:

(274) $$[\![\text{wide}]\!]_{\text{2nd/ribbon}} \coloneqq \lambda \mathrm{x} : \mathit{Ind} \,.\, \left(\begin{array}{l} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ 2\textsc{nd}(\mathrm{x}, \mathrm{v}) \end{bmatrix} + \begin{bmatrix} \mathrm{r} : \mathit{Ribbon} \\ \mathrm{w} : (0, \infty) \\ \textsc{ribbon}(\mathrm{x}, \mathrm{r}, \mathrm{w}) \end{bmatrix} \\ \\ \lambda \mathrm{p} : \mathrm{P} \,.\, \begin{pmatrix} \text{dist} \\ \text{case p of} \\ \quad \text{inl}(\mathrm{p}') \text{ then } \|\mathrm{p}'.1\| \\ \quad \mid \text{inr}(\mathrm{p}') \text{ then } \mathrm{p}'.2 \end{pmatrix} \end{array} \right)$$

The various senses of secondary and ribbon width form the network shown in Figure 3.17.

$$\begin{tikzcd}
{[\![\text{wide}]\!]_{\text{2nd, horz}}} \arrow[r] \arrow[dr] & {[\![\text{wide}]\!]_{\text{2nd}}} \arrow[dr] & \\
& {[\![\text{wide}]\!]_{\text{2nd/ribbon, horz}}} \arrow[r] & {[\![\text{wide}]\!]_{\text{2nd/ribbon}}} \\
{[\![\text{wide}]\!]_{\text{ribbon, horz}}} \arrow[r] \arrow[ur] & {[\![\text{wide}]\!]_{\text{ribbon}}} \arrow[ur] &
\end{tikzcd}$$

Figure 3.17: Secondary and ribbon width.

3.7.3 Lateral and passage width

Lateral width refers to the axis which is orthogonal to some canonical observer. For example, the lateral width of a sofa is its primary axis, whereas the lateral width of a door is its secondary axis. As discussed previously, the canonical observer does not always correspond to the location of an actual individual, rather it is adopted as an act of conceptualization. If the object is something which a human can 'inhabit', such as a car, a chair or an item of clothing, then its canonical observer is the perspective of a typical inhabitant. If the object is something with which humans interact in a typical way, like a building, a kitchen appliance or a tool, then its canonical observer is the perspective of a typical user. For objects which are themselves observers, such as people, animals or robots, the canonical observer is identical to the perspective of the object itself – such an object is its own canonical observer.

A lateral width situation consists of the following spatial elements:

(275)
- a vector p describing the position of the canonical observer
- three mutually orthogonal unit vectors $(\mathrm{t}, \mathrm{f}, \mathrm{r})$ describing the top, front and right directions of the canonical observer respectively
- a vector v describing an axis of the object

such that:

- $(\mathrm{p}, \mathrm{t}, \mathrm{f}, \mathrm{r})$ is a canonical observer for the object
- v is parallel to r

See Figure 3.18 for an illustration of a lateral width situation. As before, the origin of the position vector p is left unspecified. In the Figure, the width vector v has been drawn in the same direction as the observer's right vector r. However, it is also possible for v and r to be anti-aligned, in which case the width would go from right to left. All that is required is that there is some scalar a such that $\mathrm{v} = a \cdot \mathrm{r}$.

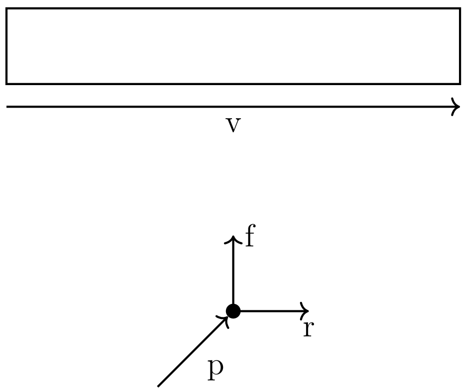


Figure 3.18: A lateral width situation, shown from a top-down perspective.

As discussed, there is a close connection between the lateral sense of *wide* and the projective prepositions *in front of*, *behind*, *beside*, and so on. In the literature on prepositions, it is common to distinguish between an 'intrinsic' sense, which involves the inherent top, front and right axes of an object, and a 'deictic' or 'relative' sense, which involves the axes of an observer. (Clark 1973, Fillmore 1997, Miller & Johnson-Laird 1976, Herskovits 1987, Levinson 1996, 2003). For example, the intrinsic sense of *the ball is in front of the car* would be 'the ball is in front of the car from the car's point of view', whereas the relative sense would be 'the ball is in front of the car from the observer's point of view'. This terminology is slightly misleading, however, because the so-called 'intrinsic' sense also involves the conceptualization of an observer, namely the *canonical* observer. For example,

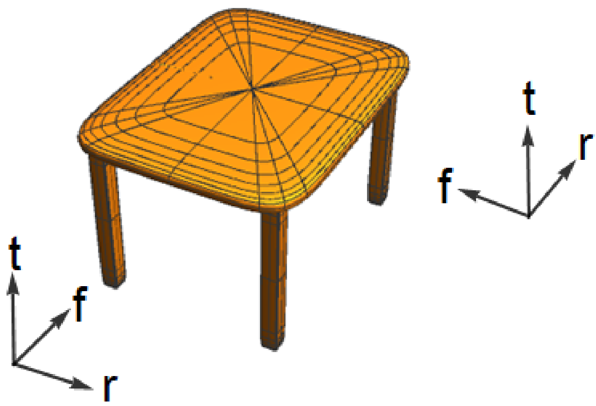


Figure 3.19: An object can have more than one canonical observer.

the intrinsic axes of a car are derived from the axes of a canonical observer, which is the perspective of an imaginary driver. Objects with no canonical observer do not support the use of intrinsic prepositions.

It is the intrinsic system rather than the relative system which is relevant to the semantics of *wide*. To see this, imagine looking at a car side-on so that its frontal axis is orthogonal to your line of sight. If width involved the actual observer, then it would be possible to describe the car's frontal axis as *wide/narrow*, but this is not the case. Rather, the width of the car always refers to its inherent left-right axis, regardless of the location of the speaker or addressee. The only situation in which the location of the actual observer can be relevant to the interpretation of *wide/narrow* is when the object has more than one possible canonical observer, as in Figure 3.19, in which case the canonical observer which is chosen for evaluating the width is usually the one which is closer to the location of the speaker or addressee. The speaker can also make a particular canonical perspective more salient through gesturing or verbal clarification.

The lateral sense of width can be written as follows:

$$(276) \quad [\![\mathrm{wide}]\!]_{\mathrm{latrl}} := \lambda \mathrm{x} : \mathit{Ind}\ .\ \begin{pmatrix} \mathrm{P} := \begin{bmatrix} \mathrm{p} : \mathit{Vector} \\ \mathrm{t} : \mathit{Vector} \\ \mathrm{f} : \mathit{Vector} \\ \mathrm{r} : \mathit{Vector} \\ \mathrm{v} : \mathit{Vector} \\ \textsc{obs}(\mathrm{x}, \mathrm{p}, \mathrm{t}, \mathrm{f}, \mathrm{r}) \\ \textsc{axis}(\mathrm{x}, \mathrm{v}) \\ \sum_{a:\mathit{Real}} \mathrm{v} = a \cdot \mathrm{r} \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.5\| \end{pmatrix} \end{pmatrix}$$

As shown, $[\![\mathrm{wide}]\!]_{\mathrm{latrl}}$ presupposes three vectors, p, t, f, r and v, such that $(\mathrm{p}, \mathrm{t}, \mathrm{f}, \mathrm{r})$

is a canonical observer for x; and v is an axis of x which is parallel to r (there exists some $\alpha : \mathit{Real}$ such that $\mathrm{v} = a \cdot \mathrm{r}$). Given a context satisfying these conditions, the measure function returns the magnitude of the axis vector v.

There is a natural connection between lateral width and the concept of ribbon width introduced in the previous subsection. Many ribbon-like objects – such as roads, rivers, tunnels and footpaths – function as passageways for people or vehicles. When moving along such an object, one's frontal axis at any particular point will typically be tangent to the path at that point, and one's lateral axis will typically be orthogonal to the path, so there will be a correspondence between the lateral width of the path and its ribbon width. The observation that lateral and ribbon width are linked through the concept of path was also made by Vogel:

> What are paths used for? Human beings tend to proceed along paths. We may easily imagine a situation in which someone is walking along a path. Thus, there is a functional situation, tied to the path. The human being is on the path, facing its maximal extension, while its minimal extension forms the left-right axis. [...] The dimension referred to as *bred* 'broad/wide' will at the same time be the smaller horizontal dimension [...] and it will be the left-right dimension for the user. (2004, p. 130)

I shall refer to this kind of width, where there is a canonical observer associated to each point along the path, as *passage* width.

For a ribbon-like object to have a passage width, there must be a canonical observer at every point along the path, corresponding to the idealized point of view of someone travelling down the path. At each point, the observer's front axis is aligned with the tangent vector to the path, and their right axis is aligned with the normal vector. The observer's top axis is given by the *cross product* of the front and right axes. Given two vectors $\mathbf{u}, \mathbf{v} : \mathit{Vector}$, the cross product $\mathbf{u} \times \mathbf{v}$ is the vector orthogonal to $\mathbf{u}$ and $\mathbf{v}$ whose magnitude is proportional to the area they span. I assume a left-handed convention so that the cross product of a vector pointing forwards and a vector pointing rightwards is a vector pointing upwards.

I propose to formalize the passage sense of width as follows:

$$(277)\quad [\![\mathrm{wide}]\!]_{\mathrm{passg}} \coloneqq \lambda \mathrm{x} : \mathit{Ind}\,.\left(\begin{array}{l} \mathrm{P} \coloneqq \left[\begin{array}{l} \mathrm{r} : \mathit{Ribbon} \\ \mathrm{w} : (0, \infty) \\ \textsc{ribbon}(\mathrm{x}, \mathrm{r}, \mathrm{w}) \\ \prod_{\mathrm{i}:[0,1]} \text{ let } \left\{\begin{array}{l} \mathrm{p} \coloneqq \mathrm{r}.1.1(\mathrm{i}) \\ \mathrm{f} \coloneqq \mathrm{dir}(\mathrm{r}.1, \mathrm{i}) \\ \mathrm{r} \coloneqq \mathrm{r}.2.1(\mathrm{i}) \\ \mathrm{t} \coloneqq \mathrm{f} \times \mathrm{r} \end{array}\right\} \text{ in } \textsc{obs}(\mathrm{p}, \mathrm{t}, \mathrm{f}, \mathrm{r}) \end{array}\right] \\ \lambda \mathrm{p} : \mathrm{P}\,.\begin{pmatrix} \mathrm{dist} \\ \mathrm{p}.2 \end{pmatrix} \end{array}\right)$$

As indicated, $[\![\text{wide}]\!]_{\text{passg}}$ presupposes that x is represented by a ribbon r with width w. At every point i : $[0, 1]$ on the ribbon, there is a canonical observer whose position is the location of that point, whose front axis is the direction of the path at that point, whose right axis is the orthogonal width vector, and whose top axis is the cross product of the front and top axes. For an illustration of passage width, see Figure 3.20. Given such a situation, the measure function returns the value w, which is the average length of the ribbon. It should be clear from (277) that passage width is a special case of ribbon width: there is an arrow from $[\![\text{wide}]\!]_{\text{passg}}$ to $[\![\text{wide}]\!]_{\text{ribbon}}$ which forgets about the canonical observer at each point along the path.

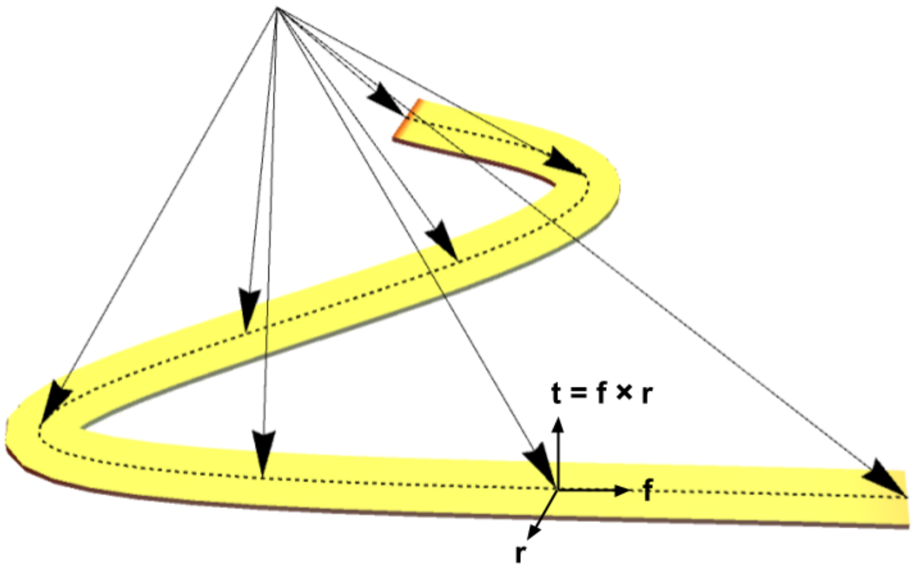


Figure 3.20: Passage width: at every point along the ribbon there is a canonical observer whose front and right axes are aligned with the tangent and normal vectors at that point and whose top axis is given by the cross product.

There is a natural connection between the passage sense of width and the lateral sense of width, since both involve a canonical observer. One can therefore sensibly say something like:

(278) The piano is as wide as the corridor.

comparing the lateral axis of the piano to the passage width of the corridor. This suggests that $[\![\text{wide}]\!]_{\text{latrl}}$ and $[\![\text{wide}]\!]_{\text{passg}}$ share a common join, which I shall refer to as $[\![\text{wide}]\!]_{\text{obs}}$:

(279) $[\![\text{wide}]\!]_{\text{obs}} :=$

$\lambda \text{x} : \mathit{Ind}\ .$

$$\left(\begin{array}{l} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{p} : \mathit{Vector} \\ \mathrm{t} : \mathit{Vector} \\ \mathrm{f} : \mathit{Vector} \\ \mathrm{r} : \mathit{Vector} \\ \mathrm{v} : \mathit{Vector} \\ \textsc{obs}(\mathrm{x}, \mathrm{p}, \mathrm{t}, \mathrm{f}, \mathrm{r}) \\ \textsc{axis}(\mathrm{x}, \mathrm{v}) \\ \sum_{a:\mathit{Real}} \mathrm{v} = a \cdot \mathrm{r} \end{bmatrix} + \begin{bmatrix} \mathrm{r} : \mathit{Ribbon} \\ \mathrm{w} : (0, \infty) \\ \textsc{ribbon}(\mathrm{x}, \mathrm{r}, \mathrm{w}) \\ \prod_{\mathrm{i}:[0,1]} \text{ let } \left\{\begin{array}{l} \mathrm{p} \coloneqq \mathrm{r.1.1(i)} \\ \mathrm{f} \coloneqq \mathrm{dir(r.1, i)} \\ \mathrm{r} \coloneqq \mathrm{r.2.1(i)} \\ \mathrm{t} \coloneqq \mathrm{f} \times \mathrm{r} \end{array}\right\} \text{ in } \\ \qquad\qquad \textsc{obs}(\mathrm{p}, \mathrm{t}, \mathrm{f}, \mathrm{r}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \;.\; \left(\begin{array}{l} \mathrm{dist} \\ \text{case p of} \\ \quad \mathrm{inl(p')} \text{ then } \|\mathrm{p'.5}\| \\ \quad \mid \mathrm{inr(p')} \text{ then } \mathrm{p'.2} \end{array}\right) \end{array}\right)$$

As indicated, $[\![\text{wide}]\!]_{\text{obs}}$ requires either that x has a global lateral axis, as in $[\![\text{wide}]\!]_{\text{latrl}}$, or that x is represented by a ribbon along which a canonical observer might travel, as in $[\![\text{wide}]\!]_{\text{ribbon}}$. In the former case, the measure function returns the magnitude of the lateral axis; in the latter case it returns the average width of the ribbon.

Figure 3.21 collects together all of the different kinds of width which have been defined up to this point, which we can refer to collectively as the distance senses, since they all return a degree with dimensions of distance. There are three prototypes: $[\![\text{wide}]\!]_{\text{2nd, horz}}$, which is the most prototypical case of secondary width; $[\![\text{wide}]\!]_{\text{passg, horz}}$, which is the width of a horizontal passage such as a corridor or road; and $[\![\text{wide}]\!]_{\text{latrl}}$, which is the lateral width of an object with a canonical observer. There are two final senses: $[\![\text{wide}]\!]_{\text{2nd/width}}$, which connects secondary and ribbon width; and $[\![\text{wide}]\!]_{\text{obs}}$, which connects passage and lateral width. The network predicts that a general lateral width cannot be coordinated with a general secondary width. For example, the sentence:

(280) ? The monitor is 30 inches wide, and so is the plank.

cannot be interpreted as comparing the lateral width of the monitor and the secondary dimension of the plank. However, given a long pathlike object which functions as a passage, its width can be compared with an ordinary secondary width (due to convergence at $[\![\text{wide}]\!]_{\text{2nd/ribbon}}$) or with an ordinary lateral width (due to convergence at $[\![\text{wide}]\!]_{\text{obs}}$). For example, the following are both acceptable:

(281) a. The corridor is 30 inches wide, and so is the plank.
b. The corridor is 30 inches wide, and so is the monitor.

A passage can be thought of as having a width both in virtue of its dimensions and in virtue of having a canonical observer.

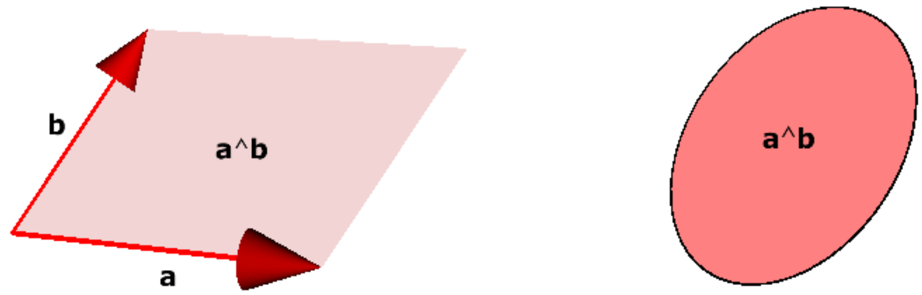
$$\begin{tikzcd}
 & \llbracket \text{wide} \rrbracket_{\text{2nd, horz}} \arrow[rr] \arrow[dr] & & \llbracket \text{wide} \rrbracket_{\text{2nd}} \arrow[dr] & \\
 & & \llbracket \text{wide} \rrbracket_{\text{2nd/ribbon, horz}} \arrow[rr] & & \llbracket \text{wide} \rrbracket_{\text{2nd/ribbon}} \\
 & \llbracket \text{wide} \rrbracket_{\text{ribbon, horz}} \arrow[rr] \arrow[ur] & & \llbracket \text{wide} \rrbracket_{\text{ribbon}} \arrow[ur] & \\
\llbracket \text{wide} \rrbracket_{\text{passg, horz}} \arrow[rr] \arrow[ur] & & \llbracket \text{wide} \rrbracket_{\text{passg}} \arrow[rr] \arrow[ur] & & \llbracket \text{wide} \rrbracket_{\text{obs}} \\
 & & & \llbracket \text{wide} \rrbracket_{\text{latrl}} \arrow[ur] &
\end{tikzcd}$$

Figure 3.21: The distance senses of width.

3.7.4 Area and arc width

As discussed, there is also a sense of *wide* which refers to the extent of an area. Examples include *wide field*, *wide plain*, *wide beach*, *wide ocean*, and so on. As things stand, the vector space ontology contains no notion of area, so it is difficult to see how to represent this idea. In physics and engineering, one often treats oriented areas in terms of the cross product, using the fact that the magnitude of the cross product $\mathbf{u} \times \mathbf{v}$ is the area of the parallelogram formed by the vectors $\mathbf{u}$ and $\mathbf{v}$. However, the representation of an oriented area in terms of an orthogonal vector has a number of disadvantages. From a cognitive point of view, it seems unintuitive, since it involves representing an area in terms of a length. From a mathematical point of view, it is only valid in three-dimensional space, and does not extend either downwards to two dimensions or upwards to four or more dimensions. Moreover, there are cases in which the representation breaks down – for instance, an oriented area changes sign under reflection in a plane whereas a vector does not.

Luckily, there is a natural way to extend a vector space to incorporate oriented areas, which is both more intuitive and more general than the cross product representation. This is done by introducing what is known as the *exterior product*. Given two vectors $\mathbf{a}$ and $\mathbf{b}$, the exterior product, written $\mathbf{a} \wedge \mathbf{b}$, is an oriented area element called a *bivector*, which can be visualized as the oriented parallelogram spanned by $\mathbf{a}$ and $\mathbf{b}$ (see Figure 3.22a). Like a vector, a bivector has a magnitude (its area) and an orientation (the plane it lies in), but no fixed location. Because it represents an oriented patch of area, a bivector can also be freely reshaped, so long as its area remains the same. Hence the bivector shown in Figure 3.22b is the same as that depicted in 3.22a. Another important property of a bivector is its *sign* (+ or $-$), which should be thought of as a direction assigned to its boundary. We



(a) The exterior product of two vectors **a** and **b** is a bivector $\mathbf{a} \wedge \mathbf{b}$.

(b) Bivectors can be reshaped so long as their area is preserved.

Figure 3.22

get a bivector of opposite sign by wedging **a** and **b** in the opposite order, giving us the relation $\mathbf{b} \wedge \mathbf{a} = -\,\mathbf{a} \wedge \mathbf{b}$. Like vectors, bivectors themselves form a vector space, which means that they can be added together and multiplied by numbers, and that these operations satisfy the vector space axioms.

The exterior product of a bivector $\mathbf{a} \wedge \mathbf{b}$ with a third vector not in the same plane **c** results in the *trivector* $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$, which can be visualized as the oriented parallelopiped spanned by **a**, **b** and **c**. Like vectors and bivectors, a trivector has a magnitude (its volume) and an orientation (its sign), and can be freely translated and reshaped as long as its magnitude remains unchanged. In a three-dimensional vector space, wedging a trivector $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$ with another vector **d** gives zero, since there is no more space available to extrude it in another dimension. Like vectors and bivectors, trivectors form a vector space – they can be added together and scalar multiplied. Multiplying a trivector by a number corresponds to scaling its weight by that number; two trivectors are added together by adding their weights.

Scalars, vectors, bivectors and trivectors are collectively called *blades*. A blade with dimension k is said to be a k-blade, where the number k is referred to as the blade's *grade*. Hence, scalars are 0-blades, vectors are 1-blades, bivectors are 2-blades, and trivectors are 3-blades. Assuming that we have four basic types *Real*, *Vector*, *Bivector*, and *Trivector*, each of which independently satisfies the vector space axioms, we can define the type of all three-dimensional blades as:

$$(282) \quad Blade := \begin{bmatrix} \mathrm{k} : Grade \\ \begin{cases} Real & \text{if } \mathrm{k} = 0 \\ Vector & \text{if } \mathrm{k} = 1 \\ Bivector & \text{if } \mathrm{k} = 2 \\ Trivector & \text{if } \mathrm{k} = 3 \end{cases} \end{bmatrix}$$

Since each grade corresponds to a vector space, *Blade* inherits the concepts of addition and scalar multiplication. The exterior product is then defined as a function $\_ \wedge \_ : Blade \to Blade$ satisfying the following axioms:

(283) • $\mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{C}$

- $\mathbf{A} \wedge (\mathbf{B} + \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) + (\mathbf{A} \wedge \mathbf{C})$
- $(\mathbf{A} + \mathbf{B}) \wedge \mathbf{C} = (\mathbf{A} \wedge \mathbf{C}) + (\mathbf{B} \wedge \mathbf{C})$
- $\mathbf{A} \wedge \mathbf{B} = (-1)^{\mathrm{grade}(\mathbf{A}) \cdot \mathrm{grade}(\mathbf{B})}\ \mathbf{B} \wedge \mathbf{A}$
- $a \wedge \mathbf{A} = \mathbf{A} \wedge a = a \cdot \mathbf{A}$

All the other properties of the exterior product follow from the above rules. For instance, it follows that the exterior product is a grade-increasing operation with the property that $\mathrm{grade}(\mathbf{A} \wedge \mathbf{B}) = \mathrm{grade}(\mathbf{A}) + \mathrm{grade}(\mathbf{B})$. See Dorst et al. (2009) for a proof of this and other properties which follow from the above axioms.

In addition to building higher-dimensional blades through the exterior product, we would also like to be able to measure the magnitude of the resulting blades. This involves extending the inner product between vectors so that it can be used to compare any two blades of the same grade. Given two k-blades, which can be factorized into vectors as $\mathbf{A} = \mathbf{a}_1 \wedge \cdots \wedge \mathbf{a}_k$ and $\mathbf{B} = \mathbf{b}_1 \wedge \cdots \wedge \mathbf{b}_k$, the inner product $\langle \mathbf{A}, \mathbf{B} \rangle$ is defined as follows:

$$(284) \quad \langle \mathbf{A}, \mathbf{B} \rangle \coloneqq \det \begin{pmatrix} \langle \mathbf{a}_1, \mathbf{b}_k \rangle & \dots & \langle \mathbf{a}_1, \mathbf{b}_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{a}_k, \mathbf{b}_k \rangle & \dots & \langle \mathbf{a}_k, \mathbf{b}_1 \rangle \end{pmatrix}$$

That is, we take the determinant of the matrix of inner products between the vector factors[1]. It can be shown that this quantity is independent of how $\mathbf{A}$ and $\mathbf{B}$ are factorized. If $\mathbf{A}$ and $\mathbf{B}$ do not have the same grade, then their inner product $\langle \mathbf{A}, \mathbf{B} \rangle$ is defined to be 0. We now use the inner product to define the magnitude of a general blade as:

$$(285) \quad \|\mathbf{A}\| \coloneqq \sqrt{\langle \mathbf{A}, \widetilde{\mathbf{A}} \rangle}$$

where $\widetilde{\mathbf{A}}$ stands for a grade-dependent change of sign, the purpose of which is to ensure that the quantity $\langle \mathbf{A}, \widetilde{\mathbf{A}} \rangle$ is positive. The magnitude of a scalar is its absolute value; the magnitude of a vector is its length; the magnitude of a bivector is its area; and the magnitude of a trivector is its volume.

Like vectors, bivectors and trivectors are primitive geometric elements which are independent of any particular choice of coordinate system. They should not be thought of as built up from lists of numbers or any other kind of object. Although we have arrived at bivectors and trivectors by combining ordinary vectors using the outer product, this does not make ordinary vectors more fundamental. Rather, scalars, vectors, bivectors and trivectors are all equally fundamental – we might just as well have started with trivectors and arrived at bivectors and vectors by considering how they can be factorized. Bivectors and trivectors are part of our intuitive spatial understanding – just as we 'see' a vector when we look at the axis of an object, or observe someone pointing at an object, so we can 'see' a bivector when we observe the spatial extent of a field, or the area swept out by a curve: note that both of these examples can be described as a kind of width – *wide field* and *wide curve*.

[1] See `https://en.wikipedia.org/wiki/Determinant` for the definition of the determinant

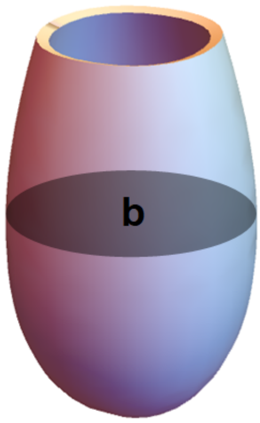


Figure 3.23: An object with a bivector axis representing its degree of area in a plane.

Recall from Section 1.3 that authors such as Bierwisch (1967) and Lang (2001) make use of 'integrated axes', which combine two or more dimensions into a single unit. We can represent this kind of two-dimensional axis by a single bivector. Unlike a vector axis, a bivector axis does not describe the extension of the object in a particular direction, but rather than amount of area it takes up in some plane. Given an individual x and a bivector b, I shall use the predicate AXIS(x, b) to mean 'A is a bivector axis of x'. We can then write the area sense of *wide* as follows:

(286) $$[\![\text{wide}]\!]_{\text{area}} \coloneqq \lambda \text{x} : \mathit{Ind} \;.\; \begin{pmatrix} \text{P} \coloneqq \begin{bmatrix} \text{b} : \mathit{Bivector} \\ \textsc{axis}(\text{x}, \text{b}) \end{bmatrix} \\ \lambda \text{p} : \text{P} \;.\; \begin{pmatrix} \text{area} \\ \|\text{p}.1\| \end{pmatrix} \end{pmatrix}$$

As shown, $[\![\text{wide}]\!]_{\text{area}}$ presupposes that x has a bivector axis b. Given such an axis, the measure function returns the magnitude of b, which is a degree on the scale of area. For an illustration of an object which can be described as *wide* in this sense, see Figure 3.23.

Because $[\![\text{wide}]\!]_{\text{area}}$ returns a degree of area, it should not be able to occur with a distance-denoting measure phrase like *10cm*, *2 meters*, *1km*, and so on. This seems correct: for instance, a field cannot be described as *500m wide* in the area sense; to say that a field is *500m wide* presupposes an axis measuring its extension in some direction (usually the direction orthogonal to an imagined observer). In languages where area width is lexicalized separately from distance width, the area term typically cannot occur with a measure phrase. For instance, Vogel writes:

> The [Swedish] noun *vidd* 'width' cannot easily combine with a measurement expressed in numbers (?*vidden är 2 meter* 'the width is 2 meters'), nor does it combine easily with a measure phrase, although there are two such instances in the corpus.

It is clear from Vogel's description of *vid* that it prototypically refers to a kind of area width: examples include *vid fjärd* 'wide bay', *vida delta* 'wide delta', *vida*

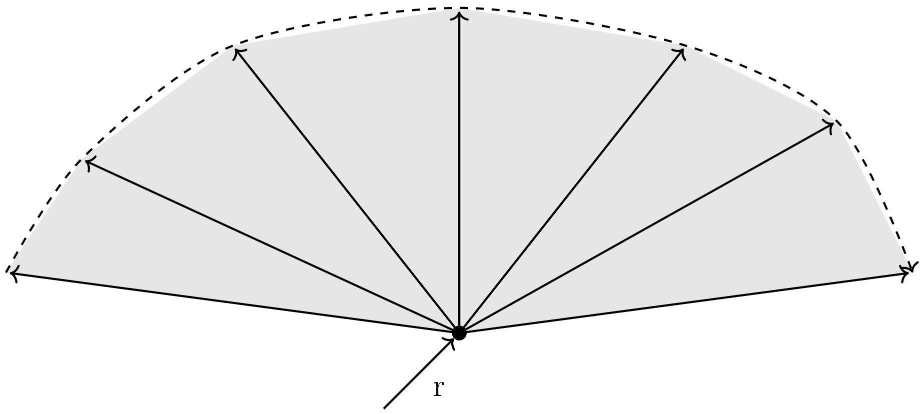


Figure 3.24: To have an arc width, an object's shape must be represented by a path of vectors which curve around a specific point.

värlen 'the wide world', *vitt uppspärrad* 'wide open', and so on. In contrast to English *wide*, Swedish *vid* cannot be used for a secondary width, e.g. ?*en vid kustrema* 'a wide coastal strip'. German *weit* 'wide' seems to behave in a similar way (see Bierwisch 1967, Lafrenz 1983, Durrell 1988). I suggest that the representation underlying these adjectives is some kind of bivector.

In addition to the straightforward area sense of width, which refers to a bivector axis, there is also a sense which refers to the area swept out by a curve. Examples include *wide arc*, *wide curve*, *wide gesture*, *wide swing*, and so on. The same sense occurs in other languages, such as Swedish *vida svängar* 'wide bends/turns' (Vogel 2004), and Polish *szeroki łuk* 'wide arc' (Linde-Usiekniewicz 2002). I shall refer to this as the 'arc' sense of *wide*. For an object to be described as *wide* in this sense, its shape must be represented by a path of vectors all centered at some point, as shown in Figure 3.24. Unlike the other kinds of path discussed in this chapter, the origin point is not chosen arbitrarily: rather, it is the specific point about which the object is understood to turn or bend. I use the predicate PATH(x, r, p) to mean 'the shape of x is represented by a path of vectors p which bend around the point r'.

Like path length, path area must be defined as a kind of upper bound to a certain measuring process. Suppose we have a path p : *Path* together with a regular partition of $[0,1]$ consisting of points $0 < t_1 < \cdots < t_{n-1} < 1$. This gives a piecewise approximation to p, like that illustrated in Figure 3.24. The area swept out by this chain of vectors is given by:

(287) $$\sum_{i=1}^{n} \frac{\|\mathrm{p}.1(t_{i-1}) \wedge \mathrm{p}.1(t_i)\|}{2}$$

That is, for each successive pair of vectors, we take half the magnitude of their exterior product, summing over all such pairs. The reason for the factor of $\frac{1}{2}$ is to get the area of the triangle rather than the parallelogram: see Figure 3.24, where the relevant areas are represented by grey triangles. The true area is defined as the supremum over all possible partitions:

(288) area : *Path* → *Real*

$$\text{area} \coloneqq \sup \sum_{i=1}^{n} \frac{\|\text{p.1}(t_{i-1}) \wedge \text{p.1}(t_i)\|}{2}$$

As for path length, people are unable to compute this quantity precisely, but can estimate it by means of a fine-grained approximation.

The arc sense of *wide* is defined as follows:

(289) $$[\![\text{wide}]\!]_{\text{arc}} \coloneqq \lambda \text{x} : \mathit{Ind}\ . \begin{pmatrix} \text{P} \coloneqq \begin{bmatrix} \text{r} : \mathit{Vector} \\ \text{p} : \mathit{Path} \\ \textsc{path}(\text{x}, \text{r}, \text{p}) \end{bmatrix} \\ \lambda \text{p} : \text{P}\ . \begin{pmatrix} \text{area} \\ \text{area}(\text{p.2}) \end{pmatrix} \end{pmatrix}$$

As indicated, $[\![\text{wide}]\!]_{\text{arc}}$ presupposes that an object is represented by a path of vectors p which curve around a point r; given such a context, the measure function returns the area of the path as defined in (288).

My impression is that $[\![\text{wide}]\!]_{\text{arc}}$ has a common join with the ordinary area sense of width, $[\![\text{wide}]\!]_{\text{area}}$. For instance, one can say something like:

(290) The curve of the bay is as wide as Lake Victoria.

where *the curve of the bay* evokes an arc width and *Lake Victoria* evokes an ordinary area width. The sense which supports this kind of comparison might be written as follows:

(291) $$[\![\text{wide}]\!]_{\text{area/arc}} \coloneqq \lambda \text{x} : \mathit{Ind}\ . \begin{pmatrix} \text{P} \coloneqq \begin{bmatrix} \text{b} : \mathit{Bivector} \\ \textsc{axis}(\text{x}, \text{b}) \end{bmatrix} + \begin{bmatrix} \text{r} : \mathit{Vector} \\ \text{p} : \mathit{Path} \\ \textsc{path}(\text{x}, \text{r}, \text{p}) \end{bmatrix} \\ \lambda \text{p} : \text{P}\ . \begin{pmatrix} \text{area} \\ \text{case p of} \\ \quad \text{inl}(\text{p}') \text{ then } \|\text{p}'.1\| \\ \quad \mid \text{inr}(\text{p}') \text{ then area}(\text{p}') \end{pmatrix} \end{pmatrix}$$

The presupposition is that x either has a bivector axis or is represented by a string of vectors emanating from a central point. In the former case, the area is given by the magnitude of the axis; in the latter case it is the area swept out by the path.

We have now completed our study of width and can collect together all of the different senses into a single network, shown in Figure 3.25. The network can be divided into three disconnected components: the distance component, the area component, and the 'scope' component, which is the metaphorical sense involved in a sentence like *John has a wide knowledge of linguistics.* All of the area senses are subsumed by $[\![\text{wide}]\!]_{\text{area/arc}}$, whereas there is no sense which subsumes all of the distance senses. It follows that some distance-related senses, such as $[\![\text{wide}]\!]_{\text{2nd}}$ and $[\![\text{wide}]\!]_{\text{latrl}}$, cannot be coordinated together. The antonymous adjective *narrow* is a near-perfect antonym of *wide*, so its lexical representation can be derived by applying the function UN to the *wide* network.

$$
\begin{tikzcd}
 & & \llbracket \text{wide} \rrbracket_{\text{2nd, horz}} \arrow[rr] \arrow[dr] & & \llbracket \text{wide} \rrbracket_{\text{2nd}} \arrow[dr] & \\
 & & & \llbracket \text{wide} \rrbracket_{\text{2nd/ribbon, horz}} \arrow[rr] & & \llbracket \text{wide} \rrbracket_{\text{2nd/ribbon}} \\
 & & \llbracket \text{wide} \rrbracket_{\text{ribbon, horz}} \arrow[rr] \arrow[ur] & & \llbracket \text{wide} \rrbracket_{\text{ribbon}} \arrow[ur] & \\
 & \llbracket \text{wide} \rrbracket_{\text{passg, horz}} \arrow[rr] \arrow[ur] & & \llbracket \text{wide} \rrbracket_{\text{passg}} \arrow[rr] \arrow[ur] & & \llbracket \text{wide} \rrbracket_{\text{obs}} \\
\llbracket \text{wide} \rrbracket_{\text{area}} \arrow[dr] & & \llbracket \text{wide} \rrbracket_{\text{arc}} \arrow[dl] & & \llbracket \text{wide} \rrbracket_{\text{latrl}} \arrow[ur] & \\
 & \llbracket \text{wide} \rrbracket_{\text{area/arc}} & & \llbracket \text{wide} \rrbracket_{\text{scope}} & &
\end{tikzcd}
$$

Figure 3.25: The wide network.

3.8 Depth

3.8.1 Introduction to depth

Like *high* and *low*, the adjectives *deep* and *shallow* are ambiguous between an intrinsic or internal sense, which refers to the distance from an object's opening to its base, and a positional sense, which refers to its distance below ground. For example, the expression *deep well* can mean either a well whose downwards extension is greater than usual, or a well which is located deep below ground. The same alternation occurs in other languages: Rakhilina (2000) writes that in Russian, the phrase *glubokie xody* 'deep subways' can mean either subways which protrude deep into the ground, or subways located at great depth. Like height, the internal and positional senses of depth cannot be coordinated unless they share the same reference plane. For instance, one cannot say something like ?*the vase is deeper than the ore*, intending to compare the internal depth of the vase to the positional depth of the ore; but it does make sense to say *the well is deeper than the ore*, comparing the extension of the well below ground to the distance of the ore deposit below ground.

Like positional height, positional depth measures distance from some contextually-specified reference plane, which is usually taken to be the surface of the earth. To have a positional depth, an object must be located below the reference plane – an object which is located *above* the reference plane is said to have a *height* rather than a depth. This creates a kind of spatial duality between the adjectives *high/low* and the adjectives *deep/shallow*, which refer to opposite sides of the reference plane. What counts as ground level can vary depending on context. For instance, when standing on the top floor of a tall building, the surface of the floor can provide a reference plane, so that one can describe objects on lower floors as *deep below*. Contrast this with a person standing at ground level outside the building, who would describe the same objects as *high above*. In certain contexts, the reference plane can even be non-horizontal. For instance, when considering the depth of an object inside a wall, the reference plane is given by the wall itself.

The internal sense of *deep/shallow* places strong restrictions on the kinds of objects it can combine with. To be described as *deep/shallow* in the internal sense, an object must be either *negative* – formed of empty space – or *hollow* – empty space bounded by a layer of solid material. Examples include *deep hole*, *deep well*, *deep vase*, *deep pocket*, *deep wardrobe*, and so on. An object which is formed entirely of solid material cannot have a depth, e.g. ?*deep tree*, ?*deep table*, ?*deep book*. Analogs of *deep/shallow* in other languages show the same behaviour: for example, Vogel (2004) writes that Swedish *djup* 'deep' and *grund* 'shallow' can describe a container, as in *djup skål* 'deep bowl', or a negative object, as in *djup grav* 'deep grave', but cannot apply to a completely solid object, as in ?*djup cykel* 'deep bicycle'. If an object is overall solid but has a salient part that is negative or hollow, then *deep/shallow* will select that part, as in *deep spoon*, *deep pan*, *deep wine glass*.

Moreover, to have an internal depth, an object must have an identifiable hole or opening through which its interior can be accessed. In her book on spatial cognition, Herskovits (1987) writes that the interior of an open container is delineated by a plane which 'closes off' the opening. For example, the interior of the bowl shown in

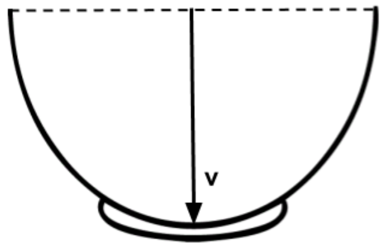


Figure 3.26: The interior of a hollow object is delineated by a virtual or actual lid. Internal depth measures the maximal orthogonal displacement v from the lid to the inside boundary of the object's base.

Figure 3.26 is bounded by the dashed plane, which seals off the bowl, making it into a closed container. Herskovits refers to this kind of imaginary plane as a *lid*. Note that the lid is always flat and cannot take the form of a curved surface. Internal depth measures the straight line distance from an object's lid to the inside boundary of its base. It answers the question: 'How far inside the object can one get by starting at its opening and travelling in a straight line orthogonal to the lid?'. The role of the lid in internal depth is analogous to the role of the ground in positional depth. As we shall see, this is what connects the two senses together in the lexical network.

Internal depth is generally directed downwards, as shown in Figure 3.26, but can also be oriented in some other direction. For example, the depth of a wardrobe, dishwasher, or oven, is evaluated from front to back, since all of these objects have an opening at the front. This is true in a wide variety of languages besides English – including German (Bierwisch 1967), Swedish (Vogel 2004), French (Spang-Hanssen 1990), Polish (Linde-Usiekniewicz 2002), Russian (Rakhilina 2000), Japanese (Shimotori 2013), and Yucatec Maya (Stolz 1996). Some objects, such as armchairs, have an opening both at the front and at the top, so expressions like *deep armchair* are ambiguous between a downwards depth and a horizontal depth. Some authors (e.g. Stolz 1996, Grzegorczykowa 1997) talk about downwards and horizontal depth as distinct senses. However, they appear to be compatible in a sentence like *the vase is deeper than the wardrobe*, where a downwards dimension is compared to a horizontal dimension. In my analysis, downwards and horizontal depth derive from two different prototypes, but nevertheless converge to a common sense.

In addition to the positional and internal senses of depth, there is also an observer-related sense which describes the axis aligned with an observer's frontal direction (Lyons 1977, Fillmore 1997). This sense occurs in sentences like *the building is 100m long, 50m high and 90m deep*, where *deep* refers to distance 'away from the viewer' or 'into the page'. The same usage occurs in some other languages: Lang writes that German *tief* 'deep' can describe an axis "along the line of sight of a (potential or actual) observer in normal position" (1989, p. 355), and Vogel (2004) shows that Swedish *djup* 'deep' can also be used for an observer axis, as in *djup hylla* 'deep shelf'. Unlike the internal sense of depth, the observer sense does not require the object to be hollow or formed of empty space. Some languages may lexicalize the internal and observer senses separately – for instance, according to Lang (2001), Korean uses *kiphi* for internal depth but *selo* for observer depth.

Finally, in addition to its spatial senses, *deep* also has a non-spatial sense which is synonymous with *profound*, as in *deep thoughts*, *deep subject*, *deep understanding*, *deep novel*, and so on. Taken together, the scope sense of *wide* and the profundity sense of *deep* are used to describe two orthogonal dimensions of an abstract subject – the number of distinct topics it embraces and the extent or explanatory power of those topics respectively. There is a possible connection with the observer-related sense, since an object which protrudes far in the direction of the observer's line of sight has hidden regions which await discovery, just as a profound subject has hidden ideas or results. As in *wide*, the canonical observer is equated with a canonical 'conceptual observer' whose gaze encompasses the subject. It is no coincidence that the two spatial adjectives which involve an observer also extend to the description of abstract bodies of knowledge.

3.8.2 Internal and positional depth

Let us begin with the concept of an internal axis, like the vector v in Figure 3.26. An internal axis is encoded by the predicate INTRNL(x, v), which can be defined roughly as follows:

(292)
- x is a hollow or empty object
- x has an actual or imaginary 'lid'
- v is orthogonal to the lid
- v describes the maximal displacement from the lid to the base of x

The reason for the 'maximal' condition is that there may be many other vectors orthogonal to the lid which end on the base of the object. For example, in Figure 3.26, there are many vectors which start at the lid of the bowl and end on its base, but only the maximal displacement **v** is counted as the depth. The concept of internal axis is intended to apply not only to containers but also to objects which are made of liquid, such as rivers, ponds and lakes. The 'lid' of such an object is given by the upper surface where it meets the air.

In a prototypical case of depth, the internal axis will also be pointed downwards in the environment. Examples include the depth of a mug, bowl, vase, basin or bucket, assuming the object is in its canonical orientation. This sense is formalized as follows:

(293) $$[\![\text{deep}]\!]_{\text{intrnl, down}} \coloneqq \lambda \text{x} : Ind\ .\ \begin{pmatrix} \text{P} \coloneqq \begin{bmatrix} \text{v} : Vector \\ \textsc{intrnl}(\text{x}, \text{v}) \\ \textsc{down}(\text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P}\ .\ \begin{pmatrix} \text{dist} \\ \|\text{p.1}\| \end{pmatrix} \end{pmatrix}$$

The presuppositions are that: (i) there is some vector v, (ii) v is the internal axis of x, and (iii) v is pointing downwards in the environment. As mentioned in the previous subsection, the downwards requirement, whilst typical, is not strictly necessary. For example, the internal axis of a wine bottle can still be described as *deep* when the bottle is placed on its side in storage. This gives us the weakened sense:

(294) $$[\![\text{deep}]\!]_{\text{intrnl}} \coloneqq \lambda \text{x} : Ind \ . \begin{pmatrix} \text{P} \coloneqq \begin{bmatrix} \text{v} : Vector \\ \textsc{intrnl}(\text{x}, \text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P} \ . \begin{pmatrix} \text{dist} \\ \|\text{p.1}\| \end{pmatrix} \end{pmatrix}$$

which is identical to $[\![\text{deep}]\!]_{\text{intrnl, down}}$ except without the DOWN(v) constraint.

In addition to the internal axis senses of *deep*, there is also the positional sense, as in *deep cave*, *deep fossil*, *deep ore deposit*. This involves two vectors, u and v, such that:

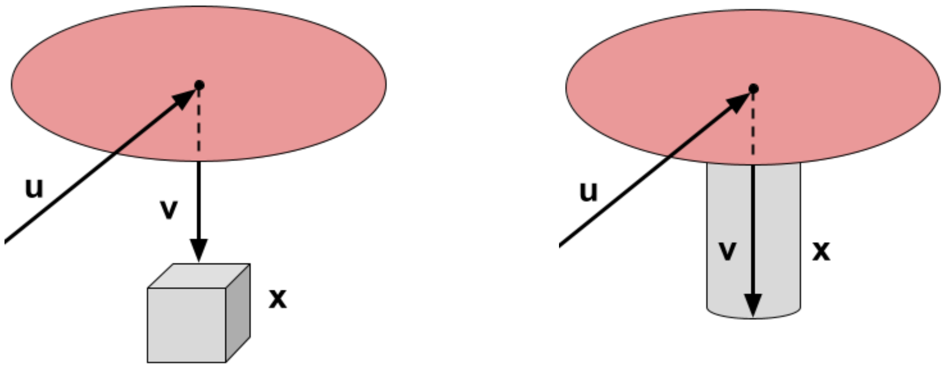
(295)
- u points to a position on the ground
- the position of x relative to u is v
- v is directed downwards

Notice that these conditions are identical to those of positional height, which was formalized back in Section 3.4.2, except that v must be pointing downwards instead of upwards. Positional depth can be written as follows:

(296) $$[\![\text{deep}]\!]_{\text{posn}} \coloneqq \lambda \text{x} : Ind \ . \begin{pmatrix} \text{P} \coloneqq \begin{bmatrix} \text{u} : Vector \\ \text{v} : Vector \\ \textsc{grnd}(\text{u}) \\ \textsc{posn}(\text{x}, \text{u}, \text{v}) \\ \textsc{down}(\text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P} \ . \begin{pmatrix} \text{dist} \\ \|\text{p.2}\| \end{pmatrix} \end{pmatrix}$$

For an illustration of positional depth, see Figure 3.27a. One can also describe as *deep* an axis which begins at ground level and extends below ground, as in Figure 3.27b. Examples of this usage include *deep burrow*, *deep root system* and *deep foundations*. I shall refer to this sense as $[\![\text{deep}]\!]_{\text{axis,grnd}}$:

(297) $$[\![\text{deep}]\!]_{\text{axis,grnd}} \coloneqq \lambda \text{x} : Ind \ . \begin{pmatrix} \text{P} \coloneqq \begin{bmatrix} \text{u} : Vector \\ \text{v} : Vector \\ \textsc{grnd}(\text{u}) \\ \textsc{axis}(\text{x}, \text{u}, \text{v}) \\ \textsc{down}(\text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P} \ . \begin{pmatrix} \text{dist} \\ \|\text{p.2}\| \end{pmatrix} \end{pmatrix}$$



(a) Positional depth below ground. **(b)** Extensional depth below ground.

Figure 3.27

Note the difference between $[\![\text{deep}]\!]_{\text{posn}}$ and $[\![\text{deep}]\!]_{\text{axis,grnd}}$: the former requires that v is the position vector of x with respect to u, whereas the latter requires that v is an axis of x based at u.

Positional depth below ground and extensional depth below ground are subsumed under a common generalization, as shown by sentences like *the ore deposit is deeper than the root system*. This exactly parallels the relationship between $[\![\text{high}]\!]_{\text{posn}}$ and $[\![\text{high}]\!]_{\text{dim,grnd}}$ which was discussed in Section 3.4.2. The common join of $[\![\text{deep}]\!]_{\text{posn}}$ and $[\![\text{deep}]\!]_{\text{axis,grnd}}$ is defined analogously to the sense $[\![\text{high}]\!]_{\text{grnd}}$, as follows:

(298) $$[\![\text{deep}]\!]_{\text{grnd}} \coloneqq \lambda \text{x} : \mathit{Ind} \,.\, \left(\begin{array}{l} \text{P} \coloneqq \begin{bmatrix} \text{u} : \mathit{Vector} \\ \text{v} : \mathit{Vector} \\ \textsc{grnd}(\text{u}) \\ \textsc{posn}(\text{x},\text{u},\text{v}) + \textsc{axis}(\text{x},\text{u},\text{v}) \\ \textsc{down}(\text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P} \,.\, \begin{pmatrix} \text{dist} \\ \|\text{p}.2\| \end{pmatrix} \end{array} \right)$$

The crucial presupposition is the fourth component, which requires either that v is the position vector of x with respect to u, as in $[\![\text{deep}]\!]_{\text{posn}}$; or that v is an axis of x based at u, as in $[\![\text{deep}]\!]_{\text{axis, grnd}}$. In either case, v is required to be pointing down and u is required to describe a point on the ground.

Finally, we can connect the senses of depth which involve ground level and those which involve an internal axis by noticing that there is a stronger sense of $[\![\text{deep}]\!]_{\text{axis,grnd}}$ where the axis is internal. Examples of objects which satisfy this sense include wells, craters, lakes and fissures – any container whose imaginary 'lid' coincides with ground level and which extends downwards into the earth. This sense is formalized as follows:

(299) $$[\![\text{deep}]\!]_{\text{intrnl, grnd}} \coloneqq \lambda \text{x} : Ind \;.\; \left(\begin{array}{l} \text{P} \coloneqq \begin{bmatrix} \text{u} : Vector \\ \text{v} : Vector \\ \textsc{grnd}(\text{u}) \\ \textsc{axis}(\text{x}, \text{u}, \text{v}) \\ \textsc{intrnl}(\text{x}, \text{v}) \\ \textsc{down}(\text{v}) \end{bmatrix} \\ \lambda \text{p} : \text{P} \;.\; \begin{pmatrix} \text{dist} \\ \|\text{p.2}\| \end{pmatrix} \end{array} \right)$$

As shown, $[\![\text{deep}]\!]_{\text{intrnl, grnd}}$ presupposes (i) a vector u, and (ii) a vector v, such that: (iii) u describes a position on the ground; (iv) v is an axis of x starting at the point u; (v) v is an internal axis of x; and (vi) v is pointing down. See Figure 3.28 for an illustration of this kind of situation.

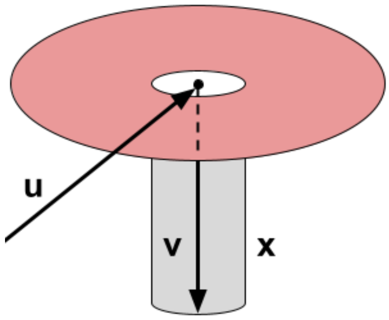


Figure 3.28: An example of $[\![\text{deep}]\!]_{\text{intrnl, grnd}}$. Note that the object's opening coincides with ground level.

The connection between the various senses of internal and positional depth is shown in Figure 3.29. There are two initial senses, $[\![\text{deep}]\!]_{\text{posn}}$, the positional depth of an object below ground, and $[\![\text{deep}]\!]_{\text{intrnl, grnd}}$, the internal depth of a container which protrudes into the ground. There are also two final senses: $[\![\text{deep}]\!]_{\text{grnd}}$, representing position or extension below ground, and $[\![\text{deep}]\!]_{\text{intrnl}}$, representing an internal axis. As before, the network makes predictions about which pairs of senses are compatible and which are incompatible. For instance, it predicts that the internal depth of a vase (a case of $[\![\text{deep}]\!]_{\text{intrnl}}$) cannot be compared with the positional depth of an ore deposit (a case of $[\![\text{deep}]\!]_{\text{posn}}$). On the other hand, the internal depth of a well or crater, which is measured from ground level, *can* be compared with the positional depth of an ore deposit, since both converge at $[\![\text{deep}]\!]_{\text{grnd}}$. The relationship between $[\![\text{deep}]\!]_{\text{posn}}$, $[\![\text{deep}]\!]_{\text{axis,grnd}}$ and $[\![\text{deep}]\!]_{\text{grnd}}$ exactly parallels the relationship between the senses $[\![\text{high}]\!]_{\text{posn}}$, $[\![\text{high}]\!]_{\text{dim,grnd}}$ and $[\![\text{high}]\!]_{\text{grnd}}$, which were discussed in Section 3.4.2.

$$
\begin{tikzcd}[column sep=small]
 & \llbracket \text{deep} \rrbracket_{\text{posn}} \arrow[dr] & \\
 & & \llbracket \text{deep} \rrbracket_{\text{grnd}} \\
 & \llbracket \text{deep} \rrbracket_{\text{axis,grnd}} \arrow[ur] & \\
\llbracket \text{deep} \rrbracket_{\text{intrnl,grnd}} \arrow[ur] \arrow[dr] & & \\
 & \llbracket \text{deep} \rrbracket_{\text{intrnl,down}} \arrow[r] & \llbracket \text{deep} \rrbracket_{\text{intrnl}}
\end{tikzcd}
$$

Figure 3.29: Internal and positional depth.

3.8.3 Observer depth

We can now move to versions of depth which involve an observer. The most prototypical version of observer depth is when an object has an opening at the front, so that its internal axis is aligned with the frontal direction of a canonical observer. For example, when standing in front of a wardrobe, one's frontal direction is aligned with the internal axis, which runs from the wardrobe's opening (the 'lid') to its back surface. Other examples of this kind of situation include looking into a building through the doorway, looking down a corridor, or looking into the mouth of a cave. This gives rise to a special case of internal depth which we might call *observed* internal depth. It is formalized as follows:

(300) $$\llbracket \text{deep} \rrbracket_{\text{intrnl, obs}} \coloneqq \lambda \text{x} : \mathit{Ind} \; . \left(\begin{array}{l} \text{P} \coloneqq \begin{bmatrix} \text{p} : \mathit{Vector} \\ \text{t} : \mathit{Vector} \\ \text{f} : \mathit{Vector} \\ \text{r} : \mathit{Vector} \\ \text{v} : \mathit{Vector} \\ \textsc{obs}(\text{x}, \text{p}, \text{t}, \text{f}, \text{r}) \\ \textsc{intrnl}(\text{x}, \text{v}) \\ \sum_{a:(0,\infty)} \text{v} = a \cdot \text{f} \end{bmatrix} \\ \lambda \text{p} : \text{P} \; . \begin{pmatrix} \text{dist} \\ \|\text{p}.1\| \end{pmatrix} \end{array} \right)$$

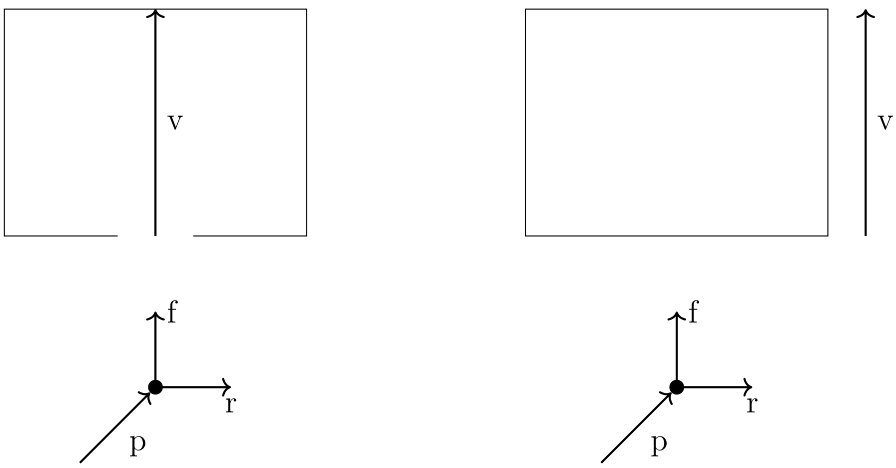
As shown, $\llbracket \text{wide} \rrbracket_{\text{intrnl, obs}}$ involves five vectors: p, the position vector of the observer; t, f and r, the top, front and right axes of the observer; and v, the depth axis itself. The constraints are that $(\text{p}, \text{t}, \text{f}, \text{r})$ is a canonical observer for x; v is the internal axis of x; and v is pointing directly forwards from the perspective of the canonical observer (in other words, there is some positive real number a such that $\text{v} = a \cdot \text{f}$). For an illustration of observed internal depth, see Figure 3.30.

Starting from observed internal depth, it is possible in English and several other languages to drop the condition that the axis is internal. This gives rise to the sense of *deep* seen in expressions like *the building is 100m long, 50m high and 90m deep*. This sense is often considered to be somewhat marginal or technical. For example, Weydt & Schlieben-Lange (1998) write that German *tief* 'deep' can be used to describe the 'third dimension' of solid rectangular objects, but describe this usage as "technical language". I include the 'pure observer' sense of depth in the lexical network, since it seems to operate fully productively, given an appropriate context.

(301) $$[\![\mathrm{deep}]\!]_{\mathrm{obs}} := \lambda \mathrm{x} : \mathit{Ind}\ .\ \left(\begin{array}{l} \mathrm{P} := \begin{bmatrix} \mathrm{p} : \mathit{Vector} \\ \mathrm{t} : \mathit{Vector} \\ \mathrm{f} : \mathit{Vector} \\ \mathrm{r} : \mathit{Vector} \\ \mathrm{v} : \mathit{Vector} \\ \textsc{obs}(\mathrm{x}, \mathrm{p}, \mathrm{t}, \mathrm{f}, \mathrm{r}) \\ \textsc{axis}(\mathrm{x}, \mathrm{v}) \\ \sum_{a:(0,\infty)} \mathrm{v} = a \cdot \mathrm{f} \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{array} \right)$$

As indicated, $[\![\mathrm{deep}]\!]_{\mathrm{obs}}$ is identical to $[\![\mathrm{deep}]\!]_{\mathrm{intrnl,\,obs}}$ except the requirement that v is an internal axis of x has been replaced by the requirement that v is simply an axis of x. For an illustration of this sense, see Figure 3.30b. As one would expect, there is an arrow from $[\![\mathrm{deep}]\!]_{\mathrm{intrnl,\,obs}}$ to $[\![\mathrm{deep}]\!]_{\mathrm{obs}}$ which forgets that the axis is internal.

The full lexical network for *deep* is given in Figure 3.31. Notice how all of the spatial senses are connected into one long chain. This chain consists of three prototypes – $[\![\mathrm{deep}]\!]_{\mathrm{posn}}$, $[\![\mathrm{deep}]\!]_{\mathrm{intrnl,\,grnd}}$ and $[\![\mathrm{deep}]\!]_{\mathrm{intrnl,\,obs}}$ – and three final senses – $[\![\mathrm{deep}]\!]_{\mathrm{grnd}}$, $[\![\mathrm{deep}]\!]_{\mathrm{intrnl}}$ and $[\![\mathrm{deep}]\!]_{\mathrm{obs}}$. In addition to the spatial senses, there is the



(a) Observed internal depth.

(b) Observed depth

Figure 3.30

non-spatial ‘profundity’ sense exemplified by *deep subject*, *deep understanding*, *deep thoughts*, and so on. I take this to be disconnected from the spatial senses since one cannot say something like ?*the vase was deep and so was the idea*. Moreover, the spatial senses require a concrete object, whereas the profundity sense requires an abstract/informational object. Nevertheless, one can imagine various metaphorical connections between $[\![\text{deep}]\!]_{\text{prfnd}}$ and the different spatial senses.

It is not clear whether the negative adjective *shallow* should be considered a perfect antonym of *deep*. On the one hand, *shallow* agrees with *deep* in most of its senses, including the internal sense (*shallow bowl*), the observer sense (*shallow wardrobe*) and the profundity sense (*shallow topic*). However, it seems to lack a counterpart to what I have called the ordinary observer sense, deep_{obs}. For instance, it seems strange to ask of two buildings “which is shallower?”, meaning which is less extended along the observer’s line of sight. The positional interpretation of *shallow* also seems less acceptable. My impression is that one can say something like *the ore deposit is very shallow*, to mean ‘not far below ground’, but I have found some disagreement on this point. Some languages seem to lack a clear antonym for depth altogether. For instance, Lafrenz (1983) and Bierwisch (1989) mention both *seicht* and *flach* as potential antonyms to German *tief* ‘deep’, with *seicht* being used mainly for liquids, and *flach* for empty space. In French, there is no direct antonym for *profond* ‘deep’; instead, one typically uses *peu profond* ‘not deep / of little depth’.

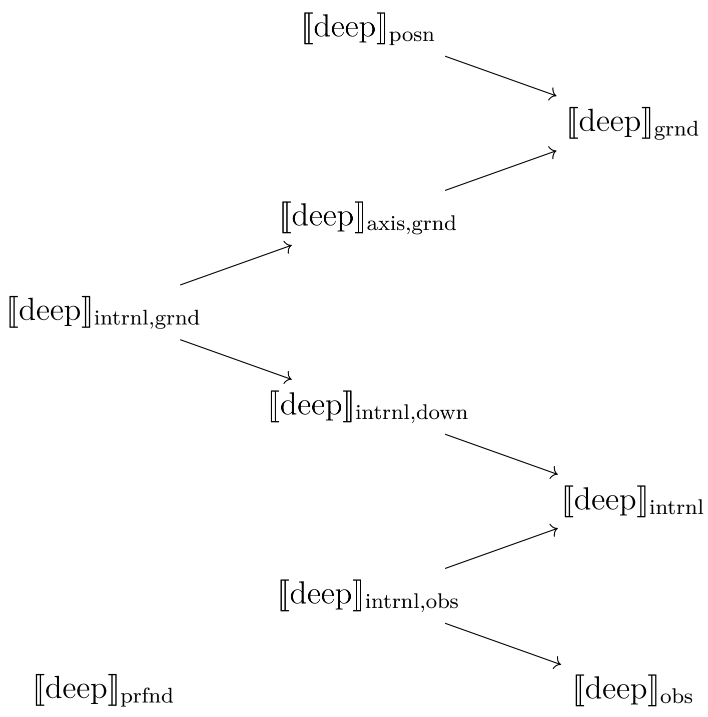


Figure 3.31: The deep network.

3.9 Thickness

3.9.1 Introduction to thickness

Unlike most of the other spatial adjectives discussed in this chapter, *thick* and *thin* do not refer to axes of an object. To understand what is meant by this, consider the combination *thick jug*. A jug has three axes – vertical, frontal and lateral – but the adjective *thick* refers not to one of these axes but to the minimal dimension of the jug's walls. Similarly, a letter of the alphabet, say the letter R, has two axes – its vertical and lateral axes – but its thickness refers to the small distance between its boundaries as shown in Figure 3.32. Another curious property of *thick* and *thin* is that they can select only part of an object. For example, in the combination *thick knife*, the adjective *thick* is typically taken to refer to the blade of the knife, rather than the handle or the entire blade + handle combination. If thick selects a particular part, then this is either the object's most functionally relevant part or a part which is significantly larger than other parts of the object.

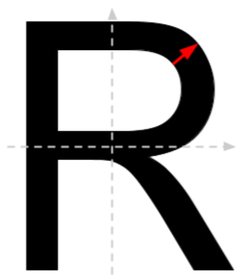


Figure 3.32: *Thick* and *thin* refer to a non-axial dimension.

The adjectives *thick* and *thin* are ambiguous between a 'cylindrical' meaning and a 'surface' meaning (Bierwisch 1967, Lyons 1977, Lang 1989, Vogel 2004). The cylindrical meaning refers to the small diameter of a long cylinder, as in *thick stick*, whereas the surface meaning refers to the minimal dimension of a surface, as in *thick plate*. Sentences which compare two cylindrical thicknesses (*the stick is thicker than the pole*) or two surface thicknesses (*the plate is thicker than the slab*) are judged as more acceptable than sentences comparing a cylindrical thickness and a surface thickness (*the stick is thicker than the plate*). Many languages distinguish lexically between cylindrical thickness and surface thickness. For example, Japanese uses *futoi* 'thick' for cylindrical thickness, but *atsui* 'thick' for surface thickness (Shimotori 2013); Yucatec Maya uses *polok* 'thick' for cylindrical thickness, but *píim* 'thick' for surface thickness (Stolz 1996); and French distinguishes *gros* 'fat/thick (cylinder)' from *épais* 'thick (surface)' (Vandeloise 1993).

In many languages, there is no distinction between the cylindrical sense of *thick* and *fat*. The *fat* sense is used not only for roughly cylindrical objects like human beings, but also for spherical objects with roughly equal dimensions, as in German *dicker Ball* 'large/fat ball', and Swedish *tjocka äpplen* 'thick/fat apples' (Vogel 2004). In languages which distinguish lexically between cylindrical thickness and surface thickness, it is always the cylindrical term which is used to mean *fat*. For example, French uses *gros* for both cylindrical thickness (*gros baton* 'thick stick') and spherical fatness (*gros ballon*) 'large/fat ball', but not for the thickness of a surface. Similarly, according to Stolz (1996), the Yucatec Maya adjective *polok* 'thick/fat' covers both cylindrical thickness and fatness, but not the thickness of a surface, which is instead described as *píim* 'thick'.

Besides the distance-related senses, *thick* can also be used to describe the resistance to penetration of a substance or collection. Here, it is important to distinguish two different senses. On the one hand, there is the use of *thick* to refer to the density of an aggregate, as in *thick forest*, *thick foliage*, *thick crowd*, *thick grass*, and so on. Roughly speaking, this refers to the number of individuals per unit area, a quantity which can vary across different parts of the aggregate. On the other hand, there is the use of *thick* to describe the viscosity or 'stickiness' of a substance, as in *thick soup*, *thick honey*, *thick syrup*, *thick mud*, and so on. Viscosity is generally thought of as an intrinsic property of a substance, like its taste or colour: it does not usually vary over different portions of the substance.

How are the different senses of *thick* connected? This question is taken up by Vandeloise (1993) in a paper entitled *The role of resistance in the meanings of thickness*. As in his previous work on length/width, Vandeloise criticises previous descriptions of thickness for only taking into account geometry and neglecting force dynamic and functional factors. He begins by distinguishing four senses of thickness, named after their French versions:

(302) a. *gros*$_1$: the diameter of a cylindrical object, e.g. *gros cigar* 'thick cigar'
b. *gros*$_2$: the size of an object with roughly equal dimensions, e.g. *grosse pouce* 'thick flea'
c. *épais*$_1$: the dimension along which a surface is most likely to break, e.g. *planche épaisse* 'thick plank'
d. *épais*$_2$: resistance to penetration of a substance or collection, e.g. *fourré épais* 'thick forest'

Senses *gros*$_1$ and *gros*$_2$ correspond to what I have called the 'cylindrical' and 'fatness' senses respectively, whereas *épais*$_1$ and *épais*$_2$ correspond to the 'surface' and 'density' senses (Vandeloise does not distinguish density and viscosity). For Vandeloise, *gros*$_1$ and *épais*$_1$ are linked together through the idea of 'resistance to breaking', since a thick cylinder is hard to deform, and so is a thick surface. Meanwhile, *épais*$_1$ and *épais*$_2$ are linked together through the concept of 'resistance to penetration', since a thick surface is difficult to penetrate, and so is a thick substance or collection.

Thick also has a number of more abstract non-spatial senses which are connected to the idea of resistance to penetration. For instance, a *thick accent* is an accent which is difficult to understand, and there is a metaphorical similarity between understanding something and penetrating or piercing an object, particularly a surface. Another metaphorical usage is the description of a person as *thick* to mean they are unintelligent. The connection here seems to be that, if a person is unintelligent, then it is difficult for new concepts to 'get through' to them, so they in some sense resistant to penetration. As with other non-spatial senses discussed in this chapter, the metaphorical similarity between these senses and their spatial counterparts is not represented by arrows in a lexical network; I take them to be diachronic explanations rather than synchronic facts which are represented in the lexicon.

3.9.2 The thick network

For an object to have a thickness, we must be able to approximate it by a lower dimensional object plus a small displacement added at every point. An object with a surface thickness, such as a slab or disk, is decomposed into a two-dimensional surface plus a small displacement; an object with a cylindrical thickness, such as a rope or wire, is decomposed into a one-dimensional curve plus a small displacement; and a small round object, such as a seed or grain of sand, is decomposed into a zero-dimensional point plus a small displacement. In each case, thickness refers to the displacement which is added in order to turn the lower-dimensional approximation into a fully-fledged volume. We might imagine adding a small ball at every point on the lower-dimensional approximation, whose diameter can vary as the shape becomes thicker or thinner. Rather than representing the thickness at every point, I shall instead assume that people represent the thickness statistically, in terms of some average thickness plus some Gaussian noise.

A cylindrical thickness is encoded using the predicate $\textsc{cyldr}(\mathrm{x}, \mathrm{p}, \mathrm{t})$, meaning 'x is approximated by a path p with average displacement t at every point'. This can be visualized by placing a sphere of diameter t at every point along p, as shown in Figure 3.33a. The cylindrical sense can be written as follows:

$$(303) \quad [\![\mathrm{thick}]\!]_{\mathrm{cyldr}} := \lambda \mathrm{x} : Ind \; . \left(\begin{array}{l} \mathrm{P} := \begin{bmatrix} \mathrm{p} : \; Path \\ \mathrm{t} : \; (0, \infty) \\ \textsc{cyldr}(\mathrm{x}, \mathrm{p}, \mathrm{t}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \; . \begin{pmatrix} \mathrm{dist} \\ \mathrm{p.2} \end{pmatrix} \end{array} \right)$$

As indicated, the presuppositions are that there is some path p and positive real number t such that x is represented by a cylinder (p, t). Given a context satisfying these conditions, the measure function returns t.

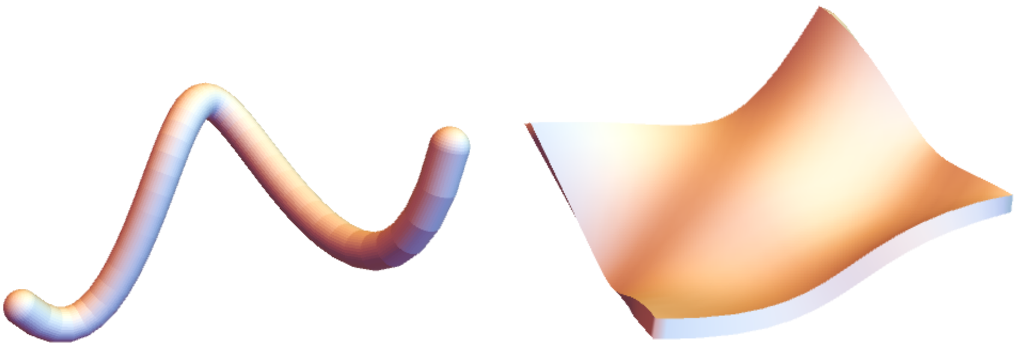
Surface thickness is similar to cylindrical thickness, except that the object is represented as a thickened surface rather than a thickened curve. Given that we are treating paths as smooth functions $[0, 1] \to Vector$, I shall take a surface to be a smooth function from $[0, 1] \times [0, 1] \to Vector$, taking points in the unit square to spatial vectors. We can write the type of surfaces as:

$$(304) \quad Surface := \begin{bmatrix} \mathrm{s} : \; [0, 1] \times [0, 1] \to Vector \\ \mathrm{smooth(s)} \end{bmatrix}$$

where smooth(s) stands for a constraint encoding the idea that s is smooth. The connection between an individual and a thickened surface is given by the predicate $\textsc{surf}(\mathrm{x}, \mathrm{s}, \mathrm{t})$, meaning 'x is approximated by a surface s with an average displacement t at every point. The most basic sense of surface thickness is implemented as follows:

$$\text{(305)} \quad [\![\text{thick}]\!]_{\text{surf}} \coloneqq \lambda x : Ind \;.\; \begin{pmatrix} P \coloneqq \begin{bmatrix} s : Surface \\ t : (0,\infty) \\ \textsc{surf}(x,s,t) \end{bmatrix} \\ \lambda p : P \;.\; \begin{pmatrix} \text{dist} \\ p.2 \end{pmatrix} \end{pmatrix}$$

See Figure 3.33b for an illustration of surface thickness.



(a) Cylindrical thickness **(b)** Surface thickness

Figure 3.33

In English, cylindrical and surface thickness are not only covered by the same lexical item, but are mutually compatible. For example, one can say something like:

(306) The stick is thicker than the plate.

where the thickness of a stick is understood in the cylindrical sense and the thickness of a plate in the surface sense. This suggests that cylindrical and surface thickness share a common join. I suggest that what joins them is the abstract idea of an object having a 'skeleton'. A skeleton is a lower-dimensional version of a shape formed from all the points which are equidistant to its boundary. The skeleton of a thickened layer of material is a two-dimensional surface; the skeleton of a long cylindrical object is a one-dimensional curve; and the skeleton of a small object with roughly equal dimensions is a zero-dimensional point. Some objects have a skeletal representation consisting of both one-dimensional and two-dimensional pieces: for example, a coffee mug consists of a surface-like bowl combined with a cylindrical handle. The concept of skeleton is commonly used in computer graphics applications, where it is generally approximated using some kind of thinning algorithm such as the medial axis transform[1].

I shall use the predicate:

(307) $x : Ind,\ S : Vector {\to} Prop,\ t : (0,\infty) \vdash \textsc{skel}(x,S,t) : Prop_1$

To mean 'x is approximated by a skeleton S, with average thickness t'. Note that the type of S is $Vector \to Prop$, that is a subset of vectors. S consists

[1]See `https://en.wikipedia.org/wiki/Medial_axis` for details

of those vectors which describe points on the skeleton of x. Skeletal thickness is then formalized as follows:

(308) $$[\![\text{thick}]\!]_{\text{skel}} := \lambda \text{x} : Ind \left(\begin{array}{l} \text{P} := \begin{bmatrix} \text{S} : Vector \rightarrow Prop \\ \text{t} : (0, \infty) \\ \textsc{skel}(\text{x}, \text{S}, \text{t}) \end{bmatrix} \\ \lambda \text{p} : \text{P} \ . \begin{pmatrix} \text{dist} \\ \text{p.2} \end{pmatrix} \end{array} \right)$$

In order for skeletal thickness to subsume both cyldrical and surface thickness, it must be possible to consider a thickened path or surface as a kind of skeleton. Given an element of *Path* or *Surface*, one can easily convert this into a predicate of vectors, as follows:

(309) $$\text{path\_to\_pred} : Path \rightarrow (Vector \rightarrow Prop)$$

$$\text{surf\_to\_pred} : Surface \rightarrow (Vector \rightarrow Prop)$$

$$\text{path\_to\_pred} := \lambda \text{P} : Path \ . \left(\lambda \text{v} : Vector \ . \left| \begin{bmatrix} \text{i} : [0,1] \\ \text{P.1(i)} = \text{v} \end{bmatrix} \right| \right)$$

$$\text{surf\_to\_pred} := \lambda \text{S} : Surface \ . \left(\lambda \text{v} : Vector \ . \left| \begin{bmatrix} \text{i} : [0,1] \\ \text{j} : [0,1] \\ \text{S.1(i,j)} = \text{v} \end{bmatrix} \right| \right)$$

That is, path_to_pred(P)(v) is true iff v lies somewhere on P, and surf_to_pred(S)(v) is true iff v lies somewhere on S. Finally, all that is required are functions:

(310) $$\sigma_1(\text{x}, \text{P}) : \textsc{path}(\text{x}, \text{P}) \rightarrow \textsc{skel}(\text{x}, \text{path\_to\_pred(P)})$$

$$\sigma_2(\text{x}, \text{S}) : \textsc{surf}(\text{x}, \text{S}) \rightarrow \textsc{skel}(\text{x}, \text{surf\_to\_pred(S)})$$

for any path P and surface S.

The proposed connections between the various senses of *thick* are illustrated in Figure 3.34. The distance-related senses appear on the bottom left and the non-distance-related senses on the top right. The distance related senses consist of the two prototypes $[\![\text{thick}]\!]_{\text{cyldr}}$ and $[\![\text{thick}]\!]_{\text{surf}}$, together with the common join $[\![\text{thick}]\!]_{\text{skel}}$, which describes a general skeletal thickness. The non-distance related senses include $[\![\text{thick}]\!]_{\text{dense}}$, which describes the density of an aggregate, e.g. *thick crowd*; and $[\![\text{thick}]\!]_{\text{visc}}$, which describes the viscosity of a substance, e.g. *thick treacle*. As shown, these are linked by a common join, $[\![\text{thick}]\!]_{\text{penetr}}$ which measures a degree of resistance to penetration. The existence of $[\![\text{thick}]\!]_{\text{penetr}}$ is motivated by the fact that $[\![\text{thick}]\!]_{\text{dense}}$ and $[\![\text{thick}]\!]_{\text{visc}}$ appear to be compatible, e.g. *the crowd was thicker than treacle*. In contrast, none of the spatial senses are compatible with either $[\![\text{thick}]\!]_{\text{visc}}$ or $[\![\text{thick}]\!]_{\text{dense}}$: one cannot say ?*the crowd was thicker than the tabletop* or ?*the treacle was thicker than the cylinder*. The adjective *thin* appears to be a perfect antonym of *thick* and is therefore given by applying UN to the *thick* network.

$$\begin{tikzcd}
\llbracket \text{thick} \rrbracket_{\text{visc}} \arrow[dr] & & \llbracket \text{thick} \rrbracket_{\text{dense}} \arrow[dl] \\
& \llbracket \text{thick} \rrbracket_{\text{penetr}} &
\end{tikzcd}$$

$$\begin{tikzcd}
\llbracket \text{thick} \rrbracket_{\text{cyldr}} \arrow[dr] & \\
& \llbracket \text{thick} \rrbracket_{\text{skel}} \\
\llbracket \text{thick} \rrbracket_{\text{surf}} \arrow[ur] &
\end{tikzcd}$$

Figure 3.34: The *thick* network.

Chapter 4

Degree Morphology Revisited

Contents

4.1 Motivation

The previous chapter showed how the lexical semantics of spatial adjectives can be formalised using lexical networks of measure functions. To get from a gradable adjective to a sentence meaning, the gradable adjective must combine with *degree morphology*. Examples of degree morphemes include the comparative (*John is taller than Mary*), the superlative (*John is the tallest*), the positive (*John is tall*), measure phrase combination (*John is 140cm tall*), *as... as...*, (*John is as tall as Andrew*), and *very* (*John is very tall*). The goal of this chapter is to develop an analysis of these constructions which is compatible with the kind of fine-grained polysemy described in Chapter 3. As explained in previous chapters, in order for the polysemy of individual words to be preserved by entire phrases, composition must map not only senses but also sense connections. I shall illustrate how most degree morphemes are monotone functions, lifting both gradable adjective senses and arrows between gradable adjectives.

Rather than proposing a completely novel theory of degree morphology, I shall instead work within the framework of degree semantics which was introduced in Chapter 1 (e.g. Cresswell 1976, von Stechow 1984, Klein 1980, Bierwisch 1989, Kennedy 1999). Degree semantics can boast a number of theoretical and empirical achievements, including explanations of polarity, cross-scalar incommensurability, and the topological restrictions associated with degree modifiers like *completely*, *mostly*, *half*. I am influenced above all by the work of Kennedy (1999, 2005, 2007), which I take to be the most detailed version of degree semantics. Following Kennedy, I treat gradable adjectives as measure functions from individuals to degrees, rather than relations between individuals and degrees, as in other authors. However, my descriptions of particular morphemes will not necessarily agree with his.

In addition to incorporating polysemy, another respect in which my analysis differs from most other versions of degree semantics is that it incorporates a dynamic account of presuppositions. Degree morphemes typically involve a lot of background information which does not appear directly in the syntax. For example, the sentence *John is the tallest person* presupposes some contextually-relevant set of people against whom John's height is measured, and requires John himself to be a member of this set. The main assertion of the sentence, namely that John's height is the greatest in the comparison set, can only be made in a context where these presuppositions are met. As explained in Chapter 2, Dependent Type Theory can be used to implement a dynamic approach to sentence meaning whereby presuppositional content, modelled as constraints on the background context, is separated from assertive content, modelled as an instruction for updating the background context with new information.

This chapter is organised as follows. Sections 4.2, 4.3, 4.4 and 4.5 are dedicated to measure phrase combination, the comparative, the superlative and the positive respectively. Each of these sections begins with an overview of previous approaches to that morpheme before presenting my own analysis. Section 4.6 covers some morphemes which appear less frequently in the literature, including *very*, *too* and *enough*. Having given an analysis of various degree morphemes, Section 4.7 addresses the question of how a language user interprets a sentence like *the wine glass is 18cm* *tall* with respect to a context. I argue that the language user does not construct all possible interpretations at once, but builds the interpretation network in a breadth-first manner beginning with the weakest or most general senses and gradually strengthening them to find the strongest assertion compatible with the context.

4.2 Measure phrase combination

4.2.1 Background

Measure phrases are expressions like *2m*, *2kg*, *2°C*, etc., which express the exact degree to which an individual possesses some property. They are combinations of a number plus an expression like *m*, *kg* or *°C*, which indicates a choice of units. All units are associated with a particular dimension: for instance, meters, kilometers, inches and yards belong to the dimension of distance; kilograms, grams, tonnes and pounds belong to the dimension of mass; and degrees Celsius, Kelvin and Fahrenheit belong to the dimension of temperature. A measure phrase can only combine with an adjective if its measure unit agrees in dimension with the adjective – one can say *2m long*, *2m high* and *2m deep* but not ?*2m hot*, ?*2m heavy*, ?*2m old*, and so on. Another important property of measure phrases is that they can only occur with positive polarity adjectives: one can say *2m long*, but not ?*2m short*, except in specific circumstances, such as in answer to the question "how short is it?".

In degree semantics, measure phrases are usually assumed to denote positive degrees. For example, *2m* would denote a positive degree on the distance scale, *2kg* a positive degree on the mass scale, *2 years* a positive degree on the age scale, and so on. One reason for thinking that measure phrases are degrees is that one can say things like *I'm not sure how long the sofa is, but it's a little longer than 2m*, where a measure phrase is commensurable with an 'unmeasured' degree whose value is unknown. Like measure phrases, unmeasured degrees make use of operations which are defined on numbers, such as multiplication, as in *some neutron stars are twice as massive as the sun*, or difference, as in *John is as much taller than Mary as he is shorter than Sally.* Such expressions either require that degrees are numerical to begin with, or else require some kind of map from the underlying structure of degrees into numbers.

Measure phrases can occur together with a wide range of degree morphology. Consider the following sentences:

(311)
a. John is 190cm tall.
b. John is 20cm taller than Mary.
c. John is 20cm less tall than Sally.
d. John is 10cm too tall.

(312)
a. * John is 190cm the tallest.
b. * Mary is 130cm the least tall.
c. * John is 190cm as tall as Henry.
d. * John is 190cm very tall.
e. * John is 5cm tall enough.
f. * The jug is 2 litres completely full.
g. * The jug is 1 litre half full.

As shown, measure phrases can occur not only with the unmarked form of an adjective (a), but also with expressions of comparison (b, c) and excess (d). However, they cannot occur in the superlative construction (a, b), nor with *as... as...* (c), *very* (d), *enough* (c), *completely* (f), or *half* (g). This section is concerned only with the unmarked form of measure phrase combination seen in (a); the occurrence of measure phrases in other constructions is explained in the sections corresponding to those constructions.

There is some disagreement in the literature about how to describe unmarked measure phrase combination. In versions of degree semantics where gradable adjectives are relations between degrees and individuals (Montague type $d \to e \to t$), a straightforward analysis is that the measure phrase saturates the degree argument of the adjective. Hence, a sentence like *John is 190cm tall* would have the interpretation:

(313) $[\![\text{John is 190cm tall}]\!] = [\![\text{tall}]\!]([\![\text{190cm}]\!])([\![\text{John}]\!])$

This is more or less the analysis given in Cresswell (1976), von Stechow (1984) and Heim (2000). One potential issue for this account, which has been pointed out by Schwarzschild (2005), is that the internal argument of an adjective usually appears to its right, as in *sick [of chocolate]*, whereas measure phrases appear on the left. An alternative analysis proposed by Klein (1980) and Kennedy & McNally (2005) is that measure phrases are really a kind of degree morpheme. On this account, rather than the adjective taking the measure phrase as an argument, the measure phrase takes the adjective as an argument, as follows:

(314) $[\![\text{John is 190cm tall}]\!] = [\![\text{190cm}]\!]([\![\text{tall}]\!])([\![\text{John}]\!])$

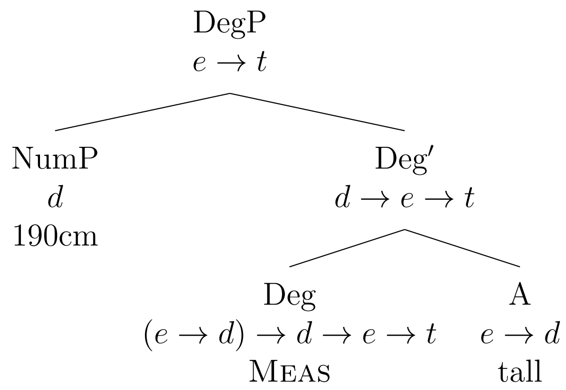
This assumes that *190cm* has something like the following interpretation (in Montague notation):

(315) $[\![\text{190cm}]\!] \coloneqq \lambda g \,.\, \lambda x \,.\, \exists d[d \geq \mathbf{190cm} \wedge g(d)(x)]$

That is, $[\![\text{190cm}]\!](g)(x)$ is true iff the degree to which x is g is greater than or equal to 190cm. The motivation for treating measure phrases as degree morphemes is that they occur in complementary distribution with other degree morphemes, such as *most/-est*, *very*, *enough*, *completely*, and so on. This analysis also explains why measure phrases appear to the left of gradable adjectives rather than to the right in the argument position. However, this account has trouble explaining why measure phrases do occur with some degree morphemes, like *more/-er*, *less* and *too*.

Both the analyses sketched above assume that gradable adjectives are relations between individuals and degrees, rather than measure functions as assumed in this thesis. If gradable adjectives are measure functions (Montague type $e \to d$), then the analysis in (313) is no longer possible since there is no degree argument for the measure phrase to saturate. The only possibility is something along the lines of (314), where the adjective functions as an argument to something else. However, rather than treat the measure phrase itself as a degree morpheme, I instead opt for an analysis found in Svenonius & Kennedy (2008) whereby a silent morpheme MEAS ('measure') takes both the adjective and the measure phrase as arguments. This allows us to preserve the idea that measure phrases themselves are degrees, which is the role they seem to play in comparatives and other constructions. An expressions like *190cm tall* is assigned a syntactic structure which parallels that of a comparative or superlative expression:

(316)



where MEAS is a silent morpheme with the following semantics:

(317) $[\![\textsc{meas}]\!] := \lambda g \,.\, \lambda d \,.\, \lambda x \,.\, g(x) \geq d$

That is, $[\![\textsc{meas}]\!]$(G)(d)(x) is true iff the degree to which x is G is at least as large as d. Notice that, on this approach the 'greater than or equal two' relation is introduced by MEAS rather than by the adjective or the degree morpheme. If one is unhappy with silent morphemes, then one can instead think about MEAS as a kind of type-shifting rule whose role is to lift a gradable adjective of type $e \to d$ to a relation of type $d \to e \to t$.

4.2.2 Action on senses

Before presenting my formulation of the MEAS morpheme, it is important to review the analysis of degrees introduced in the previous chapter. Recall that, for every scale s, there is a type of positive s-degrees and a type of negative s-degrees, written $Degree(s,+)$ and $Degree(s,-)$ respectively. The types $Degree(s,+)$ and $Degree(s,-)$ have associated linear orders $\leq_{(s,+)}$ and $\leq_{(s,-)}$. Two degrees on different scales or with different polarities cannot be compared, because one cannot form the proposition that one is greater than the other. Positive degrees and negative degrees on the same scale are isomorphic and have opposite orders. The type of all positive degrees on all scales is defined as the dependent sum:

(318) $$Degree(+) := \begin{bmatrix} \mathrm{s} : Scale \\ Degree(\mathrm{s},+) \end{bmatrix}$$

and the type of all negative degrees is defined likewise. All measure phrases belong to the type $Degree(+)$. In addition to the types $Degree(+)$ and $Degree(-)$, there is also a type of all degrees, defined as follows:

(319) $$Degree := \begin{bmatrix} \mathrm{p} : Pol \\ \mathrm{s} : Scale \\ Degree(\mathrm{s},\mathrm{p}) \end{bmatrix}$$

There are two kinds of gradable adjective: positive adjectives, which return positive degrees, and negative adjectives, which return negative degrees. These are represented by the types:

(320)
- $Gradable(+) \coloneqq Ind \rightarrow \begin{bmatrix} \mathrm{P} : Type \\ \mathrm{P} \rightarrow Degree(+) \end{bmatrix}$
- $Gradable(-) \coloneqq Ind \rightarrow \begin{bmatrix} \mathrm{P} : Type \\ \mathrm{P} \rightarrow Degree(-) \end{bmatrix}$

Finally, there is the type of all gradable adjectives, which is:

(321) $Gradable \coloneqq Ind \rightarrow \begin{bmatrix} \mathrm{P} : Type \\ \mathrm{P} \rightarrow Degree \end{bmatrix}$

Any element of *Gradable*$(+)$ or *Gradable*$(-)$ can be automatically considered as an element of *Gradable* by taking its output degree and inserting the label $+$ or $-$ respectively.

We can now turn to the semantics of MEAS, beginning with the 'greater than or equal to' interpretation, which I shall call $[\![\text{MEAS}]\!]_{\geq}$. This takes a positive gradable adjective, a positive degree, and an individual, and returns a sentence meaning (context update). It is therefore an element of:

(322) $[\![\text{MEAS}]\!]_{\geq} : Gradable(+) \rightarrow Degree(+) \rightarrow Ind \rightarrow Update$

It can be implemented as follows:

(323)
$$[\![\text{MEAS}]\!]_{\geq} \coloneqq \lambda \mathrm{G} : Gradable(+)\ .\ \lambda \mathrm{d} : Degree(+)\ .\ \lambda \mathrm{x} : Ind\ .\ \begin{pmatrix} \mathrm{P} \coloneqq \mathrm{G(x).1} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \mathrm{G(x).2(p).2} \geq \mathrm{d.2} \end{pmatrix}$$

As shown, given some gradable adjective G, degree argument d, and individual x, $[\![\text{MEAS}]\!]_{\geq}$ returns a context update which presupposes that x satisfies the presuppositions of G. Given a context satisfying these conditions, the main assertion is that the magnitude of G applied to x is greater than or equal to the magnitude of d. This is a valid comparison if and only if both degrees have the same polarity and dimensions.

To illustrate how $[\![\text{MEAS}]\!]_{\geq}$ acts on senses, it is useful to consider an example. The following is a derivation of *190cm tall*, assuming the strongest interpretation of *tall*, namely $[\![\text{tall}]\!]_{\text{up, 1st, large}}$:

(324) $[\![\text{190cm MEAS tall}]\!]_{\geq,\,\text{proto}}$

$$= [\![\text{MEAS}]\!]_{\geq}([\![\text{tall}]\!]_{\text{up, 1st, large}})([\![\text{190cm}]\!])$$

$$
= \begin{bmatrix} \lambda \mathrm{G} : \mathit{Gradable}(+) \ . \\ \quad \lambda \mathrm{d} : \mathit{Degree}(+) \ . \\ \quad\quad \lambda \mathrm{x} : \mathit{Ind} \ . \\ \quad\quad\quad \begin{pmatrix} \mathrm{P} := \mathrm{G}(\mathrm{x}).1 \\ \lambda \mathrm{p} : \mathrm{P} \ . \ \mathrm{G}(\mathrm{x}).2(\mathrm{p}).2 \geq \mathrm{d}.2 \end{pmatrix} \end{bmatrix} \begin{pmatrix} \lambda \mathrm{x} : \mathit{Ind} \ . \\ \quad \begin{pmatrix} \mathrm{P} := \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ \textsc{up}(\mathrm{v}) \\ \textsc{1st}(\mathrm{x}, \mathrm{v}) \\ \textsc{large}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \begin{pmatrix} \mathrm{dist} \\ \|\mathrm{p}.1\| \end{pmatrix} \end{pmatrix} \end{pmatrix} \begin{pmatrix} \mathrm{dist} \\ 190\mathrm{cm} \end{pmatrix}
$$

$$
= \begin{bmatrix} \lambda \mathrm{d} : \mathit{Degree}(+) \ . \\ \quad \lambda \mathrm{x} : \mathit{Ind} \ . \\ \quad\quad \begin{pmatrix} \mathrm{P} := \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ \textsc{up}(\mathrm{v}) \\ \textsc{1st}(\mathrm{x}, \mathrm{v}) \\ \textsc{large}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \ \|\mathrm{p}.1\| \geq \mathrm{d}.2 \end{pmatrix} \end{bmatrix} \begin{pmatrix} \mathrm{dist} \\ 190\mathrm{cm} \end{pmatrix}
$$

$$
= \lambda \mathrm{x} : \mathit{Ind} \ . \begin{pmatrix} \mathrm{P} := \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ \textsc{up}(\mathrm{v}) \\ \textsc{1st}(\mathrm{x}, \mathrm{v}) \\ \textsc{large}(\mathrm{v}) \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \ . \ \|\mathrm{p}.1\| \geq 190\mathrm{cm} \end{pmatrix}
$$

The result takes an individual x and returns a context update which presupposes that x satisfies the presuppositions of tall – that is, it has an upwards-directed axis which is maximal and comparable in size to a human height. As one would expect, the main assertion is that the magnitude of x's vertical axis is greater than or equal to 190cm.

There are several ways in which measure phrase modification can 'go wrong'. For example, suppose that instead of *190cm tall*, we have ?*190cm short*. This leads to a type error because MEAS requires a positive gradable adjective, but *short* is negative. Now suppose that instead of *190cm tall*, we have ?*190<u>kg</u> tall*. Now the polarity is correct, but the resulting assertion contains a type error because one cannot compare a distance and a weight. Finally, another way in which measure phrase modification can go wrong is if the subject does not satisfy the background presuppositions of the adjective. For instance, *190cm tall* will be infelicitous if the subject is not the kind of thing which can have a vertical axis, such as an abstract object or event, and will also fail if the subject is a physical object without the necessary shape, such as a coin or football.

4.2.3 Action on arrows

To act on networks, MEAS should lift not only senses but also arrows. Suppose that we have two positive gradable adjectives $\mathrm{G}, \mathrm{H} : \mathit{Gradable}(+)$. Recall from the previous chapter that an arrow α from G to H is an element of:

(325) $\alpha : \mathrm{G} \rightarrow_{Gradable(+)} \mathrm{H}$

$$: \prod_{\mathrm{x}:Ind} \begin{bmatrix} \mathrm{f} : \mathrm{G(x).1} \rightarrow \mathrm{H(x).1} \\ \prod_{\mathrm{p:G(x).1}} \mathrm{G(x).2(p)} = \mathrm{H(x).2(f(p))} \end{bmatrix}$$

That is, it specifies, for every individual x, a map from the presuppositions of G(x) to the presuppositions of H(x) which commutes with the two measure functions. We would now like to lift α to get an arrow from $[\![\textsc{meas}]\!]_{\geq}(\mathrm{G})$ to $[\![\textsc{meas}]\!]_{\geq}(\mathrm{H})$. Given that $[\![\textsc{meas}]\!]_{\geq}(\mathrm{G})$ and $[\![\textsc{meas}]\!]_{\geq}(\mathrm{H})$ are both of type $Degree(+) \rightarrow Ind \rightarrow Update$, the lifted arrow should be an element of:

(326) $[\![\textsc{meas}]\!]_{\geq}(\alpha) : \prod_{\mathrm{d}:Degree(+)} \prod_{\mathrm{x}:Ind} [\![\textsc{meas}]\!](\mathrm{G})(\mathrm{d})(\mathrm{x}) \rightarrow_{Update} [\![\textsc{meas}]\!](\mathrm{H})(\mathrm{d})(\mathrm{x})$

This is implemented as follows:

(327) $[\![\textsc{meas}]\!]_{\geq}(\alpha) \coloneqq \lambda\mathrm{d} : Degree(+)$

$$\begin{aligned} &\lambda \mathrm{x} : Ind \ . \\ &\quad \text{let } \mathrm{G}' \coloneqq [\![\textsc{meas}]\!]_{\geq}(\mathrm{G})(\mathrm{d})(\mathrm{x}) \text{ in} \\ &\quad \begin{pmatrix} \lambda \mathrm{p} : \mathrm{G}'.1 \ . \ \alpha(\mathrm{x}).1(\mathrm{p}.1) \\ \lambda \mathrm{p} : \mathrm{G}'.1 \ . \ \mathrm{id}_{\mathrm{G}'.2(\mathrm{p})} \end{pmatrix} \end{aligned}$$

The key point to notice is that the lifted arrow works by using the original arrow α to replace the presuppositions of G with presuppositions of H. The asserted content is left unchanged, so the assertion map is simply the identity.

Now that we know how $[\![\textsc{meas}]\!]_{\geq}$ acts on arrows, we know how it acts on an entire network. For example, beginning with the *tall* network, in which every sense has type *Gradable*(+), we can apply $[\![\textsc{meas}]\!]_{\geq}$ to every sense and arrow, giving us a network in which every sense has type $Degree(+) \rightarrow Ind \rightarrow Update$. We can then take the resulting network and apply it to a positive degree, giving a network of $Ind \rightarrow Update$ senses. Finally, we can apply an individual, giving us a simple *Update* network which can be used to update a context. Every step in this process is monotone, meaning it lifts arrows and preserves their direction. Figure 4.1 shows the result of applying $[\![\textsc{meas}]\!]_{\geq}$ to the entire *tall* network.

4.2.4 The exact equality reading

Thus far, we have ignored an important observation, which is that the measure morpheme MEAS is itself polysemous between two different interpretations: an 'at least' interpretation, represented by $[\![\textsc{meas}]\!]_{\geq}$, and a 'strictly equal to' interpretation. For example, consider the following two dialogues:

(328) a. Bob: How tall are you?
Alice: I am 95cm tall.

b. Bob: To go on this ride, you must be 95cm tall.
Alice: I am 95cm tall.

$$
\begin{tikzcd}
& {[\![\mathrm{tall}]\!]_{\mathrm{up,\,large}}} \arrow[dr] \arrow[dddr] & & \\
{[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st,\,large}}} \arrow[ur] \arrow[dr] \arrow[dddr] & & {[\![\mathrm{tall}]\!]_{\mathrm{up,\,sgfnt}}} \arrow[dddr] & \\
& {[\![\mathrm{tall}]\!]_{\mathrm{up,\,1st}}} \arrow[ur] \arrow[dddr] & & \\
& & {[\![\mathrm{tall}]\!]_{\mathrm{vert,\,large}}} \arrow[dr] & \\
& {[\![\mathrm{tall}]\!]_{\mathrm{vert,\,1st,\,large}}} \arrow[ur] \arrow[dr] & & {[\![\mathrm{tall}]\!]_{\mathrm{vert,\,sgfnt}}} \\
& & {[\![\mathrm{tall}]\!]_{\mathrm{vert,\,1st}}} \arrow[ur] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
{} \arrow[d, mapsto, "{[\![\textsc{meas}]\!]_{\geq}}"] \\
{}
\end{tikzcd}
$$

$$
\begin{tikzcd}
& {[\![\textsc{meas}]\!]_{\geq}([\![\mathrm{tall}]\!]_{\mathrm{up,\,large}})} \arrow[dr] \arrow[dddr] & & \\
{[\![\textsc{meas}]\!]_{\geq}([\![\mathrm{tall}]\!]_{\mathrm{up,\,1st,\,large}})} \arrow[ur] \arrow[dr] \arrow[dddr] & & {[\![\textsc{meas}]\!]_{\geq}([\![\mathrm{tall}]\!]_{\mathrm{up,\,sgfnt}})} \arrow[dddr] & \\
& {[\![\textsc{meas}]\!]_{\geq}([\![\mathrm{tall}]\!]_{\mathrm{up,\,1st}})} \arrow[ur] \arrow[dddr] & & \\
& & {[\![\textsc{meas}]\!]_{\geq}([\![\mathrm{tall}]\!]_{\mathrm{vert,\,large}})} \arrow[dr] & \\
& {[\![\textsc{meas}]\!]_{\geq}([\![\mathrm{tall}]\!]_{\mathrm{vert,\,1st,\,large}})} \arrow[ur] \arrow[dr] & & {[\![\textsc{meas}]\!]_{\geq}([\![\mathrm{tall}]\!]_{\mathrm{vert,\,sgfnt}})} \\
& & {[\![\textsc{meas}]\!]_{\geq}([\![\mathrm{tall}]\!]_{\mathrm{vert,\,1st}})} \arrow[ur] &
\end{tikzcd}
$$

Figure 4.1: Applying $[\![\textsc{meas}]\!]_{\geq}$ to the entire *tall* network.

In dialogue (a), Bob takes Alice to mean that she is *exactly* 95cm tall, whereas in (b) this strong interpretation does not arise, and Bob takes Alice to mean that she is *at least* 90cm tall, this being the requisite height to go on the ride. As this example illustrates, the exact equality reading is more typical, with the 'at least' reading only arising in a context like (b) where exact equality is not in question.

Many authors take the 'at least' interpretation to be the basic meaning of the MEAS morpheme, appealing to scalar implicature to explain the exact equality interpretation (e.g. Cresswell 1976, Klein 1980, Heim 2000, Kennedy & McNally 2005). The idea is that, in a dialogue like (328a), Bob assumes that Alice is being as informative as possible. He reasons that if Alice was taller than 95cm she would have said so: therefore she must be exactly 95cm tall. In (b), on the other hand, for Alice to report her exact height would not be more informative, since the question at issue is whether or not she is tall enough to go on the ride. If this could be made to work, then there would be no need to encode the exact equality reading into the meaning of MEAS, as it would follow from general pragmatic principles.

In my view, the difficulty with a purely pragmatic analysis of the exact equality reading is explaining which stronger alternative is supposed to generate this implicature. A scalar implicature such as:

(329) John ate some of the apples.
$\Rightarrow$ John did not eat all of the apples.

is generated by the presence of the stronger alternative *John ate all of the apples*, which is given by a straightforward substitution *some* $\mapsto$ *all*. In contrast, the inference in (328a) that Alice is exactly 95cm tall cannot be easily generated by a single alternative. Consider the following possibilities:

(330)
a. I am 96cm tall.
b. I am 97cm tall.
c. I am 100cm tall.
d. I am more than 95cm tall.
e. ...

There are an infinite number potential alternatives of the form (a), (b), (c), none of which can generate the right inference individually. For instance, $\neg$(a) combined with the original utterance leads to the inference that Alice's height is somewhere in the range $[95\text{cm}, 96\text{cm})$. Moreover, (d) is ruled out by its complexity, since the usual assumption is that only alternatives which are at most as complex as the original can give rise to implicatures (e.g. Katzir 2007). (If this was not the case, then arbitrary sentences which entail the original, e.g. *I am 95cm tall and a basketball player*, would give rise to implicatures.)

For this reason, I assume that the MEAS morpheme is ambiguous between a weaker 'at least' interpretation and a stronger exact equality interpretation. It is the presence of this stronger interpretation, *in combination* with the assumption that the speaker is being as informative as possible, which generates the scalar implicature in (328a) and similar contexts. Alongside $[\![\text{MEAS}]\!]_{\geq}$, then, there is also an exact equality sense $[\![\text{MEAS}]\!]_{=}$ with the following implementation:

(331) $$
\begin{aligned}
[\![\textsc{meas}]\!]_{=} \coloneqq \lambda \mathrm{G} : &\mathit{Gradable}(+)\ . \\
&\lambda \mathrm{d} : \mathit{Degree}(+)\ . \\
&\quad \lambda \mathrm{x} : \mathit{Ind}\ . \\
&\quad\quad \begin{pmatrix} \mathrm{P} \coloneqq \mathrm{G(x).1} \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \mathrm{G(x).2(p).2} = \mathrm{d.2} \end{pmatrix}
\end{aligned}
$$

As shown, the only difference between $[\![\textsc{meas}]\!]_{\geq}$ and $[\![\textsc{meas}]\!]_{=}$ is that the assertion that the value of d′ is less than or equal to the value of d is replaced by the assertion that it is strictly equal. $[\![\textsc{meas}]\!]_{=}$ also acts on arrows in exactly the same way as $[\![\textsc{meas}]\!]_{\geq}$. As one would expect, there is an arrow from $[\![\textsc{meas}]\!]_{=}$ to $[\![\textsc{meas}]\!]_{\geq}$ which works by weakening the strict equality requirement to a greater than or equal to requirement. This arrow, which I shall label $[\![\textsc{meas}]\!]^{\geq}_{\equiv}$, has the following type:

(332) $$
\begin{aligned}
[\![\textsc{meas}]\!]^{\geq}_{\equiv} : \textstyle\prod_{\mathrm{G}:\mathit{Gradable}(+)} &\\
&\textstyle\prod_{\mathrm{d}:\mathit{Degree}(+)} \\
&\quad \textstyle\prod_{\mathrm{x}:\mathit{Ind}} \\
&\quad\quad [\![\textsc{meas}]\!]_{=}\mathrm{(G)(d)(x)} \rightarrow_{\mathit{Update}} [\![\textsc{meas}]\!]_{\geq}\mathrm{(G)(d)(x)}
\end{aligned}
$$

It is implemented as follows, assuming that the type $d_1 \leq d_2$ is defined as $d_1 = d_2 + d_1 > d_2$:

(333) $$
\begin{aligned}
[\![\textsc{meas}]\!]^{\geq}_{\equiv} \coloneqq \lambda \mathrm{G} : &\mathit{Gradable}(+) \\
&\lambda \mathrm{d} : \mathit{Degree}(+) \\
&\quad \lambda \mathrm{x} : \mathit{Ind}\ . \\
&\quad\quad \mathrm{let\ M} \coloneqq [\![\textsc{meas}]\!]_{=}\mathrm{(G)(d)(x)\ in} \\
&\quad\quad\quad \begin{bmatrix} \mathrm{id}_{\mathrm{M.1}} \\ \lambda \mathrm{p} : \mathrm{M.1}\ .\ \lambda \mathrm{q} : \mathrm{M.2(p)}\ .\ \mathrm{inl(q)} \end{bmatrix}
\end{aligned}
$$

The presuppositions can be copied without alteration, since they are the same for $[\![\textsc{meas}]\!]_{=}$ and $[\![\textsc{meas}]\!]_{\geq}$. The two senses $[\![\textsc{meas}]\!]_{=}$ and $[\![\textsc{meas}]\!]_{\geq}$, together with the arrow between them, form a very simple lexical network:

(334) $$[\![\textsc{meas}]\!]_{=} \longrightarrow [\![\textsc{meas}]\!]_{\geq}$$

4.3 The comparative

4.3.1 Background

The comparative refers to the morphemes *more/-er*[1] and *less*, which involve the comparison of two degrees. A comparative sentence like *John is taller than Mary* consists of at least three parts: (a) an individual of which the comparative expression is predicated; (b) a gradable adjective which is applied to this individual to get a reference degree; and (c) a *than*-expression which introduces the comparison degree against which the reference degree is compared. Another optional feature of a comparative sentence is the 'difference degree' argument, which describes the difference between the reference degree and the standard of comparison, as in *John is <u>10cm</u> taller than Mary.* As one would expect, the reference degree, standard of comparison and difference degree (if present) must all agree in dimensions.

It is useful to distinguish between different types of comparative depending on the type of the *than* argument. Consider the following:

(335) a. John is taller than <u>Mary</u>.
b. John is taller than <u>150cm</u>.
c. John is taller than <u>the car is wide</u>.

In (a), the complement of *than* is a noun phrase, in (b) it is a measure phrase, and in (c) it is an entire clause. I shall refer to these as nominal, measure phrase, and clausal comparatives respectively.

Most analyses of the comparative group together measure phrase comparatives and clausal comparatives as sharing the same underlying structure, as follows:

(336)
- DegP $e \to t$
  - Deg′ $d \to e \to t$
    - Deg $(e \to d) \to d \to e \to t$ more/-er
    - A $e \to d$ tall
  - PP d than 150cm / than the car is wide

where the comparative morpheme *more/-er* is interpreted as:

(337) $[\![\text{more/-er}]\!]_1 \coloneqq \lambda g \,.\, \lambda d \,.\, \lambda x \,.\, g(x) > d$

As shown, the *than* argument in both measure phrase and clausal comparatives is assumed to denote a degree. In a measure phrase comparative, this degree is provided explicitly by the measure phrase, whereas in clausal comparatives it is

[1] The difference between *-er* and *more* is purely morphophonemic, having to do with the number of syllables in the adjective – it has no effect on the semantics

provided indirectly by means of a definite description (e.g. von Stechow 1984, Heim 1985, Moltmann 1992, Izvorski 1995). The major challenge with this account is explaining how an expression like *than the car is wide* functions as a description of a degree. The usual explanation is that such expressions are headed by a silent measurement morpheme – the morpheme which we are calling MEAS. The degree argument of MEAS is bound by a silent operator which is attached at the clause level like a wh-operator, giving us the following structure:

(338) $[_{\text{PP}}$ than $[_{\text{CP}}$ OP_{x} the car is $[_{\text{DegP}}$ d_{x} MEAS wide$]]]$

One observation which supports this analysis is that a clausal comparative may not contain an explicit degree morpheme inside the *than* clause, suggesting that the DegP position is already occupied:

(339) a. * John is taller than the car is very wide.
b. * John is taller than the car is too wide.
c. * John is taller than the car is wide enough.

The semantic role of Op_{x} is to abstract over the bound degree argument d_{x} to give a property of degrees, and then perform maximization over this property, as follows:

(340) $[\![\text{than OP}_{\text{x}} \text{ the car is d}_{\text{x}} \text{ MEAS wide}]\!] = \max(\lambda d\,.\,[\![\text{the car is d MEAS wide}]\!])$

giving us the maximum degree to which the car is wide. (The reason for the maximization is there there is not always one unique degree which satisfies the property described by the *than* clause.)

In nominal comparatives like *John is taller than Sally*, the complement of *than* appears to be a simple noun phrase. A number of authors have proposed that nominal comparatives in fact originate from clausal sources (Bresnan 1973, Lerner & Pinkal 1995, Hazout 1995). On this view, a sentence like *John is taller than Sally* would be an elided form of something like *John is taller than Sally is tall*. However, as Kennedy (1999) has pointed out, there are good reasons to think that the structure of nominal comparatives is just as it appears on the surface, with the complement of *than* being an ordinary noun phrase. For example, in a nominal comparative, the *than* expression can contain a reflexive pronoun bound by the subject, but this is not possible in a clausal comparative:

(341) a. No man is older than himself.
b. * No man is older than himself is.

Moreover, nominal comparatives allow the complement of *than* to be bound by a wh-operator, whereas material in a clausal comparative cannot be bound in this way:

(342) a. Who_{i} did you say John is taller than t_{i}?
b. * Who_{i} did you say John is taller than t_{i} thought?

This suggests an analysis whereby *than Sally* denotes an individual, as follows:

(343)

- DegP, $e \to t$
  - Deg′, $e \to e \to t$
    - Deg, $(e \to d) \to e \to e \to t$, more/-er
    - A, $e \to d$, tall
  - PP, e, than Sally

To deal with the differently typed *than* argument, this account requires an alternative version of *more/-er* with something like the following denotation:

(344) $[\![\text{more/-er}]\!]_2 \coloneqq \lambda g \,.\, \lambda y \,.\, \lambda x \,.\, g(x) > g(y)$

that is, $[\![\text{more/-er}]\!]_2$(g)(y)(x) holds iff the degree to which x is g exceeds the degree to which y is g.

Among previous approaches to the comparative, there has been little attention paid to the optional difference degree argument. The standard Montague type system does not allow for optional arguments, instead requiring a distinct version of the comparative morpheme for each possible combination. The presence vs. absence of a difference degree argument, combined with the distinction between clausal and nominal comparatives, would necessitate four different versions. This is intuitively unsatisfying, however: it suggests that sentences like *John is taller than Mary*, *John is 15cm taller than Mary*, *John is taller than Susan believes* and *John is 15cm taller than Susan believes* all involve different senses of *more/-er*. As we shall see, one advantage of using a richer type system is that it allows us to unify all four versions into the same semantic representation.

We have already discussed the constraint that the dimensions of the reference degree, comparison degree, and difference degree must all be the same. What of the polarity? Here the situation is slightly more complicated. In nominal comparatives, the polarity of the reference and comparison degree are automatically the same, because they result from applying the same measure function. In the case of clausal and measure phrase comparatives, however, the polarity of the reference and comparison degree can be different. Consider the following examples:

(345)
a. John is taller (/ less tall) than the car is wide. [positive, positive]
b. John is shorter (/ less short) than the car is wide. [negative, positive]
c. ?John is taller (/ less tall) than the car is narrow. [positive, negative]
d. ?John is shorter (/ less short) than the car is narrow. [negative, negative]

As shown, the reference degree can be either positive or negative, whereas the comparison degree must be positive. This is compatible with the theory that the sentences in (345) contain a hidden MEAS morpheme, since MEAS can only be applied to a positive degree, as discussed in Section 4.2.

4.3.2 Action on senses

Having introduced the comparative in terms of Montague semantics, let us now turn to its formulation in the type system used in this thesis. Following the usual analysis, I shall assume that the type of the *than* argument is Ind in nominal comparatives, and $Degree(+)$ in clausal and measure phrase comparatives. I shall also assume that clausal comparatives contain the silent measure morpheme MEAS, but shall leave open the question of how exactly the *than* clause is derived, whether by movement or some other mechanism for describing long-distance dependencies. One difference with previous accounts of the comparative will be the use of sum types to encode both the choice between the two different types of *than* argument and the optionality of the difference degree argument. The type of the *than* argument is:

(346) $Ind + Degree(+)$

an element of which is either an individual or a positive degree. The type of the difference degree argument is:

(347) $Degree(+)? \coloneqq Degree(+) + \top$

an element of which is either a positive degree, or the unique value $* : \top$, which is used to represent a null argument. Given a type A, the type $A? \coloneqq A + \top$ is known as an *option type*, since it is used to represent an optional element of type A.

The use of sum types to encode argument options has the advantage of avoiding an unnecessary proliferation of senses. However, the price we pay for this unification is a more complex theory of the syntax-semantics interface. Instead of a distinct lexical entry for each version of *more/-er*, we instead have a single lexical entry with several distinct composition options. We might represent this somewhat informally as follows:

(348) *more/-er*

a. Nominal comparative, with difference degree:
   syntax: $[_{\text{DegP}}$ NumP $[_{\text{Deg}'}$ $[_{\text{Deg}'}$ __ A $]$ $[_{\text{PP}}$ than NP$]$ $]$ $]$
   semantics: $[\![\text{more/-er}]\!]([\![\text{A}]\!])(\text{inl}([\![\text{NP}]\!]))(\text{inl}([\![\text{NumP}]\!]))$

b. Nominal comparative, no difference degree:
   syntax: $[_{\text{DegP}}$ $[_{\text{Deg}'}$ __ A $]$ $[_{\text{PP}}$ than NP$]$ $]$
   semantics: $[\![\text{more/-er}]\!]([\![\text{A}]\!])(\text{inl}([\![\text{NP}]\!]))(\text{inr}(*))$

c. Clausal comparative, with difference degree:
   syntax: $[_{\text{DegP}}$ NumP $[_{\text{Deg}'}$ $[_{\text{Deg}'}$ __ A $]$ $[_{\text{PP}}$ than CP$]$ $]$ $]$
   semantics: $[\![\text{more/-er}]\!]([\![\text{A}]\!])(\text{inr}([\![\text{CP}]\!]))(\text{inl}([\![\text{NumP}]\!]))$

d. Clausal comparative, no difference degree:
   syntax: $[_{\text{DegP}}$ $[_{\text{Deg}'}$ __ A $]$ $[_{\text{PP}}$ than CP$]$ $]$
   semantics: $[\![\text{more/-er}]\!]([\![\text{A}]\!])(\text{inr}([\![\text{CP}]\!])(\text{inr}(*))$

e. Measure phrase comparative, with difference degree:
   syntax: $[_{\text{DegP}}$ NumP$_2$ $[_{\text{Deg}'}$ $[_{\text{Deg}'}$ __ A $]$ $[_{\text{PP}}$ than NumP$_1]$ $]$ $]$
   semantics: $[\![\text{more/-er}]\!]([\![\text{A}]\!])(\text{inr}([\![\text{NumP}_1]\!]))(\text{inl}([\![\text{NumP}_2]\!]))$

f. Measure phrase comparative, no difference degree:
syntax: $[_{\text{DegP}}\ [_{\text{Deg}'}\ \_\_\ \text{A}\ ]\ [_{\text{PP}}\ \text{than NumP}]\ ]$
semantics: $[\![\text{more/-er}]\!]([\![\text{A}]\!])(\text{inr}([\![\text{NumP}]\!])(\text{inr}(*))$

Notice how the complement of *than* cannot be supplied to the $[\![\text{more/-er}]\!]$ function as it is, but must first be tagged with 'inl' or 'inr', depending on whether it denotes an element of Ind or $Degree(+)$. Similarly, if the difference degree argument is present, it must be tagged with 'inl' before it can be supplied as an argument; if no difference degree argument is present, then the value 'inr($*$)' must be supplied instead. Rather than develop a detailed theory of how this is done, I shall simply assume that the grammar contains some way to handle these different choices, by embedding arguments into sum types if required, and supplying the appropriate null element if no argument is available. For an approach to the syntax-semantics interface capable of this kind of flexible composition, see Asudeh et al. (2012).

We can now turn to formalising the comparative morphemes *more/-er* and *less*. These morphemes should take a gradable adjective, a *than* argument, an optional difference degree, and an individual, and return a context update. Hence, they are elements of the following type:

(349) $[\![\text{more/-er}]\!], [\![\text{less}]\!] : \mathit{Gradable} \to (\mathit{Ind} + \mathit{Degree}(+)) \to \mathit{Degree}(+)? \to \mathit{Ind} \to \mathit{Update}$

A potential implementation of $[\![\text{more/-er}]\!]$ is shown in Figure 4.2. We can understand the resulting update by breaking it into parts. Given some gradable adjective G, *than* argument T, optional difference degree d?, and individual x, the presuppositions are as follows:

(350)
1. x satisfies the background presuppositions of G
2. there is some degree d_{ref} (the reference degree)
3. d_{ref} measures the degree to which x is G
4. there is some degree d_{comp} (the comparison degree)
5. if T contains some individual y, then y satisfies the background presuppositions of G and d_{comp} is the degree to which y is G; otherwise, if T contains some degree d, then d_{comp} is given by taking d and adjusting its polarity to be the same as the reference degree (making it possible to compare the two)

Given a context satisfying all of these presuppositions, the assertion is that the magnitude of the reference degree is greater than the magnitude of the comparison degree, and the separation between the two degrees is given by the value of the difference degree argument d? (if present). To express the second part of the assertion, I assume that for every scale s and polarity p, there is a function:

(351) $\text{sep}_{(\text{s},\text{p})} : \mathit{Degree}(\text{s}, \text{p}) \times \mathit{Degree}(\text{s}, \text{p}) \to \mathit{Degree}(\text{s}, +)$

$$
\begin{array}{l}
[\![\text{more/-er}]\!] \coloneqq \\
\quad \lambda \mathrm{G} : \mathit{Gradable}\ . \\
\qquad \lambda \mathrm{T} : (\mathit{Ind} + \mathit{Degree}(+))\ . \\
\qquad\quad \lambda \mathrm{d?} : \mathit{Degree}(+)?\ . \\
\qquad\qquad \lambda \mathrm{x} : \mathit{Ind}\ . \\
\qquad\qquad \left(\begin{array}{l}
\mathrm{P} \coloneqq \left[\begin{array}{l}
\mathrm{c} : \mathrm{G(x).1} \\
\mathrm{d_{comp}} : \mathit{Degree} \\
\mathrm{d_{comp}} = \mathrm{G(x).2(c)} \\
\mathrm{d_{ref}} : \mathit{Degree} \\
\text{case T of} \\
\quad \text{inl(y) then} \left[\begin{array}{l} \mathrm{c'} : \mathrm{G(y).1} \\ \mathrm{d_{comp}} = \mathrm{G(y).2(c')} \end{array}\right] \\
\quad \mid \text{inr(d) then case } \mathrm{d_{ref}}.1 \text{ of} \\
\qquad + \text{ then } \mathrm{d_{comp}} = \begin{pmatrix} + \\ \mathrm{d.1} \\ \mathrm{d.2} \end{pmatrix} \\
\qquad \mid - \text{ then } \mathrm{d_{comp}} = \begin{pmatrix} - \\ \mathrm{d.1} \\ \mathrm{negate(d.2)} \end{pmatrix}
\end{array}\right] \\
\lambda \mathrm{p} : \mathrm{P}\ .\ \mathrm{p.2.3} > \mathrm{p.4.3} \times \\
\qquad\qquad \text{case d? of} \\
\qquad\qquad\quad \text{inl}(\mathrm{d_{diff}}) \text{ then sep(p.2.3, p.4.3)} = \mathrm{d_{diff}}.2 \\
\qquad\qquad\quad \mid \text{inr}(*) \text{ then } \top
\end{array}\right)
\end{array}
$$

Figure 4.2

which takes two degrees on the (s, p) scale and returns a degree on the $(s, +)$ scale representing the distance between them. The separation between two degrees has the same dimensions as the original degrees, but its polarity is always positive regardless of whether the original degrees are positive or negative. As one would expect, the morpheme $[\![\text{less}]\!]$ has exactly the same semantic value as $[\![\text{more}]\!]$, except that the $>$ relation in the assertion is replaced by $<$.

To understand how the comparative morpheme acts on senses, it is useful to see an example. Consider the expression *1km shorter than the Nile*. This is interpreted as follows, assuming that *short* is interpreted as $[\![\text{short}]\!]_{\text{path}}$:

(352) $[\![\text{1km shorter than the Nile}]\!]_{\text{path}}$

$$= [\![\text{more/-er}]\!](\text{as\_neg}'([\![\text{short}]\!]_{\text{path}}))(\text{inl}([\![\text{the Nile}]\!]))(\text{inl}([\![\text{1km}]\!]))$$

$$
= \lambda \mathrm{x} : \mathit{Ind}\ .\ \left(\begin{array}{l}
\mathrm{P} \coloneqq \left[\begin{array}{l}
\mathrm{c} : \left[\begin{array}{l} \mathrm{p} : \mathit{Path} \\ \textsc{path}(\mathrm{x}, \mathrm{p}) \end{array}\right] \\
\mathrm{d}_{\mathrm{ref}} : \mathit{Degree} \\
\mathrm{d}_{\mathrm{ref}} = \left(\begin{array}{l} - \\ \mathrm{dist} \\ \|\mathrm{c}.1\| \end{array}\right) \\
\mathrm{d}_{\mathrm{comp}} : \mathit{Degree} \\
\left[\begin{array}{l}
\mathrm{c}' : \left[\begin{array}{l} \mathrm{p} : \mathit{Path} \\ \textsc{path}(\mathit{Nile}, \mathrm{p}) \end{array}\right] \\
\mathrm{d}_{\mathrm{comp}} = \left(\begin{array}{l} - \\ \mathrm{dist} \\ \|\mathrm{c}'.1\| \end{array}\right)
\end{array}\right]
\end{array}\right] \\
\lambda \mathrm{p} : \mathrm{P}\ .\ \mathrm{p}.2.3 > \mathrm{p}.4.3\ \times \\
\qquad\qquad \mathrm{sep}(\mathrm{p}.2.3, \mathrm{p}.4.3) = 1\mathrm{km}
\end{array}\right)
$$

As indicated, the result takes an individual x and returns a context update with six presuppositions:

(353)
1. x satisfies satisfies the presuppositions of $[\![\text{short}]\!]_{\text{path}}$, having a pathlike shape
2. there is some degree $\mathrm{d}_{\mathrm{ref}}$ (the reference degree)
3. $\mathrm{d}_{\mathrm{ref}}$ measures the shortness (negative length) of x
4. there is some degree $\mathrm{d}_{\mathrm{comp}}$ (the comparison degree)
5. the Nile satisfies the presuppositions of $[\![\text{short}]\!]_{\text{path}}$
6. $\mathrm{d}_{\mathrm{comp}}$ measures the shortness of the Nile

Given a context satisfying these conditions, the first part of the assertion is that the shortness of x is greater than the shortness of the Nile. The second part of the assertion is that the separation between the two shortnesses is 1km.

Now instead of the nominal comparative *1km shorter than <u>the Nile</u>*, consider the measure phrase comparative *1km shorter than <u>15km</u>*. This is interpreted as follows, again assuming that *short* has the interpretation $[\![\text{short}]\!]_{\text{path}}$:

(354) $[\![\text{1km shorter than 15km}]\!]_{\text{path}}$

$$= [\![\text{more/-er}]\!](\text{as\_neg}'([\![\text{short}]\!]_{\text{path}}))(\text{inl}([\![\text{the Nile}]\!]))(\text{inl}([\![\text{1km}]\!]))$$

$$= \lambda x : \mathit{Ind}\ .\ \left(\begin{array}{l} \mathrm{P} \coloneqq \left[\begin{array}{l} \mathrm{c} : \left[\begin{array}{l} \mathrm{p} : \mathit{Path} \\ \textsc{path}(\mathrm{x}, \mathrm{p}) \end{array} \right] \\ \mathrm{d_{ref}} : \mathit{Degree} \\ \mathrm{d_{ref}} = \begin{pmatrix} - \\ \mathrm{dist} \\ \|\mathrm{c}.1\| \end{pmatrix} \\ \mathrm{d_{comp}} : \mathit{Degree} \\ \mathrm{d_{comp}} = \begin{pmatrix} - \\ \mathrm{dist} \\ 15\mathrm{km} \end{pmatrix} \end{array} \right] \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \mathrm{p}.2.3 > \mathrm{p}.4.3 \times \\ \qquad\qquad \mathrm{sep}(\mathrm{p}.2.3, \mathrm{p}.4.3) = 1\mathrm{km} \end{array} \right)$$

Now, instead of the comparison degree $\mathrm{d_{comp}}$ measuring the shortness of the Nile, it is simply set equal to 15cm. Note that the polarity of the comparison degree is required to be negative, so that it matches the polarity of the reference degree. As before, the assertion is that the magnitude of the reference degree is greater than the comparison degree and that the degree of separation between them is 1km.

There are various ways in which a comparative sentence can ‘go wrong’. For instance, consider the following sentences:

(355) a. ? John is taller (/ less tall) than 150kg.
b. ? John is taller (/ less tall) than Mary is intelligent.
c. ? John is 10°C taller (/ shorter / less tall / less short) than Mary.
d. ? John is 10cm taller (/ shorter / less tall / less short) than Mary is intelligent.

Each of these violates the requirement that the reference degree, comparison degree and difference degree must all agree in dimensions, with the result that the comparative assertion is not well-formed. The comparative will also be infelicitous if either the subject or the complement of *than* fails to satisfy the background requirements of the adjective, as in:

(356) a. ? John is taller than the meeting.
b. ? The idea was wider than the courtyard.

Both cases lead to a contradictory presupposition type, since the background requirements of the adjective cannot be satisfied. Finally, a comparative sentence can also fail if the complement of *than* is a negative degree. Consider the following sentence:

(357) ? John is taller than Mary is short.

This leads to a type error because the clausal complement of *than* denotes a negative degree, whereas $[\![\text{more/-er}]\!]$ requires a positive degree.

4.3.3 Action on arrows

As with measure phrase combination, the comparative acts not only on senses, but also on arrows, which it lifts monotonically. Given two gradable adjectives $G, H : \mathit{Gradable}$ with an arrow:

(358) $\alpha : \mathrm{G} \rightarrow_{\mathit{Gradable}} \mathrm{H}$

we can lift this to an arrow:

(359) $[\![\text{more/-er}]\!](\alpha) :$

$$\prod_{\mathrm{T}:(\mathit{Ind}+\mathit{Degree}(+))} \prod_{\mathrm{d?}:\mathit{Degree}(+)} \prod_{\mathrm{x}:\mathit{Ind}} [\![\text{more/-er}]\!](\mathrm{G})(\mathrm{T})(\mathrm{d?})(\mathrm{x}) \rightarrow_{\mathit{Update}} [\![\text{more/-er}]\!](\mathrm{H})(\mathrm{T})(\mathrm{d?})(\mathrm{x})$$

which is implemented as follows:

(360) $[\![\text{more/-er}]\!](\alpha) :=$

$$\begin{array}{l} \lambda \mathrm{T} : (\mathit{Ind} + \mathit{Degree}(+)) \ . \\ \quad \lambda \mathrm{d?} : \mathit{Degree}(+)? \ . \\ \quad\quad \lambda \mathrm{x} : \mathit{Ind} \ . \\ \quad\quad\quad \text{let } \mathrm{G}' := [\![\text{more/-er}]\!](\mathrm{G})(\mathrm{T})(\mathrm{d?})(\mathrm{x}) \text{ in} \\ \quad\quad\quad \left(\begin{array}{l} \lambda \mathrm{p} : \mathrm{G}'.1 \ . \begin{bmatrix} \alpha(\mathrm{x}).1(\mathrm{p}.1) \\ \mathrm{p}.2 \\ \mathrm{p}.3 \\ \mathrm{p}.4 \\ \text{case T of inl(y) then} \begin{pmatrix} \alpha(\mathrm{y}).1(\mathrm{p}.5.1) \\ \mathrm{p}.5.2 \end{pmatrix} \\ \qquad\qquad \mid \text{inr}(\mathrm{d}') \text{ then p.5} \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{G}'.1 \ . \ \mathrm{id}_{\mathrm{G}'.2(\mathrm{p})} \end{array} \right) \end{array}$$

As before, the lifted arrow works by using the original arrow to replace all the presuppositions of G in the background context with presuppositions of H. The only components which need to be altered in this way are the first component, and the fifth component in the case where T contains an individual, since these alone contains presuppositions associated with G. There is no need to alter the assertion, so the assertion map is just the identity. The other comparative morpheme *less* lifts arrows in exactly the same way, since its presuppositions are the same as those of *more/-er*. Knowing how it acts on both senses and arrows, we can now apply *more/-er* to an entire network of gradable adjective senses.

4.4 The superlative

4.4.1 Background

The English superlative refers to the morphemes *most/-est*[1] and *least*, as in *John is the tallest*, *Mary is the shortest*, *this car is the least expensive*, and so on. The superlative morphemes assert that the degree to which the individual possesses the relevant property is maximal or minimal with respect to some *comparison class*. For example, the sentence *John is the tallest* presupposes a set of individuals against whom John's height is evaluated. The truth of a comparative sentence depends on the comparison class – John may be *the tallest* in one context but not in another. Contrast this with measure phrase combination or the comparative, e.g. *John is 150cm tall, John is taller than Mary*, which are true or false regardless of context.

Superlative sentences are associated with a well known ambiguity, noticed by Szabolcsi (1986). Consider the sentence:

(361) John climbed the highest mountain.

This is ambiguous between an *absolute* reading in which John climbed the highest of all mountains, and a *comparative* reading in which John climbed a higher mountain than anyone else (in some contextually relevant set of individuals) climbed. The absolute reading is only true iff John climbed Mount Everest, assuming this is the highest mountain; the comparative reading requires only that no other person in the context climbed a higher mountain.

Within degree semantics, the most influential analysis of the absolute/comparative distinction is due to Heim (1995). She assumes a relational analysis of gradable adjectives whereby they denote relations between individuals and degrees of type $d \to e \to t$. They are downwards monotone in the sense that ⟦high⟧(x)(d) is true iff x's degree of height is *at least* d. Modified nominal expressions like *high mountain* are also assumed to be of type $d \to e \to t$. So, for instance, ⟦high mountain⟧(x)(d) is true iff x is a mountain whose height is at least d. Given these assumptions, she proposes the following analysis of the superlative morpheme:

(362)
$$[\![\text{most/-est}]\!] \coloneqq \lambda C\,.\;\lambda g\,.\;\lambda x \mid x \in C\,.\;\exists d\,[\,g(d)(x)\;\wedge\;\forall y\,[\,y \neq x \wedge y \in C \Rightarrow \neg g(d)(y)\,]\,]$$

where C is a set representing the comparison class, g is a gradable adjective meaning, and x is an individual which must belong to the comparison class. On this account, ⟦most/-est⟧(C)(g)(x) holds iff there exists some degree d such that g(d)(x), and there is no other individual y in the comparison class such that g(d)(y).

Heim proceeds to explain the absolute/comparative distinction as follows. In an absolute superlative, the superlative morpheme is interpreted within its host DP, resulting in the following logical form:

[1]As with *more* and *-er*, the distinction between *most* and *-est* is morphophonemic.

(363) John climbed [the [C-est [high mountain]]
'John climbed the unique mountain x such that x is a d-high mountain and no other element of C is a d-high mountain'

Notice *-est* must raise so as to take [high mountain] as its gradable adjective argument. C is a phonologically null pronominal variable containing some relevant set of mountains. The comparative superlative, on the other hand, Heim assigns the following logical form:

(364) John [C-est [1 [climbed a ~~the~~ d_1-high mountain]]]
'There is a degree d such that John climbed a d-high mountain and no other member of C climbed a d-high mountain'

This interpretation is derived by (i) moving *C-est* to a position above *climb*, (ii) abstracting over the degree argument of *high mountain*, obtaining a function $d \to e \to t$ which takes a degree and returns a set of individuals who climbed a d-high mountain, and (iii) replacing the definite determiner with an indefinite determiner. In this reading, C is a set of mountain climbers rather than a set of mountains.

Heim's description of the absolute/comparative distinction as a kind of scope ambiguity has been criticised by a number of authors. One problem, pointed out by Sharvit & Stateva (2002), has to do with the comparative interpretation of the other superlative morpheme, *least*. By direct analogy with *most/-est*, Heim ought to assign *least* the following interpretation:

(365) $$\begin{aligned}&[\![\text{least}]\!] \coloneqq \lambda C\,.\\ &\qquad \lambda g\,.\\ &\qquad\quad \lambda x \mid x \in C\,.\\ &\qquad\qquad \exists d\,[\,\neg g(d)(x) \;\wedge\; \forall y\,[\,y \neq x \wedge y \in C \Rightarrow g(d)(y)\,]\,]\end{aligned}$$

That is, [[least]](C)(g)(x) is true iff there is some degree d such that ¬g(d)(x), but for any other individual y in the comparison class we do have g(d)(y). Like *most/-est*, *least* would give rise to both an absolute and a comparative interpretation, as follows:

(366) a. Mary climbed [the [C-least [high mountain]]
'Mary climbed the unique mountain x such that x is not a d-high mountain but every other element of C is a d-high mountain'
b. Mary [C-least [1 [climbed a ~~the~~ d_1-high mountain]]]
'Mary (who climbed a mountain) did not climb a d-high mountain, but every other member of C did climb a d-high mountain'

Sharvit & Stateva point out that interpretation (b) gives the wrong truth conditions in situations where some individuals climbed more than one mountain. They give the following as an example:

(367)
- $C \coloneqq \{Mary, Bill, John\}$
- Mary climbed one mountain – a 3000ft mountain.

- Bill climbed two mountains – a 2500ft mountain and a 3500ft mountain.
- John climbed one mountain – a 4000ft mountain.

Intuitively, Bill climbed the lowest mountain in this situation, because he climbed a mountain which was only 2500ft high. However, under the analysis of *least* given above, 'Mary climbed the least high mountain' is true, because we can find a degree d such that Mary did not climb a d-high mountain but everyone else did.

To resolve this problem, Sharvit & Stateva propose doing away with the movement analysis of the absolute/comparative distinction. Instead, the comparative morpheme is always interpreted in the lower position, inside its host DP. The difference between the absolute and comparative readings is due to two different 'strategies' for determining the comparison class C, as follows:

(368) Mary climbed [the [C-least [high mountain]]
'Mary climbed the unique mountain x such that x is not a d-high mountain and every other element of C is a d-high mountain'
  a. <u>Absolute interpretation:</u> C is the set of all relevant mountains.
  b. <u>Comparative interpretation:</u> C is the set of all mountains climbed by relevant individuals.

This approach gives the correct truth conditions in the case of (367) because in both readings the comparison class C is a set of mountains, rather than a set of mountains in the absolute interpretation and a set of mountain climbers in the comparative interpretation.

Farkas & Kiss (2000) propose an alternative description of the superlative morpheme. Like Sharvit & Stateva, they assume that the superlative is always interpreted inside its host DP, the difference between the absolute and comparative readings being a matter of interpretation rather than scope ambiguity. However, unlike Sharvit & Stateva, they adopt the measure function approach to gradable adjectives pioneered by Kennedy. In line with Kennedy's theory that gradable adjectives project a DegP, they propose the syntax in Figure 4.3. Notice that there is no longer any need for *most/-est* to raise prior to interpretation, so as to take [high mountain] as an argument; instead, all elements are interpreted in their surface positions. The superlative morpheme *most/-est* is then assigned the following denotation:

(369) $[\![\text{most/-est}]\!] \coloneqq \lambda g \, . \, \lambda \mathrm{P} \, . \, \lambda x \, . \, P(x) \wedge \forall y \, [P(y) \wedge y \neq x \Rightarrow g(x) > g(y)\,]$

That is, $[\![\text{most/-est}]\!]$(g)(P)(x) holds iff x is P and the degree to which x is g is greater than the degree to which any other P is g. Farkas & Kiss propose that, in the absolute interpretation, P is identical to the denotation of the NP argument (in this case *mountain*), whereas in the comparative interpretation it is implicitly restricted to some subset of the NP denotation (e.g. all mountains climbed by some relevant set of individuals). Notice that, unlike the description of the superlative morpheme in Heim and Sharvit & Stateva, Farkas & Kiss do not include the comparison class C, which handles the restriction of P to some relevant subset. They argue that this is simply a matter of ordinary quantifier domain restriction, the same

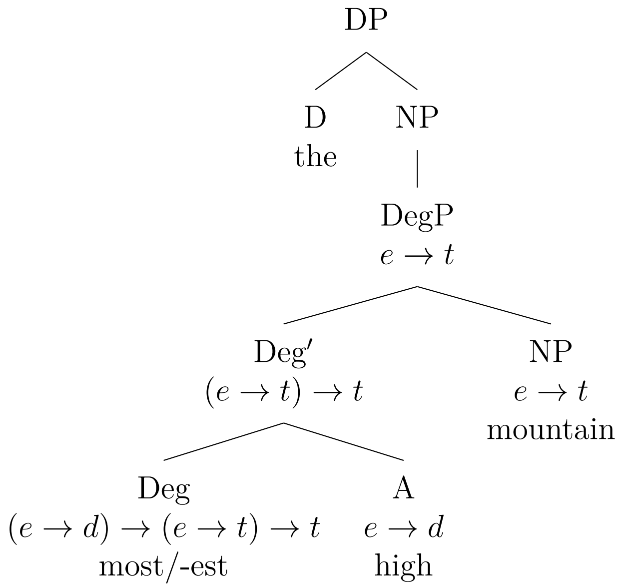


Figure 4.3: The syntax of the superlative, according to Farkas & Kiss (2000).

mechanism involved in interpreting *every mountain* or *the mountains* relative to some context. Assuming quantifier domain restriction is a matter of pragmatics, the semantic form need make no reference to it.

4.4.2 Action on senses

My analysis most closely resembles that of Farkas & Kiss. Like them, I assume that gradable adjectives denote measure functions, and that a superlative DP contains a DegP layer, as shown in (4.3). I agree with both Farkas & Kiss and Sharvit & Stateva, contra Heim, that superlatives are always interpreted in their base position and are not subject to raising. The difference between the absolutive and comparative reading is not explained structurally but involves two different interpretations of the comparison class. However, I agree with Heim and Sharvit & Stateva, contra Farkas & Kiss, that the comparison class should be represented as a logical argument to the superlative morpheme, rather than a pragmatically determined domain restriction. The reason for this is that certain superlative expressions completely specify the comparison class via an accompanying PP. Examples include:

(370) a. the highest mountain on Earth
b. the youngest of Mary, Bill and John
c. the tallest man in the UK
d. the tallest among the people in the bar

In cases like this, the comparison class is provided compositionally by the PP and the resulting expression has only an absolute interpretation. The comparative interpretation is no longer available, presumably because it requires the comparison class to be left undetermined by the semantics.

I therefore treat the superlative morpheme as dependent on both an optional NP argument and an optional PP argument. If both arguments are present, as in *John climbed the highest mountain on Earth*, then the PP acts as a restriction on the domain given by the NP, and the sentence has only an absolute interpretation. If the NP argument is present without the PP argument, as in *John climbed the highest mountain*, then the domain of the NP is pragmatically restricted, allowing for either an absolutive or a comparative reading. If the PP argument is present without the NP argument, as in *John climbed the highest on Earth*, then the context supplies an implicit domain (e.g. mountain) which is then restricted by the explicit PP, allowing only an absolute interpretation. Finally, if neither the NP nor the PP argument is present, as in *John climbed the highest*, then the context supplies both the implicit domain and an implicit domain restriction, allowing for either the absolutive or the comparative reading.

The superlative morphemes *most/-est* and *least* take a gradable adjective, an optional NP meaning, an optional PP meaning, and an individual, and return a context update. They are typed as follows:

(371) $$[\![\text{most/-est}]\!], [\![\text{least}]\!] : \mathit{Gradable} \rightarrow (\mathit{Ind} \rightarrow \mathit{Type})? \rightarrow (\mathit{Ind} \rightarrow \mathit{Update})? \rightarrow \mathit{Ind} \rightarrow \mathit{Update}$$

Note that an optional NP meaning is of type $(\mathit{Ind} \rightarrow \mathit{Type})?$, whereas an optional PP meaning is of type $(\mathit{Ind} \rightarrow \mathit{Update})?$ because it carries accompanying presuppositions. An implementation of $[\![\text{most/-est}]\!]$ is shown in Figure 4.4. Given a gradable adjective G, an optional NP meaning N?, an optional PP meaning P?, and an individual x, we get the following presuppositions:

(372)
1. there is some noun meaning D, representing the domain of the superlative
2. if the N? argument contains a non-empty value N, then D = N
3. there is some R, a predicate on elements of D, representing the domain restriction
4. if the P? argument contains a non-empty value P, then (i) elements of D must satisfy the presuppositions of P, and (ii) R is the predicate which picks out all elements of D which satisfy the assertion of P
5. x satisfies D
6. x satisfies R
7. x satisfies the presuppositions of G
8. there exists at least one other individual distinct from x which satisfies D, R and the presuppositions of G

Given a context satisfying these presuppositions, the assertion is that, for every individual y distinct from x which satisfies D, R and the presuppositions of G, the degree to which x is G is greater than the degree to which y is G. The morpheme *least* has the same semantic value as *most*, except for the assertion, where $>$ is replaced by $<$, ensuring that the degree to which x is G is minimal rather than maximal.

$$
\begin{array}{l}
[\![\text{most/-est}]\!] := \\
\quad \lambda \mathrm{G} : \mathit{Gradable}\ . \\
\quad\quad \lambda \mathrm{N?} : (\mathit{Ind} \to \mathit{Type})?\ . \\
\quad\quad\quad \lambda \mathrm{P?} : (\mathit{Ind} \to \mathit{Update})?\ . \\
\quad\quad\quad\quad \lambda \mathrm{x} : \mathit{Ind}\ . \\
\left(\begin{array}{l}
\mathrm{P}' := \left[\begin{array}{l}
\mathrm{D} : \mathit{Ind} \to \mathit{Type} \\
\text{case N? of} \\
\quad \text{inl(N) then D} = \mathrm{N} \\
\quad \text{inr}(*) \text{ then } \top \\
\mathrm{R} : \prod_{\mathrm{y}:\mathit{Ind}} \mathrm{D(y)} \to \mathit{Prop} \\
\text{case P? of} \\
\quad \text{inl(P) then} \left[\begin{array}{l}
\mathrm{c} : \prod_{\mathrm{y}:\mathit{Ind}} \mathrm{D(y)} \to \mathrm{P(y).1} \\
\mathrm{R} = \lambda \mathrm{y} : \mathit{Ind}\ . \\
\qquad\qquad \lambda \mathrm{D_y} : \mathrm{D(y)}\ . \\
\qquad\qquad\quad \mathrm{P(y).2(c(y)(D_y))}
\end{array}\right] \\
\quad \mid \text{inr}(*) \text{ then } \top \\
\mathrm{D_x} : \mathrm{D(x)} \\
\mathrm{R(x)(D_x)} \\
\mathrm{G(x).1} \\
\left\|\left[\begin{array}{l}
\mathrm{y} : \mathit{Ind} \\
\mathrm{D_y} : \mathrm{D(y)} \\
\mathrm{R(y)(D_y)} \\
\mathrm{G(y).1} \\
\mathrm{y} \neq \mathrm{x}
\end{array}\right]\right\|
\end{array}\right] \\
\lambda \mathrm{p} : \mathrm{P}'\ . \prod \left(\mathrm{r} : \left[\begin{array}{l}
\mathrm{y} : \mathit{Ind} \\
\mathrm{D_y} : \mathrm{p.1(y)} \\
\mathrm{p.3(y)(D_y)} \\
\mathrm{G(y).1} \\
\mathrm{y} \neq \mathrm{x}
\end{array}\right] \right) \mathrm{G(x).2(p.7).3} > \mathrm{G(r.1).2(r.4).3}
\end{array}\right)
\end{array}
$$

Figure 4.4

To illustrate the action of the superlative on senses, consider the expression *highest mountain on Earth*. This is interpreted as follows, assuming the sense $[\![\text{high}]\!]_{\text{dim}}$ for *high*:

(373) $[\![\text{highest mountain on Earth}]\!]_{\text{dim}}$

$$= [\![\text{most/-est}]\!](\text{as\_pos}'([\![\text{high}]\!]_{\text{dim}}))(\text{inl}([\![\text{mountain}]\!]))(\text{inl}([\![\text{on Earth}]\!]))$$

$$
= \lambda \mathrm{x} : \mathit{Ind}\ .
$$

$$
\left(\begin{array}{l}
\mathrm{P} := \left[\begin{array}{l}
\mathrm{D} : \mathit{Ind} \rightarrow \mathit{Type} \\
\mathrm{D} = \llbracket \mathrm{mountain} \rrbracket \\
\mathrm{R} : \prod_{\mathrm{y}:\mathit{Ind}} \mathrm{D}(\mathrm{y}) \rightarrow \mathit{Prop} \\
\left[\begin{array}{l}
\mathrm{c} : \prod_{\mathrm{y}:\mathit{Ind}} \mathrm{D}(\mathrm{y}) \rightarrow \llbracket \mathrm{on\ Earth} \rrbracket(\mathrm{y}).1 \\
\mathrm{R} = \lambda \mathrm{y} : \mathit{Ind}\ .\ \lambda \mathrm{D_y} : \mathrm{D}(\mathrm{y})\ .\ \llbracket \mathrm{on\ Earth} \rrbracket(\mathrm{y}).2(\mathrm{c}(\mathrm{y})(\mathrm{D_y}))
\end{array}\right] \\
\mathrm{D_x} : \mathrm{D}(\mathrm{x}) \\
\mathrm{R}(\mathrm{x})(\mathrm{D_x}) \\
\llbracket \mathrm{high} \rrbracket_{\mathrm{dim}}(\mathrm{x}).1 \\
\left\|\left[\begin{array}{l}
\mathrm{y} : \mathit{Ind} \\
\mathrm{D_y} : \mathrm{D}(\mathrm{y}) \\
\mathrm{R}(\mathrm{y})(\mathrm{D_y}) \\
\llbracket \mathrm{high} \rrbracket_{\mathrm{dim}}(\mathrm{y}).1 \\
\mathrm{y} \neq \mathrm{x}
\end{array}\right]\right\|
\end{array}\right] \\
\lambda \mathrm{p} : \mathrm{P}\ .\ \prod \left(\mathrm{r} : \left[\begin{array}{l}
\mathrm{y} : \mathit{Ind} \\
\mathrm{D_y} : \mathrm{p}.1(\mathrm{y}) \\
\mathrm{p}.3(\mathrm{y})(\mathrm{D_y}) \\
\llbracket \mathrm{high} \rrbracket_{\mathrm{dim}}(\mathrm{y}).1 \\
\mathrm{y} \neq \mathrm{x}
\end{array}\right] \right)
\begin{array}{l}
\llbracket \mathrm{high} \rrbracket_{\mathrm{dim}}(\mathrm{x}).2(\mathrm{p}.7).3 \\
> \llbracket \mathrm{high} \rrbracket_{\mathrm{dim}}(\mathrm{r}.1).2(\mathrm{r}.4).3
\end{array}
\end{array}\right)
$$

The presuppositions are as follows:

(374)
1. there is some predicate D, the domain of the superlative
2. D is equal to ⟦mountain⟧
3. there is some R, a restriction on the set of mountains
4. mountains satisfy the background conditions for being on Earth (e.g. having a spatial location), and R is the property of being on Earth
5. x is a mountain
6. x is on Earth
7. x satisfies the presuppositions for dimensional height (having an upwards-directed axis)
8. there is at least one other mountain on Earth apart from x which satisfies the presuppositions for dimensional height

The assertion is that for every other mountain on Earth y, the magnitude of y's vertical axis is less than the magnitude of x's vertical axis.

When a PP argument is present, as in *highest mountain <u>on Earth</u>*, the domain restriction is fixed by the semantics and only an absolute interpretation is possible. Now consider the expression *highest mountain* without an accompanying PP argument:

(375) $[\![\text{highest mountain}]\!]_{\text{dim}}$

$$= [\![\text{most/-est}]\!](\text{as\_pos}'([\![\text{high}]\!]_{\text{dim}}))(\text{inl}([\![\text{mountain}]\!]))(\text{inr}(*))$$

$$= \lambda \text{x} : \mathit{Ind}\ .$$

$$\left(\begin{array}{l} \text{P} \coloneqq \left[\begin{array}{l} \text{D} : \mathit{Ind} \to \mathit{Type} \\ \text{D} = [\![\text{mountain}]\!] \\ \text{R} : \prod_{\text{y}:\mathit{Ind}} \text{D}(\text{y}) \to \mathit{Prop} \\ \top \\ \text{D}_{\text{x}} : \text{D}(\text{x}) \\ \text{R}(\text{x})(\text{D}_{\text{x}}) \\ [\![\text{high}]\!]_{\text{dim}}(\text{x}).1 \\ \left\| \left[\begin{array}{l} \text{y} : \mathit{Ind} \\ \text{D}_{\text{y}} : \text{D}(\text{y}) \\ \text{R}(\text{y})(\text{D}_{\text{y}}) \\ [\![\text{high}]\!]_{\text{dim}}(\text{y}).1 \\ \text{y} \neq \text{x} \end{array} \right] \right\| \end{array} \right] \\ \lambda \text{p} : \text{P}\ .\ \prod \left(\text{r} : \left[\begin{array}{l} \text{y} : \mathit{Ind} \\ \text{D}_{\text{y}} : \text{p}.1(\text{y}) \\ \text{p}.3(\text{y})(\text{D}_{\text{y}}) \\ [\![\text{high}]\!]_{\text{dim}}(\text{y}).1 \\ \text{y} \neq \text{x} \end{array} \right] \right) \begin{array}{l} [\![\text{high}]\!]_{\text{dim}}(\text{x}).2(\text{p}.7).3 \\ > [\![\text{high}]\!]_{\text{dim}}(\text{r}.1).2(\text{r}.4).3 \end{array} \end{array} \right)$$

The resulting update is identical to (373), except that the domain restriction R is left unspecified. The domain is no longer specifically restricted to the set of mountains on Earth; rather, the relevant restriction must be supplied by the discourse context. This is what allows for the comparative reading of *John climbed the highest mountain* in which John climbed the highest mountain in some salient set rather than the highest of all mountains.

As before, the different components of the presupposition type are responsible for different ways that the superlative can be infelicitous. One possibility is that the NP and PP meanings are incompatible, as in:

(376) ? the highest mountain of all the rivers in South America

This is infelicitous because mountains are not the kinds of things which can be rivers in South America. As a result, it is impossible for the subject (or any other individual) to satisfy both the NP meaning and the PP meaning, and the presupposition type is uninhabited. Another way that the superlative can be infelicitous is if the subject fails to satisfy either (a) the domain predicate, (b) the domain restriction, or (c) the presuppositions of the adjective. Examples are given below:

(377) a. ? Mount Everest is the longest river.

(Mount Everest is not a river.)

b. ? Of all the rivers in North America, the Danube is the longest.
(The Danube is not a river in North America.)

c. ? The Danube is the longest$_{\text{axis}}$ river
(The Danube does not have a straight line length.)

The superlative will also be infelicitous if there is no individual besides the subject which satisfies the domain predicate, the domain restriction, and the adjective presuppositions. For example, suppose that a coffee table contains several objects, only one of which has a depth. Then the expression:

(378) ? the deepest object on the coffee table

is infelicitous, since it presupposes that at least one other object on the table has a depth which can be compared with that of the subject. In this case, it is the final component of the presupposition type which is uninhabited. Finally, the superlative is also associated with a uniqueness presupposition. Suppose that the only objects on the coffee table are two mugs with exactly the same depth. Then (378) will be infelicitous for a different reason: because there is no unique deepest object on the table. I take this presupposition to be a contribution of the definite article *the*, rather than a feature of the superlative morpheme itself. See Section 2.6.3 for a formulation of the definite article in which the uniqueness presupposition is made explicit.

4.4.3 Action on arrows

The action of the superlative on gradable adjective arrows is different to that of the comparative or the MEAS morpheme. To illustrate, consider the sentence:

(379) The Christmas tree is the highest object in the room.

Let us suppose that this is true under the interpretation $[\![\text{high}]\!]_{\text{dim, grnd}}$ – that is, where height refers to the vertical dimension of an object based on the ground. It does not follow that the Christmas tree is the highest object under some more general notion of height, since when we broaden the concept there may be objects which are higher than the Christmas tree. For example, if we adopt the more general notion of elevation from the ground, represented by $[\![\text{high}]\!]_{\text{elev}}$, a light on the ceiling could be higher than the Christmas tree. It follows that the superlative morpheme does not lift gradable adjective arrows in the usual monotone fashion.

However, the superlative still acts on arrows. First notice that the presuppositions are mapped monotonically: if we know that the Christmas tree satisfies the presuppositions of the stronger sense $[\![\text{high}]\!]_{\text{dim, grnd}}$, then we know it must satisfy the presuppositions of the weaker sense $[\![\text{high}]\!]_{\text{elev}}$. The assertion, on the other hand, is mapped anti-monotonically: if we know that the Christmas tree is the highest object under the more general sense, then it must also be the highest object under the more specific sense. Recall from Section 2.6.5 that the property of lifting presuppositions monotonically but assertions anti-monotonically is associated with negation. The fact that the superlative follows the same pattern should come as no surprise: to say that something is *the highest object in the room* is to deny that there is any other object in the room which is higher than it.

To summarise, given two gradable adjectives $\mathrm{G}, \mathrm{H} : \mathit{Gradable}$ linked by an arrow:

(380) $\alpha : \mathrm{G} \rightarrow_{\mathit{Gradable}} \mathrm{H}$

this can be lifted to a skewed arrow:

(381)
$$[\![\text{most/-est}]\!](\alpha) : \prod_{\mathrm{N?}:(\mathit{Ind}\rightarrow\mathit{Type})?} \prod_{\mathrm{P?}:(\mathit{Ind}\rightarrow\mathit{Update})?} \prod_{\mathrm{x}:\mathit{Ind}} [\![\text{most/-est}]\!](\mathrm{G})(\mathrm{N?})(\mathrm{P?})(\mathrm{x}) \rightleftarrows_{\mathit{Update}} [\![\text{most/-est}]\!](\mathrm{H})(\mathrm{N?})(\mathrm{P?})(\mathrm{x})$$

which is implemented as follows (see Section 2.6.5 for the definition of $\leftrightarrows_{\mathrm{Update}}$):

(382)
$$\begin{array}{l}
[\![\text{most/-est}]\!](\alpha) \coloneqq \\
\quad \lambda \mathrm{N?} : (\mathit{Ind} \rightarrow \mathit{Type})? \,. \\
\quad\quad \lambda \mathrm{P?} : (\mathit{Ind} \rightarrow \mathit{Update})? \,. \\
\quad\quad\quad \lambda \mathrm{x} : \mathit{Ind} \,. \\
\quad\quad\quad\quad \text{let} \begin{Bmatrix} \mathrm{G}' \coloneqq [\![\text{most/-est}]\!](\mathrm{G})(\mathrm{N?})(\mathrm{P?})(\mathrm{x}) \\ \mathrm{H}' \coloneqq [\![\text{most/-est}]\!](\mathrm{H})(\mathrm{N?})(\mathrm{P?})(\mathrm{x}) \end{Bmatrix} \text{in} \\
\quad\quad\quad\quad \left[\begin{array}{l}
\mathrm{f} \coloneqq \lambda \mathrm{p} : \mathrm{G}'.1 \,. \begin{pmatrix} \mathrm{p}.1 \\ \mathrm{p}.2 \\ \mathrm{p}.3 \\ \mathrm{p}.4 \\ \mathrm{p}.5 \\ \mathrm{p}.6 \\ \alpha(\mathrm{x}).1(\mathrm{p}.7) \\ \left| \lambda \mathrm{r} : \begin{bmatrix} \mathrm{y} : \mathit{Ind} \\ \mathrm{D_y} : \mathrm{p}.1 \\ \mathrm{p}.3(\mathrm{y})(\mathrm{D_y}) \\ \mathrm{G}(\mathrm{y}).1 \\ \mathrm{y} \neq \mathrm{x} \end{bmatrix} . \left| \begin{pmatrix} \mathrm{r}.1 \\ \mathrm{r}.2 \\ \mathrm{r}.3 \\ \alpha(\mathrm{r}.1).2(\mathrm{r}.4) \\ \mathrm{r}.5 \end{pmatrix} \right| \right| (\mathrm{p}.8) \end{pmatrix} \\
\lambda \mathrm{p} : \mathrm{G}'.1 \,. \\
\quad \lambda \mathrm{q} : \mathrm{H}'.2(\mathrm{f}(\mathrm{p})) \,. \left(\lambda \mathrm{r} : \begin{bmatrix} \mathrm{y} : \mathit{Ind} \\ \mathrm{D_y} : \mathrm{p}.1(\mathrm{y}) \\ \mathrm{p}.3(\mathrm{y})(\mathrm{D_y}) \\ \mathrm{G}(\mathrm{y}).1 \\ \mathrm{y} \neq \mathrm{x} \end{bmatrix} . \mathrm{q}(\begin{pmatrix} \mathrm{r}.1 \\ \mathrm{r}.2 \\ \mathrm{r}.3 \\ \alpha(\mathrm{r}.1).2(\mathrm{r}.4) \\ \mathrm{r}.5 \end{pmatrix}) \right)
\end{array}\right]
\end{array}$$

As usual, the original map α is used to replace the presuppositions of G with presuppositions of H. The two components of the presupposition type which need to be altered in this way are the seventh and eighth components. Recall that the eighth component contains a proof that there exists some individual other than the subject which belongs to the domain and satisfies the adjective presuppositions. To alter this component, we use the fact that any function $f : A \to B$, where B is a proposition, can be lifted to a function $|f| : |A| \to B$. For the assertion, the idea is to take the proof that x is maximal with respect to the more general adjective H, and restrict attention to the more specific adjective G. This is done using the original maximality proof in combination with α, as shown. The other superlative morpheme, *least*, acts on arrows in an exactly analogous way.

We have established that $[\![\text{most/-est}]\!]$ is a skew-monotone function of gradable adjective networks. It also acts skew-monotonically on arrows in its NP and PP argument positions. For example, if you know that the M6 is the longest road in the UK and you also know that the M6 is a motorway (a special case of road), then you know that it is the longest motorway in the UK. In other words, given two noun meanings $\mathrm{N}_1, \mathrm{N}_2 : \mathit{Ind} \to \mathit{Type}$ connected by an arrow:

(383) $\alpha' : \prod_{\mathrm{x}:\mathit{Ind}} \mathrm{N}_1(\mathrm{x}) \to \mathrm{N}_2(\mathrm{x})$

this is lifted to an arrow:

(384) $[\![\text{most/-est}]\!](-)(\alpha') :$

$$\begin{aligned}&\textstyle\prod_{\mathrm{G}:\mathit{Gradable}}\\ &\quad\textstyle\prod_{\mathrm{P?}:(\mathit{Ind}\to\mathit{Update})?}\\ &\qquad\textstyle\prod_{\mathrm{x}:\mathit{Ind}}\\ &\qquad\quad [\![\text{most/-est}]\!](\mathrm{G})(\mathrm{inl}(\mathrm{N}_1))(\mathrm{P?})(\mathrm{x}) \rightleftarrows_{\mathit{Update}}\\ &\qquad\qquad [\![\text{most/-est}]\!](\mathrm{G})(\mathrm{inl}(\mathrm{N}_2))(\mathrm{P?})(\mathrm{x})\end{aligned}$$

Likewise, if you know that the Nile is the longest river in the world, and you also know that the Nile is in Africa, then you can deduce that the Nile is the longest river in Africa. That is, given two PP meanings $\mathrm{P}_1, \mathrm{P}_2 : \mathit{Ind} \to \mathit{Update}$ connected by an arrow:

(385) $\alpha'' : \prod_{\mathrm{x}:\mathit{Ind}} \mathrm{P}_1(\mathrm{x}) \to_{\mathit{Update}} \mathrm{P}_2(\mathrm{x})$

this is lifted as follows:

(386) $[\![\text{most/-est}]\!](-)(-)(\alpha'') :$

$$\begin{aligned}&\textstyle\prod_{\mathrm{G}:\mathit{Gradable}}\\ &\quad\textstyle\prod_{\mathrm{N?}:(\mathit{Ind}\to\mathit{Type})?}\\ &\qquad\textstyle\prod_{\mathrm{x}:\mathit{Ind}}\\ &\qquad\quad [\![\text{most/-est}]\!](\mathrm{G})(\mathrm{N?})(\mathrm{inl}(\mathrm{P}_1))(\mathrm{x}) \rightleftarrows_{\mathit{Update}}\\ &\qquad\qquad [\![\text{most/-est}]\!](\mathrm{G})(\mathrm{N?})(\mathrm{inl}(\mathrm{P}_2))(\mathrm{x})\end{aligned}$$

The implementation of (384) and (386) can be deduced from the action of $[\![\text{most/-est}]\!]$ on senses.

4.5 The positive

4.5.1 Background

The term *positive* refers to the unmarked form of a gradable adjective. As discussed in Chapter 1, the positive is *context-dependent*, meaning its interpretation depends on a choice of comparison class, and *vague*, meaning that it does not separate entities in its domain into two sharply-bounded sets. The usual analysis of the positive in the degree-based approach is in terms of a contextually determined standard value (Bartsch & Vennemann 1974, Cresswell 1976, von Stechow 1984, Klein 1980, Kennedy 1999, Kennedy & McNally 2005). For example, a sentence like *John is tall* can be paraphrased as "the degree to which John is tall exceeds some contextually-determined standard degree s". Context-dependence is explained by the dependence of the standard on a comparison class, whereas vagueness is explained by the fuzziness of the standard value.

A typical description of the positive was given by Bartsch & Vennemann (1974). As in this thesis, they assume that gradable adjectives denote measure functions. The positive is treated as a silent morpheme with the following denotation:

(387) $[\![\textsc{pos}]\!] \coloneqq \lambda g \,.\, \lambda p \,.\, \lambda x \,.\, g(x) > \mathbf{norm}(p)(g)$

where g is a gradable adjective, p is a property representing the comparison class, and $\mathbf{norm} : (e \to t) \to (e \to d) \to d$ is a function which returns the average degree to which members of p are f. A sentence like *John is tall for a basketball player* is then interpreted as follows:

(388) $[\![\text{John is } \textsc{pos} \text{ tall for a basketball player}]\!]$

$= [\![\textsc{pos}]\!]([\![\text{tall}]\!])([\![\text{basketball player}]\!])([\![\text{John}]\!])$

$= \mathbf{tall}(\mathbf{John}) > \mathbf{norm}(\mathbf{basketball\text{-}player})(\mathbf{tall})$

In other words, the sentence is true iff John's degree of height is greater than that of the average height of a basketball player.

Other authors disagree that the comparison class should be treated as a logical argument. Kennedy (2007) provides an alternative analysis in which the only arguments of the positive are a measure function and an individual, as follows:

(389) $[\![\textsc{pos}]\!] \coloneqq \lambda g \,.\, \lambda x \,.\, g(x) > \mathbf{s}(f)$

where **s** is a function from gradable adjectives to degrees that "chooses a standard of comparison in such a way as to ensure that the objects that the positive form is true of 'stand out' in the context of utterance, relative to the kind of measurement that the adjective encodes." (ibid., p. 17). According to Kennedy, an accompanying noun provides a context in which the property denoted by the noun is highly salient and therefore has an effect on $\mathbf{s}(f)$, but this is a matter of pragmatic convention rather than composition. As evidence for this, Kennedy points to sentences like the following

(390) John is a tall basketball player, but he isn't tall for a basketball player. In fact, he's the shortest player in the tournament.

where the effect of the noun denotation on the standard is explicitly cancelled.

Turning to *for*-PPs, Kennedy proposes that, rather than functioning as an argument to the positive, the *for*-phrase actually combines directly with the adjective by restricting its domain. For instance, *tall for a basketball player* would have the following interpretation:

(391) $[\![\text{tall for a basketball player}]\!] = \lambda x : [\![\text{basketball player}]\!](x)\ .\ [\![\text{tall}]\!](x)$

That is, it is a function from basketball players to degrees of height, given by restricting the original *tall* function to apply only to basketball players. According to Kennedy, instead of supplying the *for* expression as an argument to the positive morpheme, we should instead apply the positive morpheme to the entire adjective phrase – the adjective plus its modifiers – as follows:

(392) $[\![\textsc{pos}\ \text{tall for a basketball player}]\!]$

$$= [\![\textsc{pos}]\!]([\![\text{tall for a basketball player}]\!])$$

$$= \lambda x\ .\ [\lambda y : [\![\text{basketball player}]\!](y).[\![\text{tall}]\!](y)](x) \geq \mathbf{s}([\lambda y : [\![\text{basketball player}]\!](y).[\![\text{tall}]\!](y)])$$

Note that this analysis correctly predicts a presupposition failure when *tall for a basketball player* is applied to an individual who is not a basketball player.

I disagree with Kennedy's claim that the comparison class is not a logical argument. In my view, Kennedy's analysis of the *for*-PP as restricting the domain of the gradable adjective function raises more problems than it solves. If the expression *tall for a basketball player* denotes a gradable adjective, then we might expect it to be compatible with other morphemes which take gradable adjective meanings. Hence, Kennedy's proposal would seem to predict sentences like the following:

(393) a. # John is 190cm [tall for a basketball player].
b. # John is more (/ less) [tall for a basketball player] than Mary.
c. # John is the most (/ the least) [tall for a basketball player].

Instead, only a few degree morphemes are compatible with a *for*-PP:

(394) a. John is very tall for a basketball player.
b. John is too tall for a basketball player.
c. John is tall enough for a basketball player.

The most straightforward explanation of this distribution is that POS, *very*, *too* and *enough* take an optional *for*-PP argument, whereas other degree morphemes do not. It is worth noting that the principle motivation for Kennedy's analysis is to explain the presupposition that the subject belongs to the comparison class via a domain restriction on the adjective. However, in a system such as ours, where presuppositions are not limited to domain restrictions on functions, this can be represented in a different way.

4.5.2 Approaches to vagueness

The central phenomenon associated with the positive morpheme, and the reason why it has received so much theoretical attention, is vagueness. Vagueness is considered a problem in in philosophy of language because it gives rise to a class of paradoxes, exemplified in (395):

(395) The Sorites Paradox

P1. Anyone with a height of 200cm is a tall person.

P2. Anyone with a height 1mm less than that of a tall person is tall.

C. Therefore, everyone with a height less than 200cm is a tall person (by repeated application of P2).

The premises of the argument appear to be true and the reasoning valid, but the conclusion is clearly false: for instance, it implies that someone with a height of 50cm is a tall person. Any gradable predicate will give rise to a Sorites-type argument in a similar way. A resolution of this paradox is generally considered the the primary desideratum for a theory of vagueness. Most approaches are aimed at rejecting the induction step (P2).

One approach, known as *epistemicism*, denies the induction step on the grounds of ignorance (e.g. Williamson & Simons 1992, Williamson 2002). The idea is that a predicate like *tall person* does in fact have a definite extension with a precise boundary; language users are simply unaware of the exact location of this boundary. Given this epistemic gap, the inductive step cannot be accepted, since incrementally adding 1mm might take us over the boundary. The major challenge for this approach is explaining how sharp boundaries come to be fixed given that speakers do not know where they lie. Williamson (2002) suggests that the extension of a vague term is fixed by how it is used, and can vary across different speech communities. Although speakers do not know exactly where the extension of a vague term begins and ends, they do have partial or probabilistic knowledge of this boundary, which is why some individuals can be categorized as a *tall person* with a very high degree of certainty.

It is difficult to square the epistemic account of vagueness with the conceptualist view of meaning adopted in this thesis. Epistemicism presupposes an externalist view of meaning, according to which the relation between a predicate and its denotation exists independently of whether it is grasped by a language user. For the externalist, just as *water* denotes the chemical H_2O even in a society with no knowledge of modern chemistry, *tall person* denotes a definite set of people even though no individual has precise knowledge of this set. For a conceptualist or internalist, on the other hand, the meaning of a word is completely fixed by how words are conceptualized by speakers, so there is no 'room' for an epistemic gap. Speakers who disagree about the extension of *tall person* simply have different concepts of *tall person*: there is no sense in which some speakers are right and others wrong.

Another approach to vagueness which is commonly encountered in the literature on gradable adjectives is *supervaluationism*, which characterises vagueness as resulting from the hyper-ambiguity of denotations (e.g. Fine 1975, Lewis 1982). The idea behind supervaluationism is that a vague expression like *tall person* has multiple different 'precisifications' corresponding to different exact denotations. For instance, *tall person* might be assigned the following precisifications:

(396) a. $[\![\text{tall person}]\!]_{\text{a}}$ = people with a height greater than 180cm
b. $[\![\text{tall person}]\!]_{\text{b}}$ = people with a height greater than 185cm
c. $[\![\text{tall person}]\!]_{\text{c}}$ = people with a height greater than 190cm
d. $[\![\text{tall person}]\!]_{\text{d}}$ = people with a height greater than 190.5cm
e. ...

In addition to those listed above, all intermediate cut-offs are also possible, giving us an infinite number of possible precisifications. A sentence is considered 'supertrue' iff it is true under every precisification, and 'superfalse' iff it is false under every precisification. Sentences which are true for some precisifications and false for others are neither supertrue nor superfalse and are considered borderline cases. Replacing truth with supertruth and falsity with superfalsity leads to a rejection of the inductive step of the Sorites argument, since by successively subtracting 1mm we can no longer guarantee that a person is tall under all precisifications. Adopting the supervaluationist view of truth leads to a non-classical logic in which there are truth-value gaps. An alternative to supervaluationism is *subvaluationism*, whereby a predicate is considered true simpliciter iff it is true under at least one disambiguation, giving rise to a logic in which some sentences are both true and false (Hyde & Colyvan 2008, Cobreros 2011).

One could imagine a conceptualist version of supervaluationism whereby vague terms are hyper-ambiguous, with multiple different 'conceptual precisifications': a kind of ultra fine-grained polysemy. However, this is not the approach to vagueness which I shall adopt. One issue with the supervaluation approach from a conceptualist point of view is that the truth value gap is treated discontinuously. There is always a precise height (the lowest precisification) at which *tall person* goes from being false to being indeterminate, and a precise height (the highest precisification) at which it goes from being indeterminate to being true; but people's intuitive judgements do not show this kind of discontinuous behaviour. Hence, supervaluationism merely replaces one imprecise boundary problem – the location of boundary between *tall* and *not tall* – with two imprecise boundary problems – the locations where the truth value of *tall* begins and ends. A related issue is that there is no account of variation across the gap – someone with a height at the higher edge of the gap (e.g. 179cm) is more likely to be judged as a *tall person*, whereas someone with a height at the lower edge (e.g. 169cm) is very unlikely to be judged as a *tall person*.

A third approach to vagueness, distinct from both epistemicism and supervaluationism, is to adopt a many-valued logic from the start. Some authors opt for a three-valued logic whereby each predicate divides entities into a positive extension, a negative extension and a truth value gap (e.g. Tye 1994, Field 2003). However, like supervaluationism, this approach also has problems relating to the discontinuity of truth value gaps. A more popular solution is to use an infinite valued or 'fuzzy' logic, where a truth value can be any real number from 0 ('completely false') to 1 ('completely true') (e.g. Goguen 1969, Zadeh 1975). Instead of a truth value gap, this results in a truly fuzzy boundary where the degree of truth increases continuously as one approaches the boundary region. On this view, one can reject the induction step of the Sorites argument because although subtracting 1mm from a person's height has only a small effect on their membership in the set of tall people, iterating this can have a large effect, ultimately resulting in a truth value close to 0.

Despite the intuitive treatment of vagueness in fuzzy logic, this approach is not widely adopted because it is difficult to define fuzzy versions of the logical connectives so that they behave intuitively. In fuzzy logic, the truth value of a conjunction is usually defined as the minimum of the truth value of the two conjuncts, whereas the truth value of a disjunction is the maximum. This has strange results: for instance, if *John is tall* has a truth value of 0.7, then *John is tall and not tall* also has a truth value of 0.7, when intuitively speaking it ought to have a truth value of 0. Another consequence is that adding a 'hedging expression' does not increase the truth value of a sentence – if *John is tall* has a truth value of 0.7, then *John is tall or of average height* also has a truth value of 0.7, despite our intuition that it ought to be more truthful. Given these problems, it is generally agreed that fuzzy logic, at least in its traditional formulation, cannot serve as a psychologically realistic model for vague concepts (e.g. Osherson & Smith 1981, Kamp & Partee 1995).

An interesting approach to vagueness which has been developed in the context of modern type theory is Cooper et al.'s (2015) Probabilistic Type Theory with Records (PTTR) framework. PTTR is a model-theoretic version of dependent type theory in which the extensions of types are given in terms of a background set theory. As in this thesis, propositions are encoded as types and are considered true iff they are inhabited. Every typing judgment $a : A$ is assigned a probability $p(a : A)$, which is the probability that the element a is of type A. Rather than product and sum types, PTTR uses conjunction ($A \wedge B$) and disjunction types ($A \vee B$), where the probability that an element belongs to a conjunction or disjunction is given by the familiar Kolmogorov rules for probabilities:

(397)
- $p(a : A \wedge B) = p(a : A) * p(a : B \mid a : A)$ (conjunction)
- $p(a : A \vee B) = p(a : A) + p(a : B) - p(a : A \wedge B)$ (disjunction)

The probability that a type A is inhabited ('true') is given by the infinite disjunctive probability of $a : A$ for all a:

(398) $p(A) = \bigvee_i a_i : A$

As in fuzzy logic, PTTR leads to a rejection of the inductive step of the Sorites argument because by successively adding 1mm to John's height the probability that he is a *tall person* decreases monotonically. Unlike fuzzy logic, however, PTTR does not give rise to unintuitive results associated with connectives. A proposition like *John is tall and John is not tall* has a probability of 0, just as we would expect. Moreover, if *John is tall* has a probability of 0.7 and *John is of average height* has a probability of 0.25, then *John is tall or of average height* will have a probability of $0.7 + 0.25 = 0.95$ (assuming the probability of John being both *tall* and *of average height* is 0).

In PTTR, judging the truth of a vague predicate is modelled as a perceptual classification task (Fernández & Larsson 2014, Larsson 2015, Cooper et al. 2015).

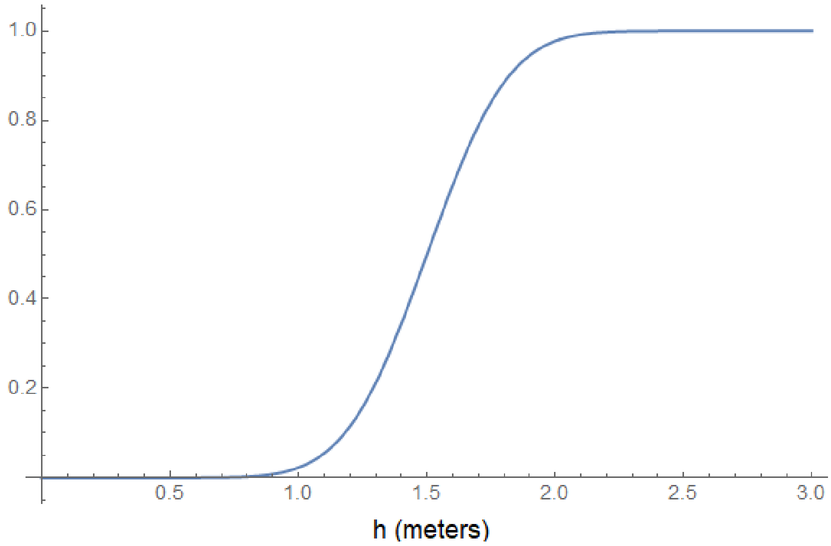


Figure 4.5: A plot of $P(h \text{ is tall} \,|\, \mu_{\text{tall}} = 1.5\text{m}, \sigma_{\text{tall}} = 0.25\text{m})$ for increasing values of h.

Each vague predicate is associated with a distinct classifier, where the classifier for a predicate P takes relevant information from the background context and returns a probability (degree of credence) that P is true in that context. Language users rely on these classifiers to compute and update their belief in others' assertions; as well as deciding what assertions they should make themselves. Fernández & Larsson (2014) apply this idea to positive gradable adjectives, using *tall* as an example. They propose that, given an individual x with height h, a speaker computes their degree of belief that x *is tall* by applying a probabilistic threshold-based classifier. The threshold is modelled as a normal random variable which depends on the mean height μ_{tall} and the standard deviation σ_{tall}, both of which are estimated from the relevant comparison class. The probability that the subject's height h exceeds the threshold is given by:

(399) $$p(h \text{ is tall} \,|\, \mu_{\text{tall}}, \sigma_{\text{tall}}) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{h - \mu_{\text{tall}}}{\sigma_{\text{tall}}\sqrt{2}}\right)\right]$$

where erf is the Gaussian error function. To illustrate, Figure 4.5 shows the probability that h is tall for increasing values of h, given $\mu = 1.5\text{m}$ and $\sigma = 0.25\text{m}$. Notice how heights above 2m are judged as *tall* with a very high credence, whereas those below 1m are judged as *tall* with a very low credence. In-between, there is an area of uncertainty, with the mean itself having a credence of 0.5. There is some evidence that human judgements conform to this statistical model. For example, Schmidt et al. (2009) presented people with collections of items of different heights and collected data on which items in each collection were judged to be *tall*. They found that human judgements were well predicted by a threshold-based statistical model with Gaussian noise (although an exemplar-based model performed equally well).

Unfortunately, PTTR as standardly formulated is incompatible with the kind of type theory used in this thesis. PTTR is model-theoretic rather than proof-theoretic: it assumes a background set theory which is used to provide extensions for types. As a result of this set-up, a term may belong to more than one type simultaneously, and type checking is no longer decidable. Nevertheless, it might be possible to implement something like the PTTR approach to vagueness within a proof-theoretic framework. Such a type theory would replace or supplement ordinary typing judgements with probabilistic judgements of the form:

(400) $x :_p A$

meaning 'x is evidence of A, with probability p'. Unlike in PTTR, each piece of evidence would be associated with one and only one type, so the following kind of context would not be well-formed:

(401) $x :_{p_1} A,\ x :_{p_2} B$

Rather, one would need to convert evidence for A into evidence for B by means of a probabilistic inference $f :_{p_3} A \to B$. An important goal of such a system would be to preserve the behaviour of ordinary, non-probabilistic types, which are still needed in many circumstances (e.g. compositionality in semantics). One possibility would be to replace every classical typing judgement $x : A$ with a judgement $x :_1 A$ of probability 1, as in PTTR. Another possibility might be to introduce two disjoint type universes: one for probabilistic types and one for non-probabilistic types. Details aside, the development of a probabilistic version of Martin-Löf Type Theory is an important goal for future research; in my view, such a system would be the most appropriate framework for studying vagueness from a conceptualist/internalist point of view.

4.5.3 Action on senses

Having discussed the issue of vagueness, let us now turn to the formulation of the positive morpheme itself. Following other authors, I assume that the positive is derived through a phonologically null morpheme POS, or alternatively a type shifting rule with the same semantics. As discussed in Section 4.5.1, the positive morpheme takes an optional comparison class argument, which can be supplied either by a *for*-PP or by an accompanying noun. The latter is due to a default rule, and can be explicitly cancelled in an example like:

(402) John is a tall basketball player, but he isn't tall for a basketball player.

When no comparison class argument is provided, the presuppositions should contain an open comparison class variable, whose value is supplied by the background context. These different compositional options would be listed in the lexical entry of POS, which we can represent informally as follows:

(403) POS

a. No comparison class argument:
   syntax: $[_{\text{DegP}}$ ___ A $]$
   semantics: $[\![\textsc{pos}]\!]([\![\text{A}]\!])(\text{inr}(*))$

b. *For*-PP argument:

syntax: $[_{\text{DegP}}\ [_{\text{Deg}'}\ \_\_\_\ \text{A}\ ]\ [_{\text{PP}}\ \text{for a(n) NP}\ ]\ ]$
semantics: $[\![\textsc{pos}]\!]([\![\text{A}]\!])(\text{inl}([\![\text{NP}]\!]))$

c. Accompanying noun (default rule):
syntax: $[_{\text{NP}}\ [_{\text{DegP}}\ \_\_\_\ \text{A}\ ]\ \text{N}'\ ]$
semantics: $([\![\textsc{pos}]\!]([\![\text{A}]\!])(\text{inl}([\![\text{N}']\!]))$

As before, these different compositional options would be handled by a flexible approach to the syntax-semantics interface such as Glue Semantics (Asudeh et al. 2012).

The positive morpheme takes a gradable adjective, an optional NP meaning, and an individual, and returns a sentence meaning, so it is typed as follows:

(404) $[\![\textsc{pos}]\!] : \mathit{Gradable} \to (\mathit{Ind} \to \mathit{Type})? \to \mathit{Ind} \to \mathit{Update}$

A potential implementation is given below:

(405)
$$[\![\textsc{pos}]\!] \coloneqq \lambda \text{G} : \mathit{Gradable}\ .\ \lambda \text{N?} : (\mathit{Ind} \to \mathit{Type})?\ .\ \lambda \text{x} : \mathit{Ind}\ .\ \left(\begin{array}{l} \text{P} \coloneqq \begin{bmatrix} \text{C} : \mathit{Ind} \to \mathit{Type} \\ \text{case N? of} \\ \quad \text{inl(N) then C} = \text{N} \\ \quad \mid \text{inr}(*) \text{ then } \top \\ \text{C(x)} \\ \text{G(x).1} \\ \text{d}_\mu : \mathit{Degree} \\ \text{mean}(\text{d}_\mu, \text{G}, \text{C}) \\ \text{d}_\sigma : \mathit{Degree}(+) \\ \text{stdev}(\text{d}_\sigma, \text{G}, \text{C}) \end{bmatrix} \\ \lambda \text{p} : \text{P}\ .\ \text{exceed}(\text{G(x).2(p.4).3, p.5.3, p.7.2}) \end{array} \right)$$

As shown, given a gradable adjective G, an optional noun meaning N?, and an individual x, we have the following presuppositions:

(406)
1. there is some property C, the comparison property
2. if the N? argument contains a non-empty value N, then C = N
3. x satisfies C
4. x satisfies the presuppositions of G
5. there is some degree d_μ
6. d_μ is the average (mean) degree to which elements of C are G
7. there is some positive degree d_σ
8. d_σ is the standard deviation of elements of C with respect to G

Note that the mean can be either a positive or a negative degree, depending on the polarity of the adjective, whereas the standard deviation is always a positive degree. This is because the standard deviation is the average distance or separation from the mean, which is always a positive quantity. Given a context p satisfying the conditions in (406), the assertion is:

(407) $\mathrm{exceed}(\mathrm{G}(\mathrm{x}).2(\mathrm{p}.4).3, \mathrm{p}.5.3, \mathrm{p}.7.2)$

which encodes the proposition 'the degree to which x is G exceeds a normal random threshold with mean d_μ and standard deviation d_σ'.

To constrain the mean and standard deviation, I assume predicates:

(408) $\mathrm{d} : \mathit{Degree},\ \mathrm{G} : \mathit{Gradable},\ \mathrm{C} : \mathit{Ind} \to \mathit{Type} \vdash \mathrm{mean}(\mathrm{d}, \mathrm{G}, \mathrm{C}) : \mathit{Prop}$

$\mathrm{d} : \mathit{Degree}(+),\ \mathrm{G} : \mathit{Gradable},\ \mathrm{C} : \mathit{Ind} \to \mathit{Type} \vdash \mathrm{stdev}(\mathrm{d}, \mathrm{G}, \mathrm{C}) : \mathit{Prop}$

The issue of how the appropriate mean and standard deviation are estimated from the comparison class is a difficult one. In their PTTR-based analysis of the positive, Fernández & Larsson (2014) assume that language users have access to a data set of previous observations for each adjective + comparison class combination (e.g. heights of people, temperatures of buildings, weights of fruit, and so on), from which they can compute the most likely mean and standard deviation for that adjective and class. However, it is implausible that a speaker would have access to such a data set for every adjective + comparison class pair. Even if we grant that speakers can record every observation – which seems unlikely – most adjective + comparison class pairs will have an empty observation set, being entirely novel combinations. In most cases, therefore, we must assume that speakers do not have precise knowledge of the relevant mean and standard deviation, but are able to simply accommodate this information. For example, a speaker who is told "John is a tall circus performer" may be unsure as to the appropriate mean and standard deviation for the height of circus performers, but understands that *whatever the mean and standard deviation happen to be*, John's height is unusually large in comparison.

The positive assertion itself is described by the predicate:

(409) $\mathrm{d}, \mu : \mathit{Degree}(\mathrm{s}, \mathrm{p}), \sigma : \mathit{Degree}(\mathrm{s}, +) \vdash \mathrm{exceed}_{\mathrm{s,p}}(\mathrm{d}, \mu, \sigma) : \mathit{Prop}$

That is, for every scale s and polarity p, there is a predicate $\mathrm{exceed}_{\mathrm{s,p}}(\mathrm{d}, \mu, \sigma)$ meaning 'd exceeds a normal random threshold with mean μ and standard deviation σ'. This predicate is vague, meaning that a speaker cannot construct a definite proof of $\mathrm{exceed}_{\mathrm{s,p}}(\mathrm{d}, \mu, \sigma)$ but rather judges it to be true with the following degree of credence:

(410) $$\mathrm{p}[\mathrm{exceed}_{\mathrm{s,p}}(\mathrm{d}, \mu, \sigma)] \coloneqq \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mathrm{d} - \mu}{\sigma\sqrt{2}}\right)\right]$$

where erf is the Gaussian error function. As discussed in the previous subsection, this yields an S-shaped curve, where values much lower than the mean have probabilities close to 0, values much higher than the mean have probabilities close to 1, and there is a region of uncertainty surrounding the mean whose width depends on the standard deviation.

As before, it is helpful to illustrate the action of the positive by means of an example. Consider the expression *tall for a basketball player*, where *tall* is interpreted in its most prototypical sense as $[\![\text{tall}]\!]_{\text{up, 1st, large}}$:

(411) $[\![\textsc{pos}\text{ tall for a basketball player}]\!]_{\text{up, 1st, large}}$

$$= [\![\textsc{pos}]\!](\text{as\_pos}'([\![\text{tall}]\!]_{\text{up, 1st, large}}))(\text{inl}([\![\text{basketball player}]\!]))$$

$$= \lambda \text{x} : \mathit{Ind}\ .\ \left(\begin{array}{l} \text{P} := \begin{bmatrix} \text{C} : \mathit{Ind} \rightarrow \mathit{Type} \\ \text{C} = [\![\text{basketball player}]\!] \\ \text{C(x)} \\ [\![\text{tall}]\!]_{\text{up, 1st, large}}(\text{x}).1 \\ \text{d}_\mu : \mathit{Degree} \\ \text{mean}(\text{d}_\mu, [\![\text{tall}]\!]_{\text{up, 1st, large}}, \text{C}) \\ \text{d}_\sigma : \mathit{Degree}(+) \\ \text{stdev}(\text{d}_\sigma, [\![\text{tall}]\!]_{\text{up, 1st, large}}, \text{C}) \end{bmatrix} \\ \lambda \text{p} : \text{P}\ .\ \text{exceed}([\![\text{tall}]\!]_{\text{up, 1st, large}}(\text{x}).2(\text{p}.4).3,\ \text{p}.5.3,\ \text{p}.7.2) \end{array}\right)$$

The resulting update has the following presuppositions:

(412)
1. there is some predicate C
2. C is equal to $[\![\text{basketball player}]\!]$
3. x is a basketball player
4. there is some degree d_μ
5. d_μ is the average degree to which elements of C satisfy $[\![\text{tall}]\!]_{\text{up, 1st, large}}$
6. there is some degree d_σ
7. d_σ is the standard deviation of elements of C with respect to $[\![\text{tall}]\!]_{\text{up, 1st, large}}$

Given a context which satisfies these presuppositions, the assertion is that the height of x exceeds the normal random threshold given by μ and σ.

There are various ways in which the positive can be unacceptable, which the implementation in (405) is intended to explain. A positive assertion is always infelicitous if the subject fails to satisfy the adjective presuppositions, in which case the fourth component of the presupposition type will be empty. For example, the sentence

(413) ? The Nile is tall.

is infelicitous because the Nile does not satisfy the background conditions for *tall*. Another way in which the positive can be infelicitous is if the subject does not belong to the comparison class. In this case, it is the third component of the presupposition type which is empty. For instance, the sentence

(414) ? The Nile is long for a road.

is infelicitous because the Nile is not a road. Finally, the positive will also be infelicitous if the comparison class is incompatible with the adjective, so that it is impossible to find a corresponding mean and standard deviation. For example, it sounds strange to say

(415) ? McDonalds is tall for a company.

because companies are not usually the kind of thing which can be assigned a tallness. As a result, the comparison class does not have an associated mean or standard deviation, and the sixth and eighth components of the presupposition type are empty.

In a combination of the form [POS A] N (e.g. [POS *tall*] *mountain*) the interpretation of N is intersected with the interpretation of [POS A] by means of ordinary adjective + noun intersection. Following the description of adjective + noun intersection given in Section 2.5.4 – where it is modelled as a kind of local context update – such a combination is only valid if N can supply the presuppositions of [POS A]. The following combinations are therefore all invalid (outside of certain very specific contexts):

(416) a. ? [POS tall] river
b. ? [POS long] sphere
c. ? [POS thick] cavity
d. ? [POS deep] lamppost
e. ? [POS wide] point

because, in each case, the noun is unable to supply the background conditions associated with A, which are part of the presuppositions of [POS A].

Unlike the other morphemes discussed up to this point, the positive does not act on gradable adjective arrows. To see this, imagine looking at a tabletop on which there are a variety of cups of different sizes, some standing upright and some toppled over. Someone points to one of the cups and says "This one is deep". The extent to which one agrees with the speaker will depend on how the adjective *deep* is interpreted. The sense $[\![\text{deep}]\!]_{\text{intrnl, down}}$ will take into account only those cups which are standing upright, so that their depth is directed down, whereas the weaker sense $[\![\text{deep}]\!]_{\text{intrnl}}$ will take into account the internal dimensions of all the cups, regardless of orientation. Going from the stronger, more restricted sense to the weaker, less restricted sense can have an unpredictable effect on the standard value – it might increase or decrease, depending on the depth of the toppled cups. Hence, knowing that a cup is deep in the stronger sense tells you nothing about whether it is deep in the weaker sense, and vice versa. Formally speaking, we can see that POS does not lift arrows between gradable adjectives because of the presence of predicates in its presupposition type which depend directly on the adjective, namely the 'mean' and 'stdev' predicates. POS also does not lift arrows between its comparison class argument, for the same reason.

4.6 Some other degree morphemes

4.6.1 As... as...

The *as... as...* construction shares a number of features with the comparative morphemes *more/-er* and *less*, but instead of asserting that the reference degree is strictly greater than the comparison degree, it asserts that the reference degree is *greater than or equal to* the comparison degree. For instance, if John's height is 150cm, then *Mary is as tall as John* iff her height is 150cm or more. The fact that *as... as...* is weaker than the comparative is shown by sentences like the following:

(417) Mary is as tall as John. In fact, she is taller.

where the possibility that Mary is exactly the same height as John is explicitly cancelled. In contrast to the measure morpheme MEAS, *as... as...* is never used to assert strict equality. For example, (a) is fine but (b) is contradictory:

(418) a. Mary isn't 150cm tall. She is 155cm tall.
b. * Mary isn't as tall as 150cm. She is 155cm tall.

Like the comparative morphemes *more/-er* and *less*, *as... as...* is associated with a number of compositional options, occurring with a nominal argument, a degree argument, or an entire phrase:

(419) a. Mary is as tall as John.
b. Mary is as tall as 150cm.
c. Mary is as tall as the river is deep.

As in the case of the comparative, I shall assume that (b) and (c) have a similar underlying structure, the difference being that (b) provides a degree argument directly, whereas (c) does so through a kind of hidden definite description – *Mary is as tall as [the degree to which] the river is deep.* The three possibilities shown in (419) therefore correspond to two choices for the *than* argument: an individual or a degree. If a degree argument is provided, then it must agree in dimensions and polarity with the adjective; hence the following are unacceptable:

(420) a. ? John is as tall as the car is heavy. [dimensions disagree]
b. ? John is as tall as the car is narrow. [polarities disagree: $+, -$]
c. ? John is as short as the car is wide. [polarities disagree: $-, +$]

This is in contrast to the comparative, where a polarity disagreement is allowed provided that the comparison degree is positive, as in *John is shorter than the path is wide.*

These observations suggest that *as... as...* has the following semantic type:

(421) $[\![\text{as...as...}]\!] : \mathit{Gradable} \rightarrow (\mathit{Ind} + \mathit{Degree}(+)) \rightarrow \mathit{Ind} \rightarrow \mathit{Update}$

A potential implementation is given below:

(422) $[\![\text{as}\ldots\text{as}\ldots]\!] :=$

$$\lambda \mathrm{G} : \mathit{Gradable}\ .$$
$$\quad \lambda \mathrm{T} : (\mathit{Ind} + \mathit{Degree}(+))\ .$$
$$\quad\quad \lambda \mathrm{x} : \mathit{Ind}\ .$$
$$\left(\begin{array}{l} \mathrm{P} := \left[\begin{array}{l} \mathrm{c} : \mathrm{G(x).1} \\ \mathrm{d_{comp}} : \mathit{Degree} \\ \text{case T of} \\ \quad \text{inl(y) then} \left[\begin{array}{l} \mathrm{c'} : \mathrm{G(y).1} \\ \mathrm{d_{comp}} = \mathrm{G(y).2(c')} \end{array}\right] \\ \quad \mid \text{inr(d) then } \mathrm{d_{comp}} = \mathrm{d} \end{array}\right] \\ \lambda \mathrm{p} : \mathrm{P}\ .\ \mathrm{G(x).2(p.1).3} \geq \mathrm{p.2.3} \end{array}\right)$$

Given a gradable adjective G, a comparison argument T (individual or positive degree), and an individual x, we get the following presuppositions:

(423)
1. x satisfies the background presuppositions of G
2. there is some degree $\mathrm{d_{comp}}$ (the comparison degree)
3. if T contains some individual y, then y satisfies the background presuppositions of G and $\mathrm{d_{comp}}$ is the degree to which y is G; otherwise, if T contains some degree d then $\mathrm{d_{comp}} = \mathrm{d}$

Note the close similarity to the comparative presuppositions discussed in Section 4.3.2. Given a context which satisfies these constraints, the assertion is that the value of the reference degree is greater than or equal to the value of the comparison degree. This is a valid comparison only if the reference and comparison degree have the same dimensions and polarity.

Like many other degree morphemes, *as...as...* lifts arrows monotonically in its gradable adjective argument. Given two gradable adjectives $\mathrm{G}, \mathrm{H} : \mathit{Gradable}$ with an arrow:

(424) $\alpha : \mathrm{G} \rightarrow_{\mathit{Gradable}} \mathrm{H}$

we can lift this to an arrow:

(425) $[\![\text{as}\ldots\text{as}\ldots]\!](\alpha) :$

$$\prod_{\mathrm{T}:(\mathit{Ind}+\mathit{Degree}(+))}$$
$$\quad \prod_{\mathrm{x}:\mathit{Ind}}$$
$$\quad\quad [\![\text{as}\ldots\text{as}\ldots]\!](\mathrm{G})(\mathrm{T})(\mathrm{x}) \rightarrow_{\mathit{Update}} [\![\text{as}\ldots\text{as}\ldots]\!](\mathrm{H})(\mathrm{T})(\mathrm{x})$$

which is implemented as follows:

(426) $[\![\text{as}\ldots\text{as}\ldots]\!](\alpha) :=$

$$\lambda \mathrm{T} : (\mathit{Ind} + \mathit{Degree}(+))\ .$$

$$
\begin{array}{l}
\lambda \mathrm{x} : \mathit{Ind}\ . \\
\quad \text{let } \mathrm{G}' \coloneqq [\![\text{as}\ldots\text{as}\ldots]\!](\mathrm{G})(\mathrm{T})(\mathrm{x}) \text{ in} \\
\qquad \left(\begin{array}{l}
\lambda \mathrm{p} : \mathrm{G}'.1\ . \left[\begin{array}{l}
\alpha(\mathrm{x}).1(\mathrm{p}.1) \\
\mathrm{p}.2 \\
\text{case T of} \\
\quad \text{inl(y) then} \begin{pmatrix} \alpha(\mathrm{y}).1(\mathrm{p}.3.1) \\ \mathrm{p}.3.2 \end{pmatrix} \\
\quad \mid \text{inr}(\mathrm{d}') \text{ then p}.3
\end{array} \right] \\
\lambda \mathrm{p} : \mathrm{G}'.1\ . \ \mathrm{id}_{\mathrm{G}'.2(\mathrm{p})}
\end{array} \right)
\end{array}
$$

As expected, the lifted arrow uses the original arrow to replace presuppositions of G with presuppositions of H. The only components which need to be altered are the first and third components, since these alone contain presuppositions associated with the G. The assertion does not need to be altered, since it depends only on the output of the measure function, which remains unchanged when G is replaced by H.

4.6.2 Very

The intensifier *very* is also a degree morpheme (Wheeler 1972, Klein 1980, von Stechow 1984, Kennedy & McNally 2005). This is shown by the fact that *very* occurs in complementary distribution with other degree morphemes, including MEAS, *more/-er*, most/-est, *as... as*, and so on:

(427) a. * The road is very 1km long
b. * John is very taller than Mary
c. * John is very the tallest
d. * John is very as tall as Sally

Very closely resembles the positive morpheme POS. Like POS, it can occur with a bare adjective, an adjective + *for*-PP, or intersected with an accompanying NP:

(428) a. John is very tall.
b. John is very tall for a basketball player.
c. John is a very tall basketball player.

Following the analysis of the positive in Section 4.5, I shall assume that *very* has an optional comparison class argument, which can be saturated either by a *for*-PP or by the accompanying NP. The latter is handled by a default rule which can be cancelled, as in:

(429) John is a very tall basketball player, but he isn't very tall for a basketball player.

Very exhibits vagueness just like the positive. One can know John's exact height and have some fixed comparison class in mind, but still be unsure whether he should be considered *very tall* relative to that comparison class. Intuitively speaking, the assertive content of *very* should be identical to POS, except with a higher standard value. This raises the question of how much higher the standard value for *very* should be in comparison to POS. It is clear that *very* cannot raise the standard additively, since the amount by which it is raised depends on the comparison class. For instance, in *John is very tall*, the standard is raised by a small amount on the order of tens of centimeters, whereas in *that building is very tall*, the standard is raised by a much larger amount on the order of tens or perhaps even hundreds of meters. An alternative analysis is that *very* modifies the standard value by multiplying it by a constant factor, say around 1.4. However, this is also problematic because it ignores the standard deviation. For instance, a person only needs to be around 1.2 times taller than average height for them to be considered very tall, whereas we would not consider a building to be very tall unless it was about twice as tall as average.

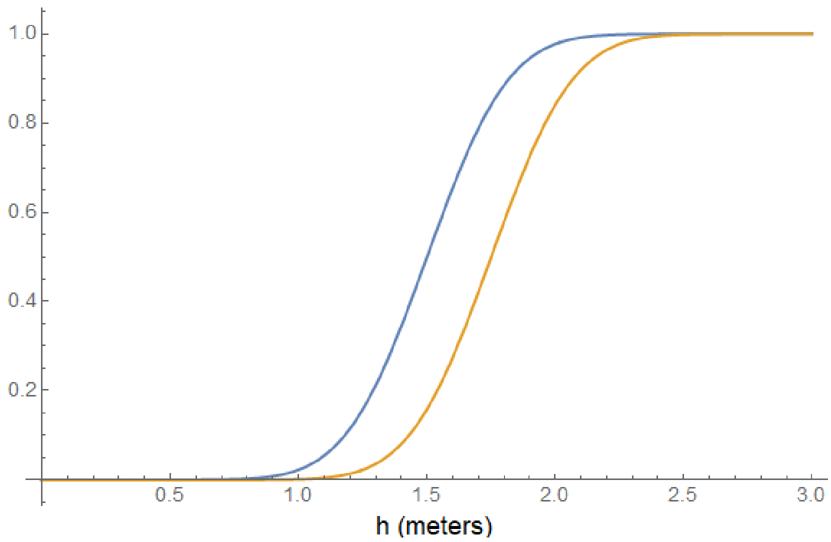


Figure 4.6: The probability of *tall* (blue) and *very tall* (yellow), for different reference heights, assuming $\mu = 1.5$m and $\sigma = 0.25$m

A better analysis would be to shift the standard by an amount proportional to the standard deviation. Instead of requiring the reference degree to be greater than the mean, we could instead require it to be greater than the mean plus one standard deviation. For example, suppose that the average height of people is 1.5m, with a standard deviation of 0.25m. Figure 4.6 shows the probability that a height is *tall* and the probability that it is *very tall*, for different reference heights. Notice how the probabilistic threshold for *very tall* is shifted to the right by one standard deviation, so that it is centered at 1.75m rather than 1.5m. This has a large effect for values close to the mean: for instance, a height of 1.7m has a probability of around 0.79 for *tall*, but a probability of only around 0.42 for *very tall*. For categories with a larger standard deviation, the amount by which the threshold gets shifted will be larger.

The morpheme *very* has the same type as the positive: $Gradable \rightarrow (Ind \rightarrow Type)? \rightarrow Ind \rightarrow Update$. My proposed implementation is given below:

$$
(430)\quad [\![\text{very}]\!] \coloneqq \lambda \mathrm{G} : \mathit{Gradable}\ .\\
\lambda \mathrm{N?} : (\mathit{Ind} \to \mathit{Type})?\ .\\
\lambda \mathrm{x} : \mathit{Ind}\ .\\
\left(\begin{array}{l}
\mathrm{P} \coloneqq \begin{bmatrix}
\mathrm{C} : \mathit{Ind} \to \mathit{Type} \\
\text{case N? of} \\
\quad \text{inl(N) then C} = \mathrm{N} \\
\quad \mid \text{inr}(*) \text{ then } \top \\
\mathrm{C(x)} \\
\mathrm{G(x).1} \\
\mathrm{d}_\mu : \mathit{Degree} \\
\text{mean}(\mathrm{d}_\mu, \mathrm{G}, \mathrm{C}) \\
\mathrm{d}_\sigma : \mathit{Degree}(+) \\
\text{stdev}(\mathrm{d}_\sigma, \mathrm{G}, \mathrm{C})
\end{bmatrix} \\
\lambda \mathrm{p} : \mathrm{P}\ .\ \text{exceed}(\mathrm{G(x).2(p.4).3},\ \mathrm{p.5.3} + \mathrm{p.7.2},\ \mathrm{p.7.2})
\end{array}\right)
$$

The presuppositions of $[\![\text{very}]\!]$ are identical to those of $[\![\textsc{pos}]\!]$:

(431)
1. there is some predicate C, representing the comparison property
2. if the N? argument contains a non-empty value N, then C = N
3. x satisfies C
4. x satisfies the presuppositions of G
5. there is some degree d_μ
6. d_μ is the average (mean) degree to which elements of C are G
7. there is some positive degree d_σ
8. d_σ is the standard deviation of elements of C with respect to G

The assertion of $[\![\text{very}]\!]$ is similar to $[\![\textsc{pos}]\!]$, except that the degree to which x is G must exceed a fuzzy threshold centered at the mean *plus one standard deviation*. As before, this is encoded by means of the vague predicate 'exceed', giving rise to the behaviour illustrated in Figure 4.6.

Like the positive, *very* does not lift arrows in either its gradable adjective argument or its comparison class argument. For example, suppose you know that a certain building is *very short*, where *short* is interpreted in the sense of $[\![\text{short}]\!]_{\text{vert, 1st}}$, that is an upwards-directed primary axis. This does not allow you to conclude that the building is *very short* in the sense of $[\![\text{short}]\!]_{\text{1st}}$ because the mean and standard deviation for $[\![\text{short}]\!]_{\text{1st}}$ may be different from $[\![\text{short}]\!]_{\text{vert, 1st}}$. Nor is the reverse inference valid, from $[\![\text{short}]\!]_{\text{vert, 1st}}$ to $[\![\text{short}]\!]_{\text{1st}}$.

4.6.3 Completely and half

Some degree morphemes are sensitive to scale topology, requiring scales which are bounded above, below, or both (Paradis 2001, Kennedy & McNally 2005). Recall that the topology of scales is that same as that of intervals. A scale has two directions – right and left – both of which may be either bounded or unbounded. For a bounded direction, the scale is said to be open in that direction if it includes the boundary point, and closed if it excludes the boundary point. This gives rise to eight different possibilities, which are listed below, together with their representative intervals:

(432) a. left- and right-bounded:
  i. open: $(0, 1)$
  ii. closed: $[0, 1]$
  iii. left-open, right-closed: $(0, 1]$
  iv. left-closed, right-open: $[0, 1)$
 b. left-bounded:
  i. open: $(0, \infty)$
  ii. closed: $[0, \infty)$
 c. right-bounded:
  i. open: $(-\infty, 0)$
  ii. closed: $(-\infty, 0]$
 d. left- and right-unbounded: $(-\infty, \infty)$

Different scales are associated with different topologies. For instance, the distance scale has the topology $(0, \infty)$, being bounded and open in both directions; the scale of age has the topology $[0, \infty)$, being bounded and closed on the left and bounded and open on the right; and the scale of 'fullness', as in *the glass is half full*, has the topology $[0, 1]$, being bounded and closed in both directions.

Morphemes such as *completely* and *totally* refer to the rightmost point on a scale which is right-bounded and closed. Consider the following combinations:

(433) a. completely full
 b. completely empty
 c. completely new
 d. ? completely old
 e. ? completely tall
 f. ? completely short

Completely is compatible with *full* because the scale of fullness is right-bounded and closed, having topology $[0, 1]$. Reversing a $[0, 1]$ scale gives a scale of the same topology, so *completely* can also occur with the antonym *empty.* The newness scale has topology $(-\infty, 0]$ and the oldness scale has topology $[0, \infty)$, so *completely new* is acceptable but *completely old* is unacceptable. The tallness scale has topology $(0, \infty)$ and the shortness scale has topology $(-\infty, 0)$, so both *completely tall* and *completely short* are unacceptable.

Another degree morpheme which is sensitive to topology is *half*, which requires a scale that is bounded and closed on both the left and the right, given a well-defined half-way point. Consider the following examples:

(434) a. half full
b. half empty
c. ? half old
d. ? half new
e. ? half tall
f. ? half short

As shown, only the combinations *half full* and *half empty* are acceptable, since the fullness and emptiness scales are bounded in both directions. In contrast, *old* and *tall* are bounded only on the left, whereas *new* and *short* are bounded only on the right.

Completely and *half* have the same semantic type:

(435) $[\![\text{completely}]\!], [\![\text{half}]\!] : \mathit{Gradable} \to \mathit{Ind} \to \mathit{Update}$

A proposed implementation for *completely* is given below:

(436)
$$[\![\text{completely}]\!] \coloneqq \lambda \text{G} : \mathit{Gradable}\ .\ \lambda \text{x} : \mathit{Ind}\ .\ \left(\begin{array}{l} \text{P} \coloneqq \begin{bmatrix} \text{c} : \text{G(x).1} \\ \text{d}_{\text{ref}} : \mathit{Degree} \\ \text{d}_{\text{ref}} = \text{G(x).2(c)} \\ \text{d}_{\text{max}} : \mathit{Degree}(\text{d}_{\text{ref}}.2,\ \text{d}_{\text{ref}}.1) \\ \prod_{\text{d}:\mathit{Degree}(\text{d}_{\text{ref}}.2,\ \text{d}_{\text{ref}}.1)} \text{d}.3 \leq \text{d}_{\text{max}}.3 \end{bmatrix} \\ \lambda \text{p} : \text{P}\ .\ \text{p}.2.3 = \text{p}.4.3 \end{array} \right)$$

Given a gradable adjective G and an individual x, the presuppositions are as follows:

(437)
1. x satisfies the background presuppositions of G
2. there is some degree d_{ref} (the reference degree)
3. d_{ref} measures the degree to which x is G
4. there is some degree d_{max} (the maximum degree) with the same polarity and dimensions as the reference degree
5. d_{max} is maximal, in the sense that it is greater than every other degree on the same scale

Given a context satisfying these conditions, the main assertion of *completely* is that the magnitude of d_{ref} is equal to the magnitude of d_{max}. Note that if the scale does not permit a maximum degree, as in ?*completely tall*, then there will be no context satisfying the presuppositions and the resulting update will be infelicitous.

The morpheme *half* requires both a maximum and a minimum. It can be implemented as follows:

(438) $[\![\text{half}]\!] \coloneqq \lambda \text{G} : \mathit{Gradable}\ .$
$\lambda \text{x} : \mathit{Ind}\ .$

$$\left(\begin{array}{l} \mathrm{P} \coloneqq \left[\begin{array}{l} \mathrm{c} : \mathrm{G(x).1} \\ \mathrm{d_{ref}} : \mathit{Degree} \\ \mathrm{d_{ref}} = \mathrm{G(x).2(c)} \\ \mathrm{d_{max}} : \mathit{Degree}(\mathrm{d_{ref}.2},\ \mathrm{d_{ref}.1}) \\ \prod_{\mathrm{d}:\mathit{Degree}(\mathrm{d_{ref}.2},\mathrm{d_{ref}.1})} \mathrm{d.3} \leq \mathrm{d_{max}} \\ \mathrm{d_{min}} : \mathit{Degree}(\mathrm{d_{ref}.2},\ \mathrm{d_{ref}.1}) \\ \prod_{\mathrm{d}:\mathit{Degree}(\mathrm{d_{ref}.2},\mathrm{d_{ref}.1})} \mathrm{d.3} \geq \mathrm{d_{min}.3} \end{array}\right] \\ \lambda\mathrm{p} : \mathrm{P}\ .\ \dfrac{\mathrm{d_{max}.3} + \mathrm{d_{min}.3}}{2} \end{array}\right)$$

As shown, ⟦half⟧ shares the first five of its presuppositions with ⟦completely⟧. It also incorporates two additional presuppositions:

(439) 6. there is some degree $\mathrm{d_{min}}$ (the minimum degree), on the same scale as $\mathrm{d_{ref}}$ and $\mathrm{d_{max}}$

7. $\mathrm{d_{min}}$ is minimal, in the sense that it is less than every other degree on the same scale

The assertion is that the magnitude of $\mathrm{d_{ref}}$ is exactly half way between the magnitude of $\mathrm{d_{max}}$ and the magnitude of $\mathrm{d_{min}}$. If the scale does not admit either a maximum or a minimum value, then the resulting update will be infelicitous. For instance, ?*half old* is unacceptable because there is no maximum degree, ?*half new* because there is no minimum degree, and ?*half hot* because there is neither a maximum nor a minimum.

The morphemes ⟦half⟧ and ⟦completely⟧ both act monotonically on arrows. Given an arrow α from an adjective G to an adjective H:

(440) $\alpha : \mathrm{G} \rightarrow_{\mathit{Gradable}} \mathrm{H}$

we can lift this to arrows:

(441) $[\![\mathrm{completely}]\!](\alpha) : \prod_{\mathrm{x}:\mathit{Ind}} [\![\mathrm{completely}]\!](\mathrm{G})(\mathrm{x}) \rightarrow_{\mathit{Update}} [\![\mathrm{completely}]\!](\mathrm{H})(\mathrm{x})$

$[\![\mathrm{half}]\!](\alpha) : \prod_{\mathrm{x}:\mathit{Ind}} [\![\mathrm{half}]\!](\mathrm{G})(\mathrm{x}) \rightarrow_{\mathit{Update}} [\![\mathrm{half}]\!](\mathrm{H})(\mathrm{x})$

This is implemented as follows for ⟦completely⟧:

(442) $$\begin{array}{l} [\![\mathrm{completely}]\!](\alpha) \coloneqq \lambda\mathrm{G} : \mathit{Gradable}\ . \\ \quad \lambda\mathrm{x} : \mathit{Ind}\ . \\ \quad\quad \text{let } \mathrm{G'} \coloneqq [\![\mathrm{completely}]\!](\mathrm{G})(\mathrm{x}) \text{ in} \\ \quad\quad \left(\begin{array}{l} \lambda\mathrm{p} : \mathrm{G'.1}\ .\ \left[\begin{array}{l} \alpha(\mathrm{x}).1(\mathrm{p.1}) \\ \mathrm{p.2} \\ \mathrm{p.3} \\ \mathrm{p.4} \\ \mathrm{p.5} \end{array}\right] \\ \lambda\mathrm{p} : \mathrm{G'.1}\ .\ \mathrm{id}_{\mathrm{G'.2(p)}} \end{array}\right) \end{array}$$

and likewise for $[\![\text{half}]\!]$:

(443)
$$[\![\text{half}]\!](\alpha) \coloneqq \lambda \mathrm{G} : \mathit{Gradable}\ .\ \lambda \mathrm{x} : \mathit{Ind}\ .\ \text{let } \mathrm{G}' \coloneqq [\![\text{half}]\!](\mathrm{G})(\mathrm{x}) \text{ in} \begin{pmatrix} \lambda \mathrm{p} : \mathrm{G}'.1\ .\ \begin{bmatrix} \alpha(\mathrm{x}).1(\mathrm{p}.1) \\ \mathrm{p}.2 \\ \mathrm{p}.3 \\ \mathrm{p}.4 \\ \mathrm{p}.5 \\ \mathrm{p}.6 \\ \mathrm{p}.7 \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{G}'.1\ .\ \mathrm{id}_{\mathrm{G}'.2(\mathrm{p})} \end{pmatrix}$$

4.7 Networks and discourse

4.7.1 The basic context update procedure

Context updates have featured heavily in this thesis, but I have not yet discussed how a context update is used to update a context, except for a brief description in Section 2.5.3. As mentioned in that section, my treatment of the context update procedure closely resembles discourse-based dynamic frameworks such as Discourse Representation Theory (Kamp 1980, Kamp et al. 2011) and the dynamic version of Type Theory with Records (Larsson 2015, Cooper 2016). Recall that a discourse context can be represented by a type, typically a large dependent sum type. For example, suppose one knows that there is some individual *John* who is 27 years old, and who has an elder sister. This can be represented as follows:

(444) $$C_t := \begin{bmatrix} \mathrm{J} : \mathit{Ind} \\ \mathrm{s} : \mathit{Ind} \\ \mathrm{John(J)} \\ \mathrm{old(J)} = 27 \text{ years} \\ \text{sister(s, J)} \\ \mathrm{old(s)} > \mathrm{old(J)} \end{bmatrix}$$

where 'old' is some measure function from individuals to degrees of age (for simplicity, I am omitting the fact that *old* carries presuppositions). The role of dependent sum types is similar to that of discourse representation structures in Discourse Representation Theory (DRT). Like DRT structures, dependent sums introduce variables which are related by various constraints, including identifications. The major difference is that types like (444) are not interpreted in some additional model-theoretic domain. Rather, they should be thought of as inherently model-theoretic, corresponding to something like sets of situations.

I shall now illustrate the most basic kind of update by means of a simplified example. Suppose that some speaker, Alice, represents the current discourse context as shown in (444), that is Alice knows that there is some individual called John, who is 27 years old and has an older sister. Alice's interlocutor, Bob, then announces "John's sister is married". Alice's first task is to derive the semantic interpretation of this sentence, which would look something like:

(445) $[\![\text{John's sister is married}]\!] : \mathit{Update}$

$$[\![\text{John's sister is married}]\!] = \begin{pmatrix} \mathrm{P} := \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{y} : \mathit{Ind} \\ \mathrm{John(x)} \\ \mathrm{sister(y, x)} \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \;.\; \mathrm{married(p.2)} \end{pmatrix}$$

As shown, the sentence presupposes that there is some individual called John who has a sister, and asserts that this sister is married. Alice then attempts to update her representation of the context C_t to incorporate this new information. For the update to be successful, its presuppositions must be supplied by the current context. In other words, there must be a function

(446) $s : C_t \to [\![\text{John's sister is married}]\!].1$

In this case, it is trivial to discover such a function:

(447) $$s := \lambda \mathrm{c} : \begin{bmatrix} \mathrm{J} : \mathit{Ind} \\ \mathrm{s} : \mathit{Ind} \\ \mathrm{John(J)} \\ \mathrm{old(J)} = 27 \text{ years} \\ \mathrm{sister(s, J)} \\ \mathrm{old(s)} > \mathrm{old(J)} \end{bmatrix} . \begin{pmatrix} \mathrm{c.1} \\ \mathrm{c.2} \\ \mathrm{c.3} \\ \mathrm{c.5} \end{pmatrix}$$

Having found a way in which the current context can supply the presuppositions, Alice then constructs the updated context, as follows:

(448) $$C_{t+1} = \begin{bmatrix} \mathrm{c} : C_t \\ [\![\text{John's sister is married}]\!].2(s(\mathrm{c})) \end{bmatrix}$$
$$= \begin{bmatrix} \mathrm{c} : \begin{bmatrix} \mathrm{J} : \mathit{Ind} \\ \mathrm{s} : \mathit{Ind} \\ \mathrm{John(J)} \\ \mathrm{old(J)} = 27 \text{ years} \\ \mathrm{sister(s, J)} \\ \mathrm{old(s)} > \mathrm{old(J)} \end{bmatrix} \\ \mathrm{married(c.2)} \end{bmatrix}$$

Notice how the updated context includes the previous context together with the new information added by the update. The new context C_{t+1} becomes the input to the next update event.

The example given in the previous paragraph was deliberately simplified by ensuring that all of the presuppositions were already satisfied in the context. In real dialogue, it is frequently the case that the current context does not by itself satisfy the presuppositions of the sentence and needs to be strengthened by the addition of extra information, a process known as *accommodation* (Karttunen 1974, Stalnaker 1974, Lewis 1979, Heim 1982). For example, suppose that instead of (444), Alice has the following take on the current context:

(449) $$C_t := \begin{bmatrix} \mathrm{J} : \mathit{Ind} \\ \mathrm{John(J)} \\ \mathrm{man(J)} \\ \mathrm{old(J)} = 27 \text{ years} \end{bmatrix}$$

That is, all Alice knows is that there is some individual called John who is a 27 year-old man. Then, faced with the sentence *John's sister is married*, Alice will be unable to supply the necessary presuppositions, since the context no longer contains John's sister. To fix this problem, Alice must strengthen the context by accommodating the required presuppositions. The first thing that Alice must discover is that the current context does satisfy *some* of the presuppositions, since it contains an individual named John. This common structure is represented by a type with functions from both C_t and $[\![$John's sister is married$]\!]$.1, as follows:

(450)
$$\begin{tikzcd}
 & \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{John(x)} \end{bmatrix} & \\
\begin{bmatrix} \mathrm{J} : \mathit{Ind} \\ \mathrm{John(J)} \\ \mathrm{man(J)} \\ \mathrm{old(J)} = 27 \text{ years} \end{bmatrix} \arrow[ur, "\sigma"] & & \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{y} : \mathit{Ind} \\ \mathrm{John(x)} \\ \mathrm{sister(y, x)} \end{bmatrix} \arrow[ul, "\tau"']
\end{tikzcd}$$

where σ and τ are the obvious functions which simply project out the relevant components. Alice then combines C_t and $[\![$John's sister is married$]\!]$.1, identifying their common structure, to give the accommodated context C_t':

(451)
$$\begin{tikzcd}
 & \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{John(x)} \end{bmatrix} & \\
\begin{bmatrix} \mathrm{J} : \mathit{Ind} \\ \mathrm{John(J)} \\ \mathrm{man(J)} \\ \mathrm{old(J)} = 27 \text{ years} \end{bmatrix} \arrow[ur, "\sigma"] & & \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{y} : \mathit{Ind} \\ \mathrm{John(x)} \\ \mathrm{sister(y, x)} \end{bmatrix} \arrow[ul, "\tau"'] \\
 & C_t' \coloneqq \begin{bmatrix} \mathrm{c} : \begin{bmatrix} \mathrm{J} : \mathit{Ind} \\ \mathrm{John(J)} \\ \mathrm{man(J)} \\ \mathrm{old(J)} = 27 \text{ years} \end{bmatrix} \\ \mathrm{p} : \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{y} : \mathit{Ind} \\ \mathrm{John(x)} \\ \mathrm{sister(y, x)} \end{bmatrix} \\ \sigma(\mathrm{c}) = \tau(\mathrm{p}) \end{bmatrix} \arrow[ul, ".1"] \arrow[ur, ".2"'] &
\end{tikzcd}$$

As shown, the accommodated context contains (1) the original context, (2) the required presuppositions, and (3) a constraint requiring that the common structure is identified. The equality constraint ensures that this is a commutative diagram: in other words, $.1 \circ \sigma = .2 \circ \tau$. This kind of construction, whereby two objects with maps to a common object are combined along those maps, is very common in mathematics, and is known as a *pullback*. Having formed the adjusted context C'_t, Alice must then check it for consistency, since there is no guarantee that combining the current context and the presuppositions in this way yields a consistent type. In this case, Alice discovers no inconsistencies, and the updated context is formed using the adjusted context, as follows:

(452) $$C_{t+1} = \begin{bmatrix} \text{c} : C'_t \\ [\![\text{John's sister is married}]\!].2(\text{c}.2) \end{bmatrix}$$

$$= \begin{bmatrix} \text{c} : \begin{bmatrix} \text{c} : \begin{bmatrix} \text{J} : \mathit{Ind} \\ \text{John(J)} \\ \text{man(J)} \\ \text{old(J)} = 27 \text{ years} \end{bmatrix} \\ \text{p} : \begin{bmatrix} \text{x} : \mathit{Ind} \\ \text{y} : \mathit{Ind} \\ \text{John(x)} \\ \text{sister(y, x)} \end{bmatrix} \\ \sigma(\text{c}) = \tau(\text{p}) \end{bmatrix} \\ \text{married(c.2.2)} \end{bmatrix}$$

Note that since the presuppositions are contained explicitly in the adjusted context, it is trivial to find a function from the adjusted context to the presuppositions, as needed for the assertion.

It is important to note that accommodation is not always possible. One way in which accommodation can fail is if the adjusted context is inconsistent. Consider the following example:

(453) John has never smoked. ?John quit smoking.

The first sentence yields a context in which there is no past event of John smoking, whereas the second sentence presupposes that John used to smoke. Accommodating the presuppositions of the second sentence to the context of the first sentence therefore results in an inconsistent adjusted context. An inconsistency can be detected automatically when a type contains a proposition P together with its negation ¬P. In some cases, multiple steps of reasoning might be required before an inconsistency can be automatically detected in this way. As part of the consistency test, the language user must therefore search for an inconsistency by enriching the context with additional information drawn from general knowledge. The problem of which knowledge should be employed in this search is notoriously difficult and beyond the scope of the present discussion.

Inconsistency aside, there are certain kinds of presuppositions, often referred to as *anaphoric* presuppositions (Kripke 2009), which do not permit accommodation. These include pronouns and expressions like *too*, *also* and *as well*. For example, suppose that instead of "John's sister is married", Bob simply says "She is married". This prompts Alice to construct roughly the following update:

(454) $[\![\text{she is married}]\!] : Update$

$$[\![\text{she is married}]\!] := \begin{pmatrix} \text{P} := \begin{bmatrix} \text{x} : Ind \\ \text{female(x)} \\ \text{person(x)} \end{bmatrix} \\ \lambda\text{p} : \text{P} \;.\; \text{married(p.1)} \end{pmatrix}$$

As shown, the utterance presupposes the presence of some female person in the context, and asserts of this individual that she is pregnant. In this case, if no female person is present in the context, then Alice immediately takes Bob's utterance to be infelicitous: it is not possible to introduce a referent for *she* in the same way as *John's sister*. It is therefore necessary to distinguish anaphoric presuppositions from ordinary, non-anaphoric presuppositions, which can be readily accommodated.

4.7.2 Adding polysemy: the weakest-first strategy

The update procedure described in the previous subsection assumed only a single context update. However, a typical sentence yields not just a single update, but an entire network of updates, connected by arrows. We therefore need to combine the context update procedure with the network approach to polysemy. In doing so, we immediately run into what I referred to in Chapter 2 as the exponential growth problem – the fact that the number of potential interpretations for an expression grows exponentially in the number of polysemous words. For instance, a sentence with 10 polysemous words, each of which has 5 possible senses, may have up to $5^{10} = 9765625$ possible interpretations. The interpreter therefore cannot efficiently search the space of all possible interpretations: the only viable strategy is to construct a small set of hypotheses, which are then tested against the context and adjusted if necessary.

In accordance with the Strongest Meaning Hypothesis (Dalrymple et al. 1994, Winter 2001*b*), the interpreter's goal is to construct the *strongest possible assertion whose presuppositions are satisfied in the context*. This can be achieved by iteratively building senses, beginning with those whose presuppositions are most general and proceeding gradually to more specific presuppositions. It might seem strange that, in searching for the *strongest* possible assertion, one should begin with the *weakest* possible presuppositions. However, this is in fact the only strategy which will allow the interpreter to efficiently search the space of possible interpretations. To see this, imagine that a sentence has 22 interpretations, which are structured into the following network:

(455)
$$\begin{tikzcd}
 & B \arrow[r] \arrow[dr] & F \arrow[r] \arrow[dr] & J \arrow[r] \arrow[dr] & N \arrow[r] \arrow[dr] & R \arrow[ddr] & \\
 & C \arrow[ur] \arrow[ddr] & G \arrow[ur] \arrow[ddr] & \mathbf{K} \arrow[ur] \arrow[ddr] & O \arrow[ur] \arrow[ddr] & S \arrow[dr] & \\
A \arrow[uur] \arrow[ur] \arrow[dr] \arrow[ddr] & & & & & & V \\
 & D \arrow[uur] \arrow[dr] & H \arrow[uur] \arrow[dr] & L \arrow[uur] \arrow[dr] & P \arrow[uur] \arrow[dr] & T \arrow[ur] & \\
 & E \arrow[ur] \arrow[r] & I \arrow[ur] \arrow[r] & M \arrow[ur] \arrow[r] & Q \arrow[ur] \arrow[r] & U \arrow[uur] &
\end{tikzcd}$$

Suppose that the strongest assertion compatible with the context happens to be sense K, shown in bold. Consider two interpretation strategies: a strongest-first strategy and a weakest-first strategy. The strongest-first strategy begins with the most specific sense and gradually weakens it in a breadth-first manner, until it discovers a sense which can be satisfied by the current context. Knowing that a sense X is infelicitous tells you nothing about whether some sense downstream of X will be felicitous, so the strongest-first strategy is forced to construct all senses up to and including the target sense. Before converging on the solution, it builds the following structure:

(456)
$$\begin{tikzcd}
 & B \arrow[r] \arrow[dr] & F \arrow[r] \arrow[dr] & J \\
 & C \arrow[ur] \arrow[ddr] & G \arrow[ur] \arrow[ddr] & \mathbf{K} \\
A \arrow[uur] \arrow[ur] \arrow[dr] \arrow[ddr] & & & \\
 & D \arrow[uur] \arrow[dr] & H \arrow[uur] \arrow[dr] & L \\
 & E \arrow[ur] \arrow[r] & I \arrow[ur] \arrow[r] & M
\end{tikzcd}$$

The weakest-first strategy, on the other hand, begins with the least specific sense and gradually strengthens it in a breadth-first manner, until it encounters senses which *cannot* be satisfied by the context. When it encounters a sense X which is infelicitous, it automatically rules out all senses upstream of X. Assuming that only senses downstream of K are felicitous, the weakest-first strategy builds the following structure before converging on the solution:

(457)
$$\begin{tikzcd}
 & N \arrow[r] \arrow[dr] & R \arrow[ddr] & \\
\mathbf{K} \arrow[ur] \arrow[ddr] & & S \arrow[dr] & \\
 & & & V \\
 & P \arrow[uur] \arrow[dr] & T \arrow[ur] & \\
 & & U \arrow[uur] &
\end{tikzcd}$$

As this example illustrates, the weakest-first strategy is much more efficient, considering only 8 senses, instead of 13. This is due to the fact that one can ignore all senses upstream of a sense which is not felicitous. By the time one gets to the fourth-weakest layer, there is only one remaining candidate, which is the target sense K. The number of senses which the weakest-first strategy must construct and test is dependent on the number of senses which are felicitous in the context. For example, suppose that the senses T, O and Q are all felicitous. Then the following structure would need to be built before converging on K:

(458)
$$\begin{tikzcd}
 & N \arrow[r] \arrow[ddr] & R \arrow[dddr] & \\
 & & & \\
\mathbf{K} \arrow[uur] \arrow[ddr] & O \arrow[uur] \arrow[ddr] & S \arrow[dr] & \\
 & & & V \\
L \arrow[uur] \arrow[ddr] & P \arrow[uur] \arrow[ddr] & T \arrow[ur] & \\
 & & & \\
 & Q \arrow[uur] \arrow[r] & U \arrow[uuur] &
\end{tikzcd}$$

There are cases in which the number of senses required by the weakest-first strategy is greater than the number required by the strongest-first strategy. However, the weakest-first strategy remains more efficient in general since it makes better use of the implicational structure.

Another huge advantage of the weakest-first strategy is that the way in which weaker senses are satisfied by the context can guide the search for how stronger senses are satisfied by the context. Imagine a sentence with two interpretations $A, B : \mathit{Update}$ connected by an arrow $\alpha : A \rightarrow_{\mathit{Update}} B$. Suppose the interpreter knows that the presuppositions of the weaker sense B can be supplied by the current context via a function $\sigma : C_t \rightarrow B.1$. The interpreter is interested in whether the presuppositions of the stronger sense can be supplied by the context. It is helpful to assume that the way in which the context satisfies the stronger sense is compatible with the way in which it satisfies the weaker sense. In other words, the interpreter should search for a function σ' which makes the following diagram commute:

(459)
$$\begin{tikzcd}
\mathrm{A.1} \arrow[r, "\alpha.1"] \arrow[dr, phantom, "\checkmark"] & \mathrm{B.1} \\
\mathrm{C}_t \arrow[u, "\sigma'"] \arrow[ur, "\sigma"'] &
\end{tikzcd}$$

where σ is the function the interpreter has already discovered and $\alpha.1$ is given by the arrow α connecting A and B. Rather than starting from scratch, all the interpreter needs to figure out is how to supply the extra information which is present in $A.1$ but not in $B.1$. This further illustrates the logic of searching senses from weakest to strongest instead of strongest to weakest, since the latter strategy does not allow partial results to constrain the search process.

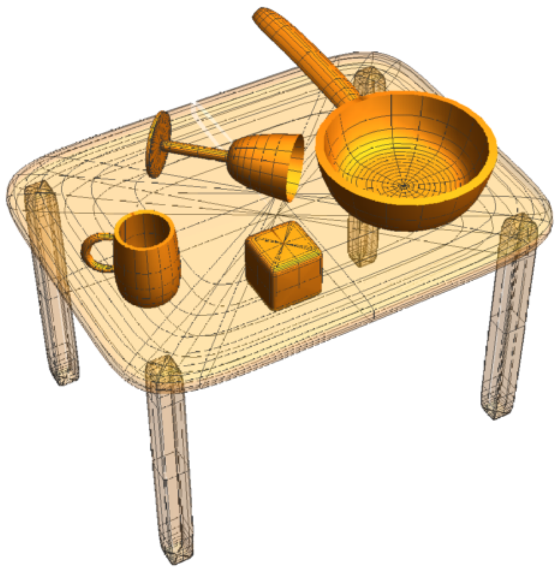
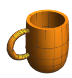
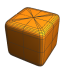
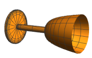
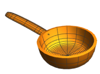


Figure 4.7: An illustration of the current context.

4.7.3 A worked example

I shall now go through a detailed example of how a speaker might use a network to guide the interpretation of a sentence with respect to a context, using the weakest-first strategy. Consider the scene shown in Figure 4.7. It consists of a small table, on which stand several objects: a mug, a cube, a toppled wine glass and a frying pan. Suppose that an agent, Alice, is observing the scene from the perspective shown in the figure, and is attending to the objects on the table, rather than the table itself. We can assume that Alice has access to a perceptual representation of the shape of each object. Let us suppose that she is not concerned with categorizing the objects, but conceptualizes them only as shapes – rather than as a mug, a cube, a wine glass, and a frying pan. We might represent Alice's take on the context as follows:

(460) $C_t \coloneqq$

$\mathrm{o}_1 : \mathit{Ind}$
$\mathrm{o}_2 : \mathit{Ind}$
$\mathrm{o}_3 : \mathit{Ind}$
$\mathrm{o}_4 : \mathit{Ind}$
has-shape(o_1, )
has-shape(o_2, )
has-shape(o_3, )
has-shape(o_4, )

That is, Alice understands the context to contain four individuals, o_1, o_2, o_3, o_4, each of which has a particular shape. The use of images is intended to leave open the question of exactly how shape–related information is encoded.

Now suppose that Alice has a conversation partner, Bob, who is also observing the scene in Figure 4.7. Bob wishes to draw Alice's attention to a certain aspect of the scene, and utters the following sentence:

(461) The wine glass is 18cm tall.

Alice's aim is to find the strongest assertion which is compatible with the current context, which in this case is $[\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{top,1st,=}}$. Along the way, she must identify which object Bob means by 'the wine glass' and discover proofs of predicate types such as $\textsc{top}(\mathrm{x},\mathrm{v})$ and $\textsc{1st}(\mathrm{x},\mathrm{v})$ which are not immediately available in the context.

Alice begins by constructing the most general interpretation of Bob's utterance. This is the interpretation which is derived from the weakest sense of *tall*, $[\![\text{tall}]\!]_{\text{vert,sgfnt}}$, and the weakest sense of MEAS, $[\![\textsc{meas}]\!]_{\geq}$:

(462) $[\![\text{the wine glass is } \textsc{meas} \text{ 18cm tall}]\!]_{\text{vert,sgfnt,}\geq}$

$$= [\![\text{the}]\!]([\![\text{wine glass}]\!])([\![\textsc{meas}]\!]_{\geq}([\![\text{tall}]\!]_{\text{vert,sgfnt}})([\![\text{18cm}]\!]))$$

$$= \left(\begin{array}{l} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ [\![\text{wine glass}]\!](\mathrm{x}) \\ \begin{bmatrix} \mathrm{v} : \mathit{Vector} \\ (\textsc{axis}(\mathrm{x},\mathrm{v}) \times \textsc{up}(\mathrm{v})) + \textsc{top}(\mathrm{x},\mathrm{v}) \\ |\textsc{1st}(\mathrm{x},\mathrm{v}) + \textsc{large}(\mathrm{v})| \end{bmatrix} \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \,.\, \|\mathrm{p}.3.1\| \geq 18\text{cm} \end{array} \right)$$

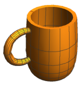
As shown, this update presupposes that there is some individual x, that x is a wine glass, and that x satisfies the presuppositions of $[\![\text{tall}]\!]_{\text{vert, sgfnt}}$, namely having an axis which points up, either in the environment or in the object's canonical orientation, and which is either primary or large in comparison to a human being. Given a context satisfying these conditions, the assertion is that the magnitude of the wine glass' vertical axis is greater than or equal to 18cm.

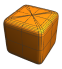
Having formed this context update, Alice must then try to use her representation of the current context to satisfy its presuppositions. That is, she searches for a function:

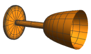
(463) $\sigma : C_t \rightarrow [\![\text{the wine glass is } \textsc{meas} \text{ 18cm tall}]\!]_{\text{vert,sgfnt,}\geq}.1$

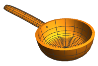
Discovering this function requires some general knowledge, such as knowledge about the typical shape and orientation of wine glasses, as well as some spatial reasoning, such as the capacity to extract a primary axis from a 3D model. For our purposes, it is sufficient to write σ in an abstract form as follows:

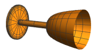
(464) $\sigma := \lambda c : \begin{bmatrix} o_1 : \mathit{Ind} \\ o_2 : \mathit{Ind} \\ o_3 : \mathit{Ind} \\ o_4 : \mathit{Ind} \end{bmatrix}$

has-shape(o_1, )

has-shape(o_2, )

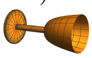
has-shape(o_3, )

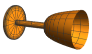
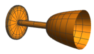
has-shape(o_4, )

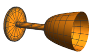
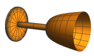
$.\ \Big(c.3,\ \sigma_{\text{wine\_glass}}(c.7),$ (axes()[1], $\text{inr}(\sigma_{\text{top}}(c.7))$, $|\text{inl}(\sigma_{\text{1st}}(c.7))|$) $\Big)$

where:

axes : $\mathit{Shape} \to \text{List}(\mathit{Vector})$

$\sigma_{\text{wine\_glass}}$: has-shape(o_3, ) $\to$ $[\![\text{wine glass}]\!](o_3)$

σ_{top} : has-shape(o_3, ) $\to$ $\textsc{top}(o_3$, axes()[1])

σ_{1st} : has-shape(o_3, ) $\to$ $\textsc{1st}(o_3$, axes()[1])

As indicated, Alice uses her representation of the shape of the individual o_3 to produce proofs that it is a wine glass, and that its longest axis is inherently vertical and primary. She concludes that the sense $[\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq}$ is felicitous.

Having found that the most general sense can be supplied by the context, Alice now attempts to strengthen the interpretation, constructing all senses which can be weakened to $[\![\text{the wine glass is } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq}$ in a single step:

(465)
$$[\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{up,sgfnt},\geq} \to [\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq}$$
$$[\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,1st},\geq} \to [\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq}$$
$$[\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,large},\geq} \to [\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq}$$
$$[\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},=} \to [\![\text{the wine glass is 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq}$$

Note that the first three candidates are derived by strengthening the interpretation of *tall* by one step, whereas the last candidate is derived by strengthening the interpretation of MEAS. Alice cannot know ahead of time which kind of strengthening will lead to the strongest possible assertion. It might seem like this step would take four times longer than the previous step, since there are now four senses to check against the context. However, as discussed in the previous subsection, the way in which the context satisfies the presuppositions of a stronger sense is assumed to be compatible with the way in which it satisfies the presuppositions of the weaker sense. For each of the stronger senses S' in (465), Alice searches for an arrow σ' which makes the following diagram commute:

(466)
$$\begin{tikzcd} S'.1 \arrow[rr] & & \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\text{vert,sgfnt},\geq}.1 \\ & \checkmark & \\ & C_t \arrow[uul, "\sigma'"] \arrow[uur, "\sigma"'] & \end{tikzcd}$$

where σ is the function she has already discovered. It is easy to see that only two of the stronger senses admit such a commuting diagram: $\llbracket$the wine glass is 18cm MEAS tall$\rrbracket_{\text{vert,1st},\geq}$ and $\llbracket$the wine glass is 18cm MEAS tall$\rrbracket_{\text{vert,sgfnt},=}$. Of the other two senses, $\llbracket$the wine glass is 18cm MEAS tall$\rrbracket_{\text{up,sgfnt},\geq}$ is ruled out because the wine glass is not standing upright, and $\llbracket$the wine glass is 18cm MEAS tall$\rrbracket_{\text{vert,large},\geq}$ is ruled out because the glass is not large in comparison to a human being.

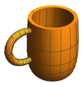
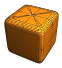
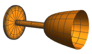
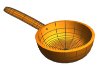
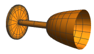
Having found two stronger candidates which are felicitous, Alice proceeds to construct all senses which can be weakened to one of these candidates in a single step, *which do not also have an arrow into an infelicitous sense.* Only one such sense exists, namely $\llbracket$the wine glass is 18cm MEAS tall$\rrbracket_{\text{vert,1st},=}$:

(467)
$$\begin{tikzcd} & \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\substack{\text{up}\\ \text{sgfnt}\\ \geq}} \arrow[ddr] & \\ & \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\substack{\text{vert}\\ \text{1st}\\ \geq}} \arrow[dr] & \\ \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\substack{\text{vert}\\ \text{1st}\\ =}} \arrow[ur] \arrow[ddr] & & \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\substack{\text{vert}\\ \text{sgfnt}\\ \geq}} \\ & \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\substack{\text{vert}\\ \text{large}\\ \geq}} \arrow[ur] & \\ & \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\substack{\text{vert}\\ \text{sgfnt}\\ =}} \arrow[uur] & \end{tikzcd}$$

As before, Alice must find a map from the context to the new candidate which respects the way in which she has satisfied the presuppositions of weaker candidates. In other words, she must find an arrow σ'' which makes the following diagram commute:

(468)
$$\begin{tikzcd} \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\text{vert,1st},=}.1 \arrow[rr] & & \left\llbracket\begin{array}{l}\text{the wine glass is}\\ \text{18cm }\textsc{meas}\text{ tall}\end{array}\right\rrbracket_{\text{vert,sgfnt},\geq}.1 \\ & \checkmark & \\ & C_t \arrow[uul, "\sigma''"] \arrow[uur, "\sigma"'] & \end{tikzcd}$$

where σ is the original supply function and the top arrow is the unique horizontal arrow from $[\![$the wine glass is 18cm MEAS tall$]\!]_{\text{vert,1st,=}}.1$ to $[\![$the wine glass is 18cm MEAS tall$]\!]_{\text{vert,sgfnt,}\geq}.2$ given by the interpretation network. The function σ'' is implemented as follows:

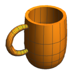
(469) $\sigma'' := \lambda \mathrm{c} :$ [
$\mathrm{o}_1 : \mathit{Ind}$
$\mathrm{o}_2 : \mathit{Ind}$
$\mathrm{o}_3 : \mathit{Ind}$
$\mathrm{o}_4 : \mathit{Ind}$
has-shape(o_1, )
has-shape(o_2, )
has-shape(o_3, )
has-shape(o_4, )
] . (
c.3
$\sigma_{\text{wine\_glass}}(\mathrm{c}.7)$
(axes()[1]
inr($\sigma_{\text{top}}(\mathrm{c}.7)$)
$\sigma_{\text{1st}}(\mathrm{c}.7)$))

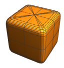
which is identical to σ except that there is no need to inject the proof of $\textsc{1st}(\mathrm{x}, \mathrm{v})$ into a sum type.

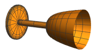
Having discovered that $[\![$the wine glass is 18cm MEAS tall$]\!]_{\text{vert,1st,=}}$ is a viable sense, Alice should once more attempt to strengthen the interpretation by one step, constructing all senses with an arrow into a felicitous sense which do not have an arrow into an infelicitous sense. At this point, however, there are no such senses, because every sense with an arrow into $[\![$the wine glass is 18cm MEAS tall$]\!]_{\text{vert,1st,=}}$ also has an arrow into one of the senses which Alice has already ruled out. She therefore concludes that $[\![$the wine glass is 18cm MEAS tall$]\!]_{\text{vert,1st,=}}$ is the strongest possible assertion which is compatible with the context. She uses the arrow $\sigma'' : C_t \to [\![$the wine glass is 18cm MEAS tall$]\!]_{\text{vert,1st,=}}.1$ to construct the updated context in the usual way:

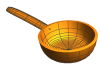
(470) $C_{t+1} = \begin{bmatrix} \mathrm{c} : C_t \\ [\![\text{the wine glass is 18cm MEAS tall}]\!]_{\text{vert,1st,=}}.2(\sigma''(\mathrm{c})) \end{bmatrix}$

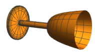
$=$ $\left[\text{c} : \left[\begin{array}{l} \text{o}_1 : \mathit{Ind} \\ \text{o}_2 : \mathit{Ind} \\ \text{o}_3 : \mathit{Ind} \\ \text{o}_4 : \mathit{Ind} \end{array} \right. \right.$

has-shape(o_1, )

has-shape(o_2, )

has-shape(o_3, )

has-shape(o_4, ) $\Big]$

‖axes()[1]‖ = 18cm]

As shown, the new context contains the previous context C_t, together with the assertion that the magnitude of the wine glass' primary axis is strictly equal to 18cm. Alice is now in a position to judge the truth or falsity of the sentence with respect to the context. She takes the sentence to be true iff she can find a proof of its assertion type, and false iff she can find a proof of its negation.

The above example involved a network of context updates whose presupposition and assertion arrows go in the same direction. However, there are also 'skew' networks whose presupposition and assertion arrows go in opposite directions: as discussed previously, this pattern is associated with negation and words which involve negation such as *most* and *least*. For networks of this type, the interpretive strategy is different. For example, suppose that Alice represents the current context as before, but Bob instead utters the sentence:

(471) The wine glass is *not* 18cm tall.

As before, Alice's goal is to find the strongest assertion which is compatible with the context, in accordance with the Strongest Meaning Hypothesis. She begins in the same way by constructing the sense with the most general presuppositions:

(472) $[\![\text{the wine glass is not 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq}$

$$= [\![\text{not}]\!]([\![\text{the wine glass is } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq})$$

$$= \left(\begin{array}{l} \text{P} := \left[\begin{array}{l} \text{x} : \mathit{Ind} \\ [\![\text{wine glass}]\!](\text{x}) \\ \left[\begin{array}{l} \text{v} : \mathit{Vector} \\ (\textsc{axis}(\text{x}, \text{v}) \times \textsc{up}(\text{v})) + \textsc{top}(\text{x}, \text{v}) \\ |1\textsc{st}(\text{x}, \text{v}) + \textsc{large}(\text{v})| \end{array} \right] \end{array} \right] \\ \lambda \text{p} : \text{P} \;.\; \neg(\|\text{p}.3.1\| \geq 18\text{cm}) \end{array} \right)$$

Note that the presuppositions are the same as those of $[\![$the wine glass is 18cm $\textsc{meas}$ tall$]\!]_{\text{vert,sgfnt},\geq}$, but the assertion has been negated. As before, Alice discovers the function σ shown in (464). Now, however, due to the skew-monotone action of $[\![\text{not}]\!]$, the sense with the weakest presuppositions also has the strongest assertion. Alice can therefore terminate without considering senses with stronger presuppositions. She forms the updated context:

(473) $$C_{t+1} = \begin{bmatrix} \mathrm{c} : C_t \\ [\![\text{the wine glass is not 18cm } \textsc{meas} \text{ tall}]\!]_{\text{vert,sgfnt},\geq}.2(\sigma(\mathrm{c})) \end{bmatrix}$$

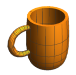
$= \Big[$ c : $\Big[$

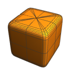
$\mathrm{o}_1 : \mathit{Ind}$

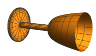
$\mathrm{o}_2 : \mathit{Ind}$

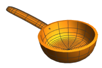
$\mathrm{o}_3 : \mathit{Ind}$

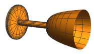
$\mathrm{o}_4 : \mathit{Ind}$

has-shape(o_1, )

has-shape(o_2, )

has-shape(o_3, )

has-shape(o_4, ) $\Big]$

$\neg$(‖axes()[1]‖ $\geq$ 18cm) $\Big]$

As shown, Alice takes Bob to mean that the central axis of the wine glass is not greater than or equal to 18cm in length. Notice that Alice's interpretation of the positive sentence *the wine glass is 18cm tall* involves the exact equality sense of MEAS, whereas her interpretation of the negative sentence *the wine glass isn't 18cm tall* involves the 'at least' sense of MEAS. This is a consequence of the assumption that the speaker makes the strongest possible assertion licensed by the context.

My goal in this section has been to suggest how an interpreter can take advantage of network structure in order to find the optimal interpretation of a sentence with respect to a context. The interpreter cannot consider every possible interpretation of a sentence, because the number of potential senses grows exponentially in the number of words. Instead, the interpreter should begin with the most general presuppositions and work their way backwards through the network, constructing and testing senses in a breadth-first manner in order to find the strongest assertion which is licensed by the context. Many of the details remain to be worked out, including the interaction between network structure and accommodation, and the distinction between anaphoric and non-anaphoric presuppositions. I leave a full integration of implicational polysemy and dynamic semantics as a goal for future research.

Chapter 5

Summary and Conclusion

Contents

5.1 Summary of chapters

This thesis has illustrated an approach to formal semantics which takes seriously the observation from cognitive linguistics that different senses of a word can be organised into a network (e.g. Brugman & Lakoff 1988, Tyler & Evans 2001). I began by distinguishing between implicational networks, which represent micro-senses of the same semantic type, and derivational networks, which represent macro-senses of potentially different semantic types. When one implicational network acts on another in a composition $N(M)$, the senses multiply, giving us one sense for every pair $n(m)$, where n is a sense in N and m is a sense in M. I have argued that composition should not 'throw away' information about how the senses in N and M are connected together. Rather, this information should be preserved if possible, and lifted to the level of entire phrases and sentences. The result is a framework with two kinds of composition – the 'vertical' composition of arrows within a network, and the 'horizontal' composition of senses across networks.

In Chapter 2, I argued that a traditional Montague-style semantics based on Simple Type Theory is not well-suited to implementing the preservation of implication under composition. This is because Simple Type Theory (STT) does not treat proofs as first-class objects which can be taken and returned by functions, so networks can only be described indirectly, for example using meaning postulates. Instead, I suggested replacing STT with Dependent Type Theory (DTT), a more elaborate type system originally proposed as a foundation for mathematics (Martin-Löf 1984). DTT subsumes both the role of set theory and the role of logic, due to its inherent theory of truth and entailment. Proofs become first-class objects, allowing for a direct implementation of networks and network-based composition. DTT has already been advocated as a framework for semantics for a number of independent reasons, such as its straightforward treatment of Donkey anaphora (Ranta 1994), selectional restrictions (Luo 2010) and presupposition (Tanaka et al. 2017).

Chapter 3 was a lexical analysis of the English spatial adjectives, drawing on previous cognitive approaches. This chapter is best seen as a synthesis of three ideas: the lexical network approach to polysemy, Kennedy's (1999) analysis of gradable adjectives as measure functions, and Zwarts' (2000) use of vectors to describe spatial language. Each spatial adjective sense is analysed as a measure function which takes an individual and presupposes a relation between that individual and a certain configuration of vectors; the function itself then returns the length of one of the vectors in the configuration. The presuppositions are built from a set of primitives such as AXIS, UP, PATH, OBS, and so on, which are hypothesized to be universal across different languages. An arrow between two gradable adjectives is a map of their presupposition types which commutes with the action of the measure function. Unlike most previous approaches, my networks contain not only prototypes (meets, points of divergence), but also generalizations (joins, points of convergence). Two senses share a common prototype if they are compatible and typically occur together; two senses share a common generalization if they can be coordinated without giving rise to zeugma.

Having investigated the lexical semantics of spatial adjectives, Chapter 4 then turned to their compositional semantics. The aim of this chapter was to describe the semantics of degree morphemes such as MEAS, *more/er*, *less*, *most*, *least*, POS, and so on, drawing on the work of previous authors in the degree semantics tradition (e.g. Klein 1980, von Stechow 1984, Heim 1985, Kennedy 1999, Kennedy & McNally 2005). What distinguishes my approach from previous work is: (1) its close attention to the presuppositional content of degree morphemes, and (2) the idea that degree morphemes are capable of acting not only on individual adjective senses, but also on the arrows which connect senses together. The degree morphemes covered in this chapter included MEAS (measure phrase combination), *more/-er*, *less*, *most*, *least*, POS (the positive), *very*, *completely* and *half.* The chapter concluded with a brief description of how, if a sentence has a network of interpretations, this can be used to search for the best interpretation relative to a context.

The structure of this final chapter is as follows. Section 5.2 summarizes what I take to be the main theoretical contributions of the thesis, defending them against some potential objections. Section 5.3 compares my approach to a number of similar approaches in the literature which might be thought to describe something like lexical networks, with the aim of bringing out points of agreement and disagreement. Finally, Section 5.4 provides some very brief remarks on how this thesis relates to more general questions in the study of language and mind.

5.2 Novel contributions

5.2.1 Implication networks

In the literature on lexical networks, there is no consensus about what constitutes a node and what constitutes an arrow. For some, nodes are frames and arrows are frame relations such as ‘frame addition’ and ‘profile shift’ (e.g. Norvig & Lakoff 1987); for others nodes are image schemas and arrows are image schema transformations (e.g. Tyler & Evans 2001); for others, nodes are combinations of features and arrows correspond to dropping features (e.g. Lakoff’s 1987 analysis of *mother*). Moreover, the structure of networks varies widely, from trees in which everything diverges from a common node (Tyler & Evans 2001), to partial orders (Zwarts 2004), to undirected graphs (Vandeloise 1993). The large variety of lexical networks on offer makes it difficult to give a general characterisation of lexical network theory and how networks compose.

In this thesis, I have focused on a particular category of lexical networks: implication networks. These are networks in which all of the nodes belong to the same semantic type: for instance, intersective adjectives, gradable adjectives, transitive verbs, generalized quantifiers, etc. The different nodes of an implication network correspond to what we might call *microsenses*: fine-grained distinctions between different uses of a word which do not have morphosyntactic consequences. As shown in Chapter 3, spatial adjectives provide a rich source of examples – for instance, the distinction between the secondary sense of *wide* (e.g. *wide strip*) and the lateral sense (e.g. *wide piano*) is a distinction between two microsenses. The arrows in an implication network represent implicational relationships between microsenses. For instance, the secondary + horizontal sense of *wide* (e.g. *wide road*) implies the secondary sense.

Implication networks are *partial orders*, meaning that (a) there can be at most one arrow between any two nodes, (b) arrows compose transitively, and (c) no two distinct nodes can form a loop. As partial orders, implication networks can contain *both meets and joins*. This distinguishes them from most networks one finds in the cognitive literature, which contain only meets. I have argued that a meet between two senses is justified when they are prototypically found in combination, whereas a join is justified when they can be coordinated non-zeugmatically. For example, compare the *climb* network described in Chapter 2 with a portion of the *high* network described in Chapter 3:

(474) a.
$$\begin{tikzcd} & \llbracket \text{climb} \rrbracket_{\text{rise}} \\ \llbracket \text{climb} \rrbracket_{\text{rise,clamber}} \arrow[ur] \arrow[dr] & \\ & \llbracket \text{climb} \rrbracket_{\text{clamber}} \end{tikzcd}$$

b.

$$\begin{tikzcd}[column sep=small] \llbracket \text{high} \rrbracket_{\text{posn}} \arrow[dr] & \\ & \llbracket \text{high} \rrbracket_{\text{elev}} \\ \llbracket \text{high} \rrbracket_{\text{dim,grnd}} \arrow[ur] & \end{tikzcd}$$

The structure of the *climb* network reflects the fact that rising and clambering are typically found in combination. At the same time, it also correctly predicts that $\llbracket \text{climb} \rrbracket_{\text{rise}}$ and $\llbracket \text{climb} \rrbracket_{\text{clamb}}$ cannot be sensibly coordinated (e.g. ?*the plane climbed into the air and the monkey down the tree*) due to the lack of a common join. On the other hand, the structure of the *high* network predicts that position above ground and extension above ground can be sensibly coordinated (e.g. *the airplane is higher than the building*), but do not typically occur in combination, due to the lack of a common meet. (In fact they are inconsistent with each other.)

One might object to the need for a network analysis in order to express the behaviour of words like *climb* and *high*. Why not simply define $\llbracket \text{climb} \rrbracket$ as 'rise *or* clamber'? As Jackendoff (2002) has pointed out, the definition of $\llbracket \text{climb} \rrbracket$ as 'rise or clamber' does not capture the appropriate typicality effects because a logical disjunction is not 'more prototypically satisfied' when both disjuncts are true compared to only one. Conversely, defining $\llbracket \text{high} \rrbracket$ as 'position or extension above ground' does not capture the fact that a comparison involving a position and an extension (*the plane is higher than the building*) is less typical than a comparison involving two positions (*the plane is higher than the cloud*) or two extensions (*building A is higher than building B*). Moreover, describing both $\llbracket \text{climb} \rrbracket$ and $\llbracket \text{high} \rrbracket$ in terms of disjunction does not explain why rising and clambering are zeugmatic, whereas vertical position and extension are not.

The concept of an implication network is not entirely new to this thesis. Rosch's *family resemblance categories*, Lakoff's *radial categories*, and Jackendoff's *cluster concepts* all refer to a similar idea. One could also consider taxonomies and inheritance hierarchies of the sort found in feature-based grammars to be kinds of implication network. The major contribution of this thesis is the formalization of implication networks using Dependent Type Theory (DTT). Because DTT encodes proofs directly as functions, one can write expressions not only for senses but also for arrows. Distinct semantic types give rise to distinct kinds of implication network: one can have networks of nouns, intersective adjectives, transitive verbs, intransitive verbs, prepositions, generalized quantifiers, tense morphemes, and so on. Moreover, each type of network has an associated (parametric) arrow type. For example, given two noun senses $N, M : \mathit{Ind} \to \mathit{Type}$, an arrow connecting them is an element of:

(475) $\prod_{\mathrm{x}:\mathit{Ind}} N(\mathrm{x}) \to M(\mathrm{x})$

whereas given two generalized quantifiers $Q, R : (\mathit{Ind} \to \mathit{Type}) \to (\mathit{Ind} \to \mathit{Update}) \to \mathit{Update}$, an arrow connecting them is an element of:

(476) $\prod_{\mathrm{N}:\mathit{Ind}\to\mathit{Type}} \prod_{\mathrm{P}:\mathit{Ind}\to\mathit{Update}} Q(\mathrm{N})(\mathrm{P}) \to_{\mathit{Update}} R(\mathrm{N})(\mathrm{P})$

5.2.2 Preservation of implication

One of the central claims of this thesis is that the preservation of implication is an automatic consequence of semantics, rather than, say, dependent on proof by in background inference engine. Tracking entailment relations between interpretations is important for cooperative communication because of the assumption of speaker informativeness (e.g. Grice 1975, Horn 1984, Levinson et al. 2000, Chierchia 2004). Given a sentence with multiple possible interpretations, a hearer assumes (a) that the speaker intends the strongest possible assertion which is compatible with the context, and (b) that stronger sentences which the speaker might have uttered but did not are unassertable. Both (a) and (b) require the speaker to use implication relations at the word level to derive implication relations at the sentence level. In traditional semantics of the Montague variety, this requires inference, which is slow in comparison to semantic composition. A great deal of reasoning is required just to establish all the necessary entailments, before one even gets to the kinds of inferences described in pragmatics. In semantics based on DTT, however, implication preservation can be completely automated, removing the need for the intermediate step.

In a theory where preservation of implication is automatic, every function capable of preserving implication must have both an action on senses and an action on arrows. In practical terms, this means that the lexical entries of many words need to be altered so as to include not only the original semantic interpretation (suitably rendered in DTT), but also an action on arrows. One might wonder whether such lifting rules are really necessary, given knowledge of network structure. What need is there to lift arrows when we already know the relative informativeness of individual senses? Given a composition $N(M)$, why not take the 'most informative' interpretation to be given by $n_{proto}(m_{proto})$, where n_{proto} is the most informative sense in N and m_{proto} is the most informative sense in M? The problem is that this does not always work. To reiterate an example discussed in Chapter 1, the strongest interpretation of *meat* refers to the flesh of a land animal, typically a mammal, whereas the strongest interpretation of *John does not eat meat* is that John does not eat animals of any kind, including fish. This is explained by the fact that *not* lifts assertion maps anti-monotonically.

The observation that some contexts preserve implication in a monotone direction, others in an anti-monotone direction, and others not at all, is usually associated with the distribution of polarity items. It is a well-known observation that negative polarity items such as *anything*, *ever*, *any*, *at all*, and so on, are licensed in anti-monotone contexts (Fauconnier 1975, Ladusaw 1979). This is often discussed in connection with generalized quantifiers: for instance, *everyone* is monotone in its first argument but anti-monotone in its second argument, so (a) is acceptable but (b) is not:

(477) a. Everyone who has ever been there remembers something.
b. Everyone who has *never been there remembers *anything.

In the framework developed in this thesis, the concept of monotonicity is extended far beyond generalized quantifiers to encompass many additional semantic types, including degree morphology, tense/aspect morphology, sentential modifiers, and so on: anything which can act on a lexical network. For instance, we saw in Section 4.4 that the degree morpheme *most* is anti-monotone in its gradable adjective and comparison class arguments. This is reflected in the distribution of polarity items:

(478) John is the tallest man *everywhere/anywhere.

In a compositional theory of lexical networks, one can easily check whether a lexical item is monotone, anti-monotone or non-monotone in a particular argument position simply by inspecting its action on arrows in that position. This is in contrast to Montague semantics, where the monotonicity of a lexical item cannot be easily determined by inspection and must be hard-coded, for example using meaning postulates or syntactic markers.

Another reason why it is important to lift implications from the level of words to the level of entire sentences has to do with what I have called the exponential growth problem – the observation that the number of potential interpretations of a sentence grows exponentially with the number of words. If senses form an unstructured set, then there is no way to effectively screen out interpretations which are inconsistent or incompatible with the background context without simply checking them one-by-one. However, if senses are partially ordered by strength, then one can take advantage of this structure to efficiently identify the strongest consistent sense. As explained in Section 4.7, this is done by searching through the network in a breadth-first manner, beginning with the weakest or most general senses and gradually strengthening them. When a sense is found to be inconsistent/infelicitous, the interpreter removes it from consideration, along with all the senses with arrows into it. This has the effect of dramatically reducing the number of senses which the interpreter needs to consider.

5.2.3 Networks and presupposition

An additional contribution which I take to be novel is the idea that lexical networks relate not only the assertive content of words but also their presuppositional content. Following the tradition of dynamic semantics (e.g. Kamp 1980, Heim 1983, Seuren 1994), I describe the meaning of a sentence as an instruction for updating a context with new information. Presuppositions appear as constraints on the current context which must be satisfied in order for the update to take place (this is similar to the 'two-step' version of Discourse Representation Theory described in Kamp et al. 2011, and also the implementation of dynamic semantics in Type Theory with Records). A context update is described by an element of the following dependent sum type:

(479) $Update \coloneqq \begin{bmatrix} \mathrm{P} : Type \\ \mathrm{P} \to Prop \end{bmatrix}$

where the first component encodes the presuppositions and the second component takes a proof of the presuppositions and yields an assertion.

I have argued that there are two kinds of update networks: parallel networks and skew networks. In a parallel update network, an arrow from an update U to an update V consists of an arrow from the presuppositions and assertion of U to the presuppositions and assertion of V:

(480) $U \to_{Update} V \coloneqq \begin{bmatrix} \mathrm{f} : U.1 \to V.1 \\ \prod_{\mathrm{p}:U.1} U.2(\mathrm{p}) \to V.2(\mathrm{f}(\mathrm{p})) \end{bmatrix}$

In a skew network, on the other hand, an arrow from U to V maps presuppositions and assertions in the opposite order:

(481) $U \rightleftarrows_{Update} V \coloneqq \begin{bmatrix} \mathrm{f} : U.1 \to V.1 \\ \prod_{\mathrm{p}:U.1} V.2(\mathrm{f}(\mathrm{p})) \to U.2(\mathrm{p}) \end{bmatrix}$

Skew networks are associated with negative contexts because negation lifts presupposition arrows monotonically but assertion arrows anti-monotonically. For instance, consider the various interpretations of

(482) The vase is not 10cm deep.

The stronger the interpretation of *deep*, the stronger the presuppositions: the sense $[\![\text{deep}]\!]_{\text{intrnl,down}}$ presupposes that the vase is canonically oriented, whereas $[\![\text{deep}]\!]_{\text{intrnl}}$ does not. However, a stronger interpretation of *deep* also results in a weaker assertion because of the presence of negation: *not 10cm deep* in the $[\![\text{deep}]\!]_{\text{intrnl}}$ sense entails *not 10cm deep* in the $[\![\text{deep}]\!]_{\text{intrnl,down}}$ sense. This is characteristic of a skew network.

Unlike other authors who work in DTT (e.g. Luo 2010, Chatzikyriakidis & Luo 2017*a*), I analyse the selectional restrictions associated with adjectives and verbs as presuppositions rather than domain restrictions on predicates. For example, rather than each intersective adjective having a unique type:

(483) $A \to Prop$

where A is the domain of the adjective (e.g. $[\![\text{red}]\!] : \textit{Physical Object} \to Prop$, $[\![\text{happy}]\!] : Animal \to Prop$, and so on), I instead group all intersective adjectives together as elements of the type:

(484) $Ind \to Update$

that is, functions from individuals to sentence meanings. The selectional restrictions of the adjective are then encoded as presuppositions on the resulting update. For instance, *red* would presuppose that the subject is capable of having a colour, *happy* that the subject is capable of having emotions, and so on.

The advantage of this approach is that it permits type uniformity for syntactically-relevant semantic categories – intersective adjectives, intransitive verbs, adverbs, and so on – whilst also capturing their distinct selectional restrictions. Type uniformity is important because one often needs to write functions which can take elements with distinct selectional restrictions. For example, degree morphemes like *more/-er*, *most*, POS, etc., take an arbitrary gradable adjective. One can get around this problem to some extent by introducing *universes* (types of types). For example, Luo (2010) introduces a universe he calls CN ('common noun'), which contains all noun denotations together with all possible selectional restrictions on adjectives and verbs. CN is then used parametrically to write functions capable of taking elements with distinct selectional restrictions. However, I have chosen not to follow Luo's analysis of selectional restrictions for a number of reasons, which are explained in greater detail in Section 5.3.1 below.

One consequence of my approach to intersective adjectives is that adjective + noun composition becomes a kind of local context update of the sort described in dynamic semantics, where the noun plays the role of the discourse context and the adjective plays the role of the context update. To combine a noun $N : Ind \to Type$ and an intersective adjective $A : Ind \to Update$, one must first find a function $\sigma : \prod_{\mathrm{x}:Ind} N(\mathrm{x}) \to A(x).1$, from the noun contents to the adjective presuppositions. Assuming such a function can be found, one then forms the updated noun contents as follows:

(485) $AN \coloneqq \lambda \mathrm{x} : Ind \ . \begin{bmatrix} \mathrm{c} : N(\mathrm{x}) \\ A(\mathrm{x}).2(\sigma(\mathrm{x})(\mathrm{c})) \end{bmatrix}$

which is identical to the procedure for updating a discourse context, except for the lambda abstraction over individuals. If no function σ can be found, then one attempts to form an adjusted noun meaning by accommodating the adjective presuppositions, following the accommodation procedure described in Chapter 4. As far as I know, this is the first analysis of adjective + noun composition to emphasize its connections to dynamic semantics.

5.2.4 Contributions to gradable adjectives

Following Kennedy (1999, 2005, 2007), I analyse gradable adjectives as measure functions from individuals to degrees. This differs from more traditional degree-based approaches, which describe gradable adjectives as relations between individuals and degrees (e.g. Cresswell 1976, Bierwisch 1989, Heim 2000). It can also be contrasted with a vague predicate approach (e.g. Kamp 1975, Klein 1980, 1982), where gradable adjectives are analysed as predicates with a truth-value gap. The novel aspect of my approach is that it takes advantage of the resources of DTT in order to (a) express the structure of scales, (b) represent the presuppositions associated with gradable adjectives whilst preserving type uniformity, and (c) connect different senses of gradable adjectives into lexical networks. The resulting framework is well suited to describing both the lexical structure of gradable adjectives and their compositional behaviour.

DTT is ideal for expressing the organisation of degrees into ordered scales. One begins by introducing the type *Scale*, consisting of labels for scales (dist, weight, temp, etc.), and the type *Polarity*, consisting of the two values $+, -$. For every scale s and polarity p, one introduces a distinct set of degrees $Degree(s, p)$ together with a relation $\geq_{(s,p)}$: $Degree(s, p) \times Degree(s, p) \to Prop$ satisfying the axioms for a total order. It follows that degrees on different scales or with different polarities are disjoint and cannot be compared. This contrasts with approaches in which all degrees belong to the same underlying space (e.g. Faller 2000, Winter 2005, Kennedy & McNally 2005), where there is no reason in principle why degrees on different scales must be disjoint. The relationship between positive and negative degrees on the same scale is expressed by introducing a bijective order-reversing function $\mathrm{negate}_{\mathrm{s}} : Degree(s, +) \to Degree(s, -)$ for every scale s.

The distinction between positive and negative degrees supports a distinction between positive and negative gradable adjectives, which are defined as follows:

(486)
- $$\mathit{Gradable}(+) \coloneqq \lambda \mathrm{x} : \mathit{Ind} \,.\, \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \begin{bmatrix} \mathrm{s} : \mathit{Scale} \\ \mathit{Degree}(\mathrm{s}, +) \end{bmatrix} \end{bmatrix}$$
- $$\mathit{Gradable}(-) \coloneqq \lambda \mathrm{x} : \mathit{Ind} \,.\, \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \begin{bmatrix} \mathrm{s} : \mathit{Scale} \\ \mathit{Degree}(\mathrm{s}, -) \end{bmatrix} \end{bmatrix}$$

This distinction is syntactically relevant: for example only positive polarity adjectives can occur unmarked with an accompanying measure phrase (*15cm tall*, ?*15cm short*). As before, DTT makes it possible to retain type uniformity whilst expressing differences in selectional restrictions, which appear as components of the presupposition type P. This is not possible in Simple Type Theory, where all gradable adjectives would need to belong to a distinct type as a consequence of their distinct selectional restrictions. To write expressions for degree morphemes like *more/-er*, it is also necessary to have a type for all gradable adjectives, whether positive or negative, which is defined as follows:

(487) $$\mathit{Gradable} \coloneqq \lambda \mathrm{x} : \mathit{Ind} \,.\, \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \begin{bmatrix} \mathrm{s} : \mathit{Scale} \\ \mathrm{p} : \mathit{Pol} \\ \mathit{Degree}(\mathrm{s}, \mathrm{p}) \end{bmatrix} \end{bmatrix}$$

Any element of *Gradable*$(+)$ or *Gradable*$(-)$ can be considered as an element of *Gradable* in a canonical way.

I have argued that, just like other lexical items, gradable adjectives consist of multiple senses organised into an implicational network. Given two gradable adjectives $G, H : \mathit{Gradable}$, an arrow connecting them is a function f from the presuppositions of G to the presuppositions of H, together with a proof that f commutes with the two measure functions:

(488) $$G \to_{\mathit{Gradable}} H \;\coloneqq\; \textstyle\prod_{\mathrm{x}:\mathit{Ind}} \begin{bmatrix} \mathrm{f} : G.1(\mathrm{x}) \to H.1(\mathrm{x}) \\ \prod_{\mathrm{p}:G.1} G.2(\mathrm{p}) = H.2(\mathrm{f}(\mathrm{p})) \end{bmatrix}$$

Degree morphemes can act on both gradable adjective senses and gradable adjective arrows, lifting them to the level of entire sentences. In Chapter 4, I showed that the degree morphemes MEAS, *more/-er*, *less*, *completely* and *half* are monotone in the usual way, lifting an arrow of type $\to_{\mathit{Gradable}}$ to an arrow of type $\to_{\mathit{Update}}$. The superlative morphemes *most* and *least* act monotonically on presuppositions but anti-monotonically on assertions, lifting an arrow of type $\to_{\mathit{Gradable}}$ to an arrow of type $\rightleftarrows_{Update}$. Finally, the morphemes POS and *very* are non-monotone: they have no action on arrows.

One respect in which my analysis of degree morphology differs from that of other authors, besides the preservation of implication, is its close attention to presuppositional information and the various ways in which degree morphemes can be infelicitous. The vast majority of the information associated with degree morphemes is presupposed rather than asserted. For example, a superlative statement like *John is the tallest man in the room* presupposes that: (i) John is a man, (ii) a man is the kind of thing which can be in a room, (iii) John is in the room, (iv) John satisfies the presuppositions of *tall*, and (v) there is at least one other man in the room besides John who satisfies the presuppositions of *tall*. Each one of these presuppositions could potentially be false, in which case the sentence is infelicitous. All of this information needs to be included in the superlative morpheme alongside its assertive content.

5.2.5 Contributions to spatial language

Chapter 3 was a decompositional analysis of the English spatial adjectives, focusing on the spatial configurations which they presuppose. Following Zwarts & Winter (2000), I take the basic domain underlying spatial language to be spatial vectors. A spatial vector is an arrow in 3D Euclidean space with a direction and a magnitude, but no fixed location. They are introduced as fundamental geometric entities, not based on numerical constructions such as tuples of real numbers. Spatial vectors can be used to represent both the position of an object relative to some landmark and the magnitude and direction of an object's axes. Continuous functions from $[0, 1] \rightarrow Vector$ can be thought of as paths in space, and continuous functions from $[0, 1] \times [0, 1] \rightarrow Vector$ can be thought of as surfaces. The concepts of position, axis, path and surface are fundamental to all forms of spatial language, including locative and directional prepositions, axial part terms, spatial adjectives, shape-based numeral classifiers and shape-based classificatory verbs.

I have proposed a collection of basic relations between individuals and configurations of vectors. These include concepts such as $1\text{ST}(x, v)$ 'v is a primary axis of x', $\text{INTRNL}(x, v)$ 'v is an internal axis of x', and $\text{POSN}(x, u, v)$ 'v is the position of x with respect to u'. These primitives are intended to be universal across different languages, whereas the particular way in which they are combined into lexical networks can vary from language to language. For example, English *tall* always refers to a vertical axis, whereas Italian *alto* 'high/tall' can refer either to a vertical axis or a vertical position. Many of these primitives are not my own invention, but are drawn from other studies on spatial language, notably the work of Bierwisch (1967) and Lang (1989, 2001) on spatial adjectives, and the work of Herskovits (1987) and Zwarts (2000, 2005) on spatial prepositions. The role of these basic predicates is similar to that of *image schemas* or *spatial scenes* in cognitive approaches (e.g. Johnson 1987, Tyler & Evans 2003).

A novel suggestion from Chapter 3 is the analysis of the area and arc senses of width in terms of *bivectors*. A bivector is an oriented area, just as a vector is an oriented length. Bivectors can be constructed from vectors through the introduction of the exterior product, which also gives rise to oriented volume elements known as trivectors. Examples of bivector quantities include the area swept out by a curve or the spatial extent of a field, whereas examples of trivector quantities include the size of a house or the volume of a cavity. I have argued that bivectors are involved in area-based senses of *wide* such as *wide bend*, *wide gesture*, *wide field*, *wide landscape*, and so on. Since it has units of area, the magnitude of a bivector cannot be described by a distance-denoting measure phrase, e.g. ?*10cm wide bend*, ?*1m wide gesture*, ?*5 mile wide landscape*. I would suggest that bivectors and trivectors are also involved in other aspects of spatial language. For instance, bivectors can be used to describe rotations, and might therefore be involved in verbs of rotational motion, e.g. *swing*, *circle*, *spiral*, *turn*, many of which can also combine adverbially with *wide*.

Chapter 3 proposed a lexical network model for each of the English spatial adjectives, *high*, *tall*, *long*, *wide*, *thick*, *deep*, and their antonyms. Although many of my ideas are based on the work of cognitive authors – particularly Vandeloise (1988, 1993), Dirven & Taylor (1986) and Vogel (2004) – each of the networks in Chapter 3 is original with this thesis and represents a novel contribution to the study of spatial adjectives. Each network is a micro-theory which makes a range of predictions about speaker intuitions, such as whether sense A is more typical than sense B, or whether A and B are zeugmatic or non-zeugmatic. A network is successful insofar as it correctly predicts speaker judgements. Given this, the networks in Chapter 3 should be regarded not as definitive solutions, but as working models which might need to be altered in light of additional data.

5.3 Comparison with other approaches

5.3.1 Dot types and copredication

Dot types are a construction introduced by Pustejovsky (1995) which have since been taken up by other authors, notably Asher (2011) and Luo (2012). Although more restricted in their applications, dot types are similar in spirit to the framework proposed in this thesis, since they yield a network of senses which is preserved under composition. The principle motivation for dot types is to explain *copredication*, a situation in which an individual appears to belong to more than one type simultaneously, permitting the application of predicates with seemingly disjoint selectional requirements. An example is given in (489):

(489) The book about the history of Philosophy is too heavy to carry.

The predicate *about the history of Philosophy* requires an informational subject, like a website or a lecture, whereas the predicate *too heavy to carry* requires a physical object, like a rock or a tree. A book, it seems, can be understood in two ways: on the one hand as an informational object, and on the other hand as a physical object, permitting the application of both predicates.

The approach to dot types which I shall describe is based on Luo (2012), who also works within a DTT framework. (As explained in Section 2.5.2, the major difference between Luo's approach and my own is that he analyses common nouns as denoting types rather than predicates.) Luo begins by introducing a binary relation between types $A <_c B$, meaning 'A is a subtype of B, by virtue of the unique coercion $c : A \to B$'. To give an example, one might consider the type of dogs to be a subtype of the type of animals, $Dog <_i Animal$, by virtue of the unique coercion $i : Dog \to Animal$. The coercion allows any dog to be automatically considered as an animal, and therefore passed to a function which accepts animals. This is formalized by the following rule:

(490) $$\frac{\Gamma \vdash f : B \to C \qquad \Gamma \vdash a : A \qquad \Gamma \vdash A <_c B}{\Gamma \vdash f(a) \equiv f(c(a)) : C}$$

In other words, given a unique coercion $A <_c B$, any element of A can be automatically passed to a function $B \to C$ by using c to 'fill in the gap'. Additional rules are required to ensure that the subtype relation is transitive and antisymmetric: see Luo (1999) for a complete formulation.

Dot types encode the intuitive idea that a multifaceted entity like a book should belong to a complex semantic type which combines all of its aspects. One begins with types for different classes of entities, like PHYSICAL OBJECT, INFORMATION, EVENT, INSTITUTION, and so on. These are used to encode the selectional restrictions of predicates: for instance, the predicate *too heavy to carry* has type PHYSICAL OBJECT $\to$ *Prop*, whereas the predicate *about the History of philosophy* has type INFORMATION $\to$ *Prop*. One then introduces the dot type constructor $\bullet$, to form types for compound entities. Given two types A and B, the type $A \bullet B$ consists of entities with both an A aspect and a B aspect. For example, a book would belong to type PHYSICAL OBJECT $\bullet$ INFORMATION, having both a physical aspect and an informational aspect. Luo's rules for dot types are given below:

(491) Formation rule: $$\frac{\Gamma \vdash A : \mathit{Type} \quad \Gamma \vdash B : \mathit{Type} \quad \mathcal{C}(A) \cap \mathcal{C}(B) = \emptyset}{\Gamma \vdash A \bullet B : \mathit{Type}}$$

Introduction rule: $$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash \langle a, b \rangle : A \bullet B}$$

Elimination rules: $$\frac{\Gamma \vdash c : A \bullet B}{\Gamma \vdash \pi_1(c) : A} \qquad \frac{\Gamma \vdash c : A \bullet B}{\Gamma \vdash \pi_2(c) : B}$$

Computation rules: $$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash \pi_1(\langle a, b \rangle) \equiv a : A} \qquad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash \pi_2(\langle a, b \rangle) \equiv b : B}$$

Projections as coercions: $$\frac{\Gamma \vdash A \bullet B : \mathit{Type}}{\Gamma \vdash A \bullet B <_{\pi_1} A} \qquad \frac{\Gamma \vdash A \bullet B : \mathit{Type}}{\Gamma \vdash A \bullet B <_{\pi_2} B}$$

Notice that the dot type construction is almost the same as the product $A \times B$, the major difference being the extra condition $\mathcal{C}(A) \cap \mathcal{C}(B) = \emptyset$ in the Formation Rule. The extra condition forbids A and B from sharing any common components in the case where they are themselves dot types. For example, the type:

(492) EVENT $\bullet$ (INFORMATION $\bullet$ EVENT)

is not well-formed, because both sides share the common component EVENT. It is this condition which permits the two projections π_1 and π_2 to be unique coercions. To see why this is the case, notice that the type in (492), if it was well-formed, would not have a unique coercion to EVENT.

Dot types in combination with coercive subtyping yield a straightforward analysis of copredication. Suppose that the predicates *about the history of Philosophy* and *too heavy to carry* are typed as follows:

(493) ⟦about the history of Philosophy⟧ : INFORMATION $\rightarrow$ *Prop*

⟦too heavy to carry⟧ : PHYSICAL OBJECT $\rightarrow$ *Prop*

Moreover, suppose that the interpretation of *book* is a subtype of PHYSICAL OBJECT$\bullet$ INFORMATION (recall that for Luo, common nouns are interpreted as types):

(494) ⟦book⟧ $<$ PHYSICAL OBJECT $\bullet$ INFORMATION

Since PHYSICAL OBJECT $\bullet$ INFORMATION can be coerced to PHYSICAL OBJECT on the one hand and INFORMATION on the other, it follows that both predicates can be applied to a term of type ⟦book⟧:

(495) ⟦about the history of Philosophy⟧

: INFORMATION $\rightarrow$ *Prop*

$<$ PHYSICAL OBJECT $\bullet$ INFORMATION $\rightarrow$ *Prop*

$<$ ⟦book⟧ $\rightarrow$ *Prop*

$$\begin{aligned}
&[\![\text{too heavy to carry}]\!]\\
&\quad : \textsc{physical object} \to \mathit{Prop}\\
&\quad < \textsc{physical object} \bullet \textsc{information} \to \mathit{Prop}\\
&\quad < [\![\text{book}]\!] \to \mathit{Prop}
\end{aligned}$$

The sentence in (489) can therefore be successfully interpreted.

The dot type construction might be considered to give rise to a kind of lexical network consisting of a prototype and a collection of peripheral senses. A prototypical book consists of both a physical aspect and an informational aspect, but it is also possible for a book to be purely physical, as in a physical volume containing no writing, or purely informational, as in a book stored in a computer. We can therefore introduce three senses of *book*:

(496) $[\![\text{book}]\!]_{\text{proto}} < \textsc{physical object} \bullet \textsc{information}$

$[\![\text{book}]\!]_{\text{phys}} < \textsc{physical object}$

$[\![\text{book}]\!]_{\text{info}} < \textsc{information}$

Where $[\![\text{book}]\!]_{\text{phys}}$ lacks the informational component and $[\![\text{book}]\!]_{\text{info}}$ lacks the physical component. The arrows $[\![\text{book}]\!]_{\text{proto}} \to [\![\text{book}]\!]_{\text{phys}}$ and $[\![\text{book}]\!]_{\text{proto}} \to [\![\text{book}]\!]_{\text{info}}$ are given by lifting the projections $\pi_1 : \textsc{physical object} \bullet \textsc{information} \to \textsc{physical object}$ and $\pi_2 : \textsc{physical object} \bullet \textsc{information} \to \textsc{information}$, as follows:

(497)
$$\begin{tikzcd}
 & {[\![\text{book}]\!]_{\text{phys}}} \arrow[rr, "<"] & & \textsc{physical object} \\
{[\![\text{book}]\!]_{\text{proto}}} \arrow[ur, "\pi_1'"] \arrow[dr, "\pi_2'"'] \arrow[rr, "<"] & & \begin{array}{l}\textsc{physical object}\\ \bullet\ \textsc{information}\end{array} \arrow[ur, "\pi_1"] \arrow[dr, "\pi_2"'] & \\
 & {[\![\text{book}]\!]_{\text{info}}} \arrow[rr, "<"] & & \textsc{information}
\end{tikzcd}$$

The result is a radial network of the familiar V-shaped kind. We can get radial networks with a more complex structure by iterating the dot type construction. For instance, the prototypical interpretation of *conference* is plausibly a subtype of EVENT • INFORMATION • INSTITUTION. Assuming that none of these features is necessary, but any combination is sufficient, we could derive 3 senses containing two components (EVENT • INFORMATION, INFORMATION • INSTITUTION and EVENT • INSTITUTION) and 3 senses containing one component (EVENT, INFORMATION and INSTITUTION), with arrows given by lifting the appropriate projection functions. Since dot types are product-like, they provide only outbound arrows. In order to represent convergent networks, they would need to be combined with some version of sum types.

Although they can be used to express networks for multifaceted nouns, dot types are unsuited to representing larger networks, such as those which connect the interpretations of sentences. The reason has to do with the condition that the two sides of a dot type do not share any components, which can cause problems for larger networks in which senses inevitably contain multiple entries of the same type. For example, every sense of *the wine glass is deeper than the frying pan* presupposes the existence of two individuals – *the wine glass* and *the frying pan*, and so could not be represented using a dot type. Even single words can contain entries with the same type: for instance, the position sense of height we saw in Section 3.4.2 involves two vectors – one describing a point on the ground and another describing the object's elevation. Because of the widespread need for multiple entries of the same type, it follows that dot types are not well suited to representing arbitrary lexical networks.

Moreover, as discussed briefly in Section 2.5.4, there are good reasons to question one of the fundamental assumptions behind the dot type construction, which is that coercions should always be unique. In general, the way in which an argument may satisfy the selectional restrictions of a predicate is not always unique. To give an example, the expression *silver tree* might be taken to refer to the colour of the tree's leaves, or to the colour of its bark. In other words, there are at least two different coercions from the type *Tree* to the domain type of the predicate *silver*. The phenomenon of multiple coercion is also prevalent in spatial adjectives: for instance *thick spoon* might refer to the thickness of the spoon's handle, the thickness of the bowl, or both. This is why I have not adopted coercive subtyping in this thesis, but instead allow for the possibility of multiple coercion functions.

Given that dot types cannot serve as a replacement to dependent sum types for representing lexical networks, the question remains whether they need to be introduced at all. Provided one is willing to abandon unique implicit coercions, the phenomenon dot types are intended to explain, namely copredication, can be handled by the existing machinery of dependent sum types. Let us return to the description of nouns as elements of $Ind \to Type$, that is functions from individuals to contexts. The most prototypical sense of the noun *book* might be written as follows:

(498) $$[\![\mathrm{book}]\!]_{\mathrm{proto}} \coloneqq \lambda \mathrm{x} : Ind \, . \begin{bmatrix} \mathrm{p} : Physical \\ \mathrm{i} : Information \\ \mathrm{book(x, p, i)} \end{bmatrix}$$

where $Information$ is a type containing informational properties, $Physical$ is a type containing physical properties, and the predicate $\mathrm{book(x, i, p)}$ means something like 'x is a book with physical structure p and informational structure i'.

Now suppose we have two predicates, *about the history of philosophy* and *too heavy to carry*, which in my system are both elements of $Ind \to Update$, that is functions from individuals to sentence meanings. We might write them, somewhat simplistically, as shown below:

(499) $$[\![\text{about the history of Philosophy}]\!] \coloneqq \lambda \mathrm{x} : Ind \, . \begin{bmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{i} : Information \\ \text{has-info(x, i)} \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \, . \\ \quad \text{about-history-of-philosophy(p.1)} \end{bmatrix}$$

$$[\![\text{too heavy to carry}]\!] \coloneqq \lambda x : \mathit{Ind} \; . \begin{bmatrix} P \coloneqq \begin{bmatrix} w : \mathit{Weight} \\ \text{has-weight}(x, w) \end{bmatrix} \\ \lambda p : P \; . \; \text{too-heavy-to-carry}(p.1) \end{bmatrix}$$

For the present discussion the important content is the selectional restrictions. As shown, $[\![\text{about the history of Philosophy}]\!]$ presupposes that its subject has some information associated with it, whereas $[\![\text{too heavy to carry}]\!]$ presupposes that its subject has some associated weight. Recall that in adjective + noun composition, the noun acts as a context which must supply the presuppositions of the adjective. In order to combine *book* with the predicates *about the history of philosophy* and *too heavy to carry*, the interpreter must discover two coercion functions:

(500) $\sigma_1 : \prod_{x:\mathit{Ind}} [\![\text{book}]\!]_{\text{proto}}(x) \to [\![\text{about the history of Philosophy}]\!](x).1$

$\sigma_2 : \prod_{x:\mathit{Ind}} [\![\text{book}]\!]_{\text{proto}}(x) \to [\![\text{too heavy to carry}]\!](x).1$

Without going into details, σ_1 would use the proof of $\text{book}(x, p, i)$ to derive a proof of $\text{has-info}(x, i)$, whereas σ_2 would extract a weight w and use the proof of $\text{book}(x, p, i)$ to derive a proof of $\text{has-weight}(x, w)$. Assuming that these coercions exist, the sentence in (489) would then be interpreted as follows:

(501) $[\![\text{The book about the history of Philosophy is too heavy to carry}]\!]_{\text{proto}}$

$= [\![\text{the}]\!]([\![\text{book about the history of Philosophy}]\!]_{\text{proto}})([\![\text{too heavy to carry}]\!])$

$$= \begin{bmatrix} P \coloneqq \begin{bmatrix} x : \mathit{Ind} \\ c : \begin{bmatrix} p : \mathit{Physical} \\ i : \mathit{Information} \\ \text{book}(x, p, i) \end{bmatrix} \\ \text{about-history-of-philosophy}(c.2) \end{bmatrix} \\ \lambda p : P \; . \; \text{too-heavy-to-carry}(p.2.1) \end{bmatrix}$$

Notice how *about the history of philosophy* and *too heavy to carry* select different parts of the context provided by *book*: the former selects the physical component, whereas the latter selects the informational component. The two major differences with the dot type approach are (1) nouns are treated as predicates rather than types, and (2) the interpreter must discover the coercion functions σ_1 and σ_2, rather than these being automatically provided. Nevertheless, both accounts agree that copredication is a result of multifaceted entities which combine components from different domains, e.g. physical properties and information.

5.3.2 Default logic and enthymemes

Another approach which might be seen as implementing a kind of lexical network is the use of non-monotonic rules of inference of the sort which are described by default logic (Reiter 1980). Non-monotonic inference refers to the idea that conclusions might be retracted in the light of new information. This kind of inference is not possible in ordinary monotonic logic, where strengthening one's assumptions can only lead to stronger conclusions, but is common in real life situations. To give an example, if told that x is a snake, one is likely to conclude that x lives on land, suggesting a rule of the following kind:

(502) $\forall x \,.\, \text{snake}(x) \Rightarrow \text{land-animal}(x)$

However, given the additional information that x is a sea snake, this conclusion must be revised, due to the presence of the following two rules:

(503) a. $\forall x \,.\, \text{sea-snake}(x) \Rightarrow \text{snake}(x)$
b. $\forall x \,.\, \text{sea-snake}(x) \Rightarrow \neg\text{land-animal}(x)$

In classical logic, the rules in (502) and (503) taken together would lead to a contradiction. The only way to avoid this would be to reformulate (502) as follows:

(504) $\forall x \,.\, (\text{snake}(x) \wedge \neg\text{sea-snake}(x)) \Rightarrow \text{land-animal}(x)$

In other words, all snakes are land animals, *except sea snakes*. The idea behind default logic is that, given that most generalizations have exceptions, and that we may not know all of the exceptions to any given rule, there ought to be a way to formalise general rules without having to 'hard-code' all their exceptions as in (504). This is done by means of default rules, which are often written as follows:

(505) $$\frac{\text{snake}(x) : \text{land-animal}(x)}{\text{land-animal}(x)}$$

This should be read as "if x is a snake and there is nothing to contradict that x is a land animal, then x is a land animal". The rule is non-monotonic because if we know only that x is a snake then we should conclude that x is a land animal; but if we know that x is a snake and a sea snake, then we should not draw this conclusion, because being a sea snake is incompatible with being a land animal. A logic in which rules like (505) are permitted was first described by Reiter (1980).

Default rules might be seen as describing something like an implication network. To illustrate, suppose that, in addition to the rule in (505), we also know some other facts concerning snakes: that they typically have scales and lay eggs. For all we know, there may be exceptions, so these generalizations are represented as default rules:

(506) $$\frac{\text{snake}(x) : \text{has-scales}(x)}{\text{has-scales}(x)}$$
$$\frac{\text{snake}(x) : \text{lays-eggs}(x)}{\text{lays-eggs}(x)}$$

Now consider the set of all things which can be called snakes. Some will satisfy all three of our default properties, others will satisfy only some, and yet others may satisfy none. We can use our default rules to construct a poset of types of snakes, where the most specific type contains snakes which satisfy all the defaults, and the most general type contains all snakes, regardless of how many of the defaults they satisfy: this is illustrated in Figure 5.1.

$$
\begin{tikzcd}
& \begin{bmatrix}\text{x} : \mathit{Ind}\\ \text{snake(x)}\end{bmatrix} & \\
\begin{bmatrix}\text{x} : \mathit{Ind}\\ \text{snake(x)}\\ \text{land-animal(x)}\end{bmatrix} \arrow[ur] & \begin{bmatrix}\text{x} : \mathit{Ind}\\ \text{snake(x)}\\ \text{has-scales(x)}\end{bmatrix} \arrow[u] & \begin{bmatrix}\text{x} : \mathit{Ind}\\ \text{snake(x)}\\ \text{lays-eggs(x)}\end{bmatrix} \arrow[ul] \\
\begin{bmatrix}\text{x} : \mathit{Ind}\\ \text{snake(x)}\\ \text{land-animal(x)}\\ \text{has-scales(x)}\end{bmatrix} \arrow[u] \arrow[ur] & \begin{bmatrix}\text{x} : \mathit{Ind}\\ \text{snake(x)}\\ \text{land-animal(x)}\\ \text{lays-eggs(x)}\end{bmatrix} \arrow[ul] \arrow[ur] & \begin{bmatrix}\text{x} : \mathit{Ind}\\ \text{snake(x)}\\ \text{has-scales(x)}\\ \text{lays-eggs(x)}\end{bmatrix} \arrow[ul] \arrow[u] \\
& \begin{bmatrix}\text{x} : \mathit{Ind}\\ \text{snake(x)}\\ \text{land-animal(x)}\\ \text{has-scales(x)}\\ \text{lays-eggs(x)}\end{bmatrix} \arrow[ul] \arrow[u] \arrow[ur] &
\end{tikzcd}
$$

Figure 5.1: A poset of different types of snakes according to whether they satisfy certain stereotypes.

It is important to emphasize that the status of this kind of network is different from that of the other networks discussed in this thesis. Although it might justifiably be called an implication network, since its arrows correspond to implications, it is not really a *lexical* network in the sense that it describes the polysemy of an individual lexical item. Whilst it may be useful for organising information about snakes, it tells us little about what it means to *be* a snake. Contrast this to the networks given in Chapter 3, which attempt to decompose concepts like *high*, *tall*, *long*, *wide*, and so on, in terms of simpler predicates. Default rules are not decompositional, because in order to state a rule like 'most snakes have scales' or 'most snakes are land animals', one needs a pre-existing notion of 'snake'. It is impossible to use default rules to express features which are diagnostic of being a snake – such as being alive or having a snake-like shape – without assuming the very concept one is trying to describe.

This is not to say that default rules have no role in a decompositional theory. Rather, we should distinguish between *prototypical* features which confer membership in a category, and *stereotypical* features which are merely associated with a category. To use an example from Coleman & Kay (1981), since surgery is a male-dominated profession, maleness might be considered a stereotypical property of surgeons. Nevertheless, it should not be considered part of the category SURGEON since, “a female surgeon is in no way a marginal or defective member of the category” (ibid.). Social changes which bring about a more balanced ratio of male to female surgeons will not alter the meaning of the word *surgeon*; rather, they will alter people’s attitudes regarding surgeons. Compare this to the property of having medical knowledge, or habitually performing surgery, which are both inherent, prototypical features of surgeons. The proper role of default rules is to describe stereotypes such as ‘surgeons are male’, not to decompose a category like SURGEON into simpler features. Likewise, default rules involving snakes express properties stereotypically associated with snakes, but cannot be used to define SNAKE itself.

Whilst the stereotypical features of a category are not part of its inherent decomposition, they do play an important role in how it is used and understood. For instance, consider the following dialogue, based on an example by Reiter:

(507) A: Is John a pacifist?
B: He’s a Quaker.

A understands B’s utterance as affirming that John is a pacifist, because she knows the generalization that most Quakers are pacifists. This is not a universal generalization, since there are some Quakers who are not pacifists. Moreover, being a pacifist is not an inherent feature of the meaning of *Quaker* – which denotes membership in a particular religious group. Rather, the generalization that Quakers are pacifists is a stereotype: part of culturally-specific world knowledge about Quakers. It is precisely this kind of knowledge which is best represented by a default rule.

How should default rules be expressed in DTT? Note that a stereotype like ‘Quakers are pacifists’ or ‘surgeons are male’ cannot be represented by a dependent function, like an arrow in a lexical network, since it is not universally true. Ellen Breitholz (2011, 2020) has developed a type-theoretic approach to dialogue semantics in which default rules are implemented by what she calls *enthymemes*, a kind of ‘free’ context update. An enthymeme can be thought of as an element of the type:

(508) $$\mathit{Enthymeme} \coloneqq \begin{bmatrix} \mathrm{P} : \mathit{Type} \\ \mathrm{P} \to \mathit{Type} \end{bmatrix}$$

that is, it takes a context satisfying a collection of presuppositions, and extends it by the addition of further information. (Note that, in contrast to an element of type *Update* – i.e. a sentence meaning – an enthymeme need not extend the context by only a single proposition, but can add an arbitrarily large amount of information.) To give an example, the stereotype ‘Quakers are pacifists’ would be written as follows:

(509) $$\begin{bmatrix} \mathrm{P} \coloneqq \begin{bmatrix} \mathrm{x} : \mathit{Ind} \\ \mathrm{Quaker(x)} \end{bmatrix} \\ \lambda \mathrm{p} : \mathrm{P} \;.\; \mathrm{pacifist(p.1)} \end{bmatrix}$$

This simple enthymeme takes a context in which an individual x is known to be a Quaker and adds the additional information that x is a pacifist. The procedure for updating a context using an enthymeme is very similar to the context update procedure described in 4.7. Given a current context $C_t : Type$ and an enthememe $E : Enthymeme$, one begins by finding a function

(510) $\sigma : C_t \to E.1$

from the context to the presuppositions of E. One then forms the extended context:

(511) $C_t' := \begin{bmatrix} \mathrm{c} : C_t \\ E.2(\sigma(\mathrm{c})) \end{bmatrix}$

which includes both the original context and the information added by the enthymeme. See Breitholtz (2020) for a detailed illustration of how enthymemes can be used in conjunction with a dialogue gameboard semantics to explain interactions like (507).

5.3.3 Optimality theoretic semantics

The next approach I shall compare with my own is an alternative conception of how a language user ought to decide between senses when interpreting a sentence with respect to a context. A number of authors (e.g. Blutner 2000, Hendriks & De Hoop 2001, Zeevat 2000, Zwarts 2004) have argued for a form of interpretation based on *Optimality Theory*, whereby the interpretation which gets selected is the one which best satisfies a system of ranked constraints. To illustrate this approach, I shall focus on Zwarts' (2004) analysis of the directional preposition *around*. The choice of this example is motivated in part because it is based on Vector Space Semantics, the same framework used in Chapter 3 for the description of spatial adjectives. It will therefore serve both to illustrate the Optimality Theoretic approach, and to show how Vector Space Semantics might be extended to other domains.

Zwarts begins by noting the diverse range of paths which may be described as *round* or *around*, as shown by sentences like the following:

(512) Senses of *round* (Zwarts 2004):
  a. The postman ran round the block.
  b. The burglar drove round the barrier.
  c. The steeplechaser ran round the corner.
  d. The tourist drove round.
  e. The driver took the long way round.
  f. The woman came round again.

For each of the above sentences, Figure 5.2 gives an example of a path which might fit the description. As discussed in Chapter 2, Zwarts understands a path as a continuous function from the real interval $[0, 1]$ to vectors, where each vector provides the position of a point along the path. The origin from which the vectors emanate is determined by the object of the preposition, or by some contextually understood reference point if no object is present.

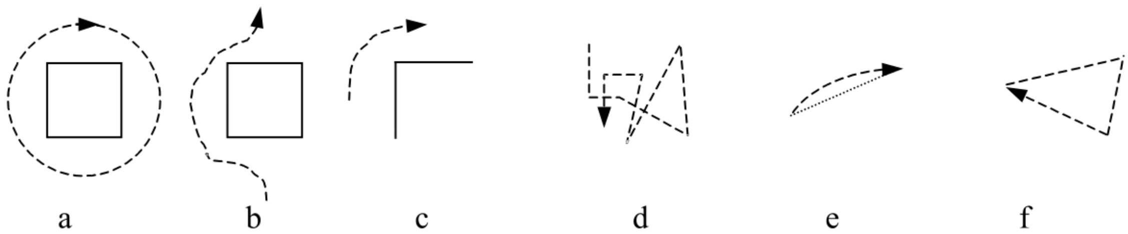


Figure 5.2: Paths described as *round*, from Zwarts (2004). Cases range from a perfectly closed circle, to a path which veers around an object, to a path which explores an area.

Zwarts takes the prototypical sense of *round* to refer to a circle, as shown in Figure 5.2a. He proceeds to formalize various properties of circular paths which are absent in less typical interpretations:

(513)
- COMPLETENESS: for every direction in a 2D plane, some vector in the path points in that direction
- CONSTANCY: all the vectors in the path maintain a constant length
- INVERSION: at least two vectors in the path point in opposite directions
- ORTHOGONALITY: at least two vectors in the path point in orthogonal directions
- DETOUR: the distance between the start and end point of the path is shorter than the length of the path itself
- LOOP: the start and end point of the path are identical

For each of the above properties, Zwarts shows that an interpretation of *round* can be found which does not include the property. He concludes that the weakest possible meaning of *round* is "any path". The various properties are partially ordered by strength, as follows:

(514)
$$\begin{tikzcd}
& \text{PATH} & \\
\text{INVERSION} \arrow[ur] & \text{DETOUR} \arrow[u] & \text{ORTHOGONALITY} \arrow[ul] \\
\text{CONSTANCY} \arrow[ur] & \text{COMPLETENESS} \arrow[ul] \arrow[u] \arrow[ur] & \text{LOOP} \arrow[ul] \\
& \text{CIRCLE} \arrow[ul] \arrow[u] \arrow[ur] &
\end{tikzcd}$$

To explain how a speaker decides on the correct interpretation of *round* in a given context, Zwarts invokes Optimality Theory (OT). OT is a theory in which different linguistic structures compete with each other with respect to a system of ranked constraints; the structure which gets selected is the one which optimally satisfies the constraints by incurring fewer violations than the alternatives. In this case, the alternatives which compete for selection are different interpretations of the word *round*. Zwarts proposes the following constraints:

(515)
- **Strength**: Prefer stronger intepretations to weaker interpretations
- **Fit**: Interpretations should not conflict with the context (including the linguistic context)
- **Vagueness:** The strongest interpretation should be avoided

The constraints **Strength** and **Fit** are present in some form in all versions of OT semantics: they work together to ensure that the interpreter chooses the strongest possible interpretation compatible with the context. **Fit** must be ranked over **Strength** to ensure that a weaker, consistent meaning is preferred over a stronger, inconsistent meaning. The role of **Vagueness** is to block the strongest possible interpretation, which according to Zwarts is usually too restricted.

The example which Zwarts uses to illustrate the theory is *round the door*. Following standard practice in OT, the competition between candidate interpretations is illustrated by means of a tableau, where rows correspond to possible candidates, columns correspond to constraints, and each cell registers a degree of violation:

(516)

	round the door	**Fit**	**Strength**
	a. COMPLETENESS, LOOP	*	
☞	b. INVERSION		*
	c. ORTHOGONALITY		**
	d. DETOUR		* * *

As shown, the candidate interpretations which Zwarts considers are: (a) a path satisfying COMPLETENESS and LOOP, (b) a path satisfying INVERSION, (c) a path satisfying ORTHGONALITY, and (d) a path satisfying DETOUR. Candidate (a) violates **Fit**, since the door is connected to a wall, making it impossible to have a complete path around it. The other three candidates are consistent, but differ in relative strength. Candidate (b), being the strongest, violates **Strength** the least, and gets selected as the optimal candidate.

There are a number of crucial differences between the OT semantics approach and the approach which I have advocated in this thesis. OT semantics begins from the assumption that all the possible interpretations are available all at once, together with their entailment relations. From our perspective, however, the entailment relations between senses cannot simply be taken for granted: rather, they must be computed on the basis of the constituents of the expression and the way they are combined. For instance, in a negative context like *The burglar drove, not round the barrier, but towards it*, the relative strength of interpretations is reversed, and the strongest sense of the entire sentence is derived from the weakest sense of *round*. The OT account therefore needs to be connected to a theory of how the relative strength of senses is derived, such as that given in this thesis.

Moreover, in my approach, the interpreter does not derive all the possible interpretations of an expression at once, and then evaluate them, because this is prohibitively expensive. Rather, interpretations are constructed in a breadth-first manner beginning with the most general, and building only as much structure as necessary in order to identify the strongest possible assertion compatible with the context. This procedure is not necessarily incompatible with OT semantics, but it is incompatible with the naive assumption often adopted in OT that the generator yields a list of all possible candidates prior to evaluation. An OT semantics in which (a) the relative ordering of senses is derived rather than taken for granted, and (b) candidates are constructed and evaluated in a breadth-first manner from most to least general, would be very similar to the interpretation strategy described in Section 4.7.2.

5.3.4 Neo-Gricean pragmatics

The final theoretical idea which I shall discuss in relation to my own approach is neo-Gricean pragmatics, as represented by the work of Horn (1972), Gazdar (1980), Levinson et al. (2000), Chierchia (2004), Sauerland (2004), and others. The primary focus of this work is explaining scalar implicatures such as the following:

(517) John ate some of the apples.
$\Rightarrow$ John did not eat all of the apples.

That this is a case of implicature as opposed to entailment is easily demonstrated using the standard cancellation test:

(518) John ate some of the apples. In fact, he ate all of them.

The starting point for all neo-Gricean approaches is the following kind of explanation. Given a choice between two sentences ϕ' and ϕ, where ϕ' is strictly more informative than ϕ (that is, ϕ' entails ϕ but ϕ does not entail ϕ'), a cooperative speaker ought to prefer the more informative option ϕ'. If the speaker instead utters ϕ, then one can conclude that they have a good reason for not uttering the more informative option ϕ': that is, either the speaker believes that ϕ' is false, or the speaker does not know whether ϕ' is true or false. To apply this to the implicature in (517), the two alternatives in this case are:

(519) a. John ate some of the apples.
b. John ate all of the apples.

A cooperative speaker should prefer (b) to (a), because (b) is more informative than (a). Given that the speaker has uttered (a), we can therefore conclude that they had a good reason for not uttering (b). There is no reason to assume ignorance on the part of the speaker, so we infer that (b) must be false.

The major challenge in neo-Gricean pragmatics is spelling out which stronger alternatives dialogue participants should take into account in the above kind of procedure. For any actual utterance, there are an infinite number of more informative alternatives which entail it. For example, the following are all more informative than (519a):

(520) a. John ate some of the apples and some of the pears.
b. John ate some of the apples and so did Mary.
c. John and his friends ate some of the apples.
d. ...

However, none of these stronger alternatives gives rise to a scalar implicature outside of certain very specific contexts. That is, we do not usually get any of the following implicatures:

(521) John ate some of the apples.
$\not\Rightarrow$ John did not eat any of the pears.
$\not\Rightarrow$ Mary did not eat any of the apples.
$\not\Rightarrow$ None of John's friends ate any of the apples
$\not\Rightarrow$ ...

Moreover, as Katzir (2007) has pointed out, negating all possible more informative alternatives can lead to contradictions. Consider the following sentences:

(522) a. John ate some of the apples.
b. John ate all of the apples.
c. John ate some but not all of the apples.

Both (b) and (c) are more informative than (a), so an utterance of (a) ought to generate the following implicatures:

(523) John ate some of the apples.
$\Rightarrow$ John didn't eat all of the apples.
$\Rightarrow$ John didn't eat some but not all of the apples.

The problem is that all three of these statements cannot be true at the same time. In general, for any sentence ϕ with a more informative alternative ϕ', the sentence $\phi \wedge \neg\phi'$ is also a more informative alternative, giving rise to two contradictory implicatures. At best, we would have to conclude that the speaker is unable to assert either ϕ or $\phi \wedge \neg\phi'$ due to ignorance, but then every sentence would give rise to an ignorance implicature.

Clearly, if the neo-Gricean account is to be preserved, the set of relevant alternatives for any given sentence needs to be somehow restricted. One proposal put forward by Horn (1972) and developed by Gazdar (1980) is based on lexically specified scales, often known as Horn scales. *All* is a valid alternative to *some* because they belong to a common Horn scale ⟨*all*, *some*⟩, whereas *some but not all* does not. Generating alternative candidates for a given sentence is a matter of substituting, for each word which belongs to a Horn scale, each of its stronger scale-mates. Other examples of Horn scales include ⟨*and*, *or*⟩, ⟨*always*, *often*, *sometimes*⟩, ⟨*necessary*, *possible*⟩ and so on. Like degrees, the items on a Horn scale must share the same polarity: the negative polarity scale ⟨*no*, *few*⟩ is distinct from its positive polarity counterpart ⟨*all*, *some*⟩.

Besides the use of Horn scales, there is also a non-lexical approach to restricting alternatives based on the notion of complexity. The idea is that the valid alternatives to a sentence are other sentences which are *at most as complex*. For this account to work, one needs to provide an adequate definition of complexity. Phonological definitions (e.g. number of syllables) yield incorrect results: what is needed is a definition which takes into account the syntactic structure of the sentence. Katzir proposes the following:

(524) STRUCTURAL COMPLEXITY (Katzir 2007, p.11): Let ϕ, ϕ' be parse trees. If we can transform ϕ into ϕ' by a finite series of deletions, contractions, and replacements of constituents of ϕ with constituents of the *substitution source* $L(\phi)$, then ϕ' is less structurally complex than ϕ, written $\phi' \lesssim \phi$.

where the substitution source $L(\phi)$ is defined as "the union of the lexicon of the language and the set of all subtrees of ϕ". The set of viable alternatives to ϕ which can feasibly license scalar implicatures is then given by all sentences which are (1) at most as structurally complex as ϕ, and (2) logically entail ϕ. To give an example of a complexity-based implicature, consider the following sentences:

(525) (ibid. p.9)
 a. If we meet John but not Mary it will be strange.
 b. If we meet John it will be strange.

Here, uttering (a) implicates that (b) is false. In other words, we get the implicature that not all situations in which we meet John will give rise to strangeness (provided he is accompanied by Mary). That $\neg$(b) is an implicature can be shown using explicit cancellation:

(526) If we meet John but not Mary it will be strange. In fact, if we meet John it will be strange.

As required, the complexity approach admits (b) as a viable alternative to (a), because it is less structurally complex than (a) and at the same time entails (a). On the other hand, this example cannot be explained by invoking Horn scales, since there is no Horn scale relating (a) and (b). Examples like these illustrate the need to combine Horn scales with a complexity-based approach.

Katzir suggests that the complexity approach removes the need to appeal to Horn scales altogether, claiming that "the effects of a seemingly stipulative definition of scales are directly predicted from our complexity-sensitive alternatives". The idea is that replacing a word with one of its scale-mates, e.g. *some* $\mapsto$ *all*, yields a structurally viable alternative, so the lexically specified scale $\langle$*some*, *all*$\rangle$ can be replaced by the possibility of arbitrary substitution. However, allowing for arbitrary substitutions in this way creates more problems than it solves. For instance, consider the following sentences:

(527)
 a. John saw a dog.
 b. John saw a Labrador.

According to Katzir's definition of structural complexity, (a) and (b) are equal in structural complexity since either can be derived from the other by lexical substitution. Moreover, (b) is a viable alternative to (a), because it semantically entails it. Therefore, uttering (a) ought to implicate $\neg$(b), which is not the case outside of certain very specific contexts. In order to allow substitutions like *some* $\mapsto$ *all* but disallow substitutions like *dog* $\mapsto$ *Labrador*, we must appeal to the fact that $\langle$*some*, *all*$\rangle$ form a Horn scale, whereas $\langle$*Labrador*, *dog*$\rangle$ do not.

Neo-Gricean pragmatics and my compositional theory of lexical networks have quite different aims. My framework describes the possible semantic interpretations which are licensed by a given sentence, the various connections between them, and how the interpreter should go about choosing the interpretation which best matches a given context. Neo-Gricean pragmatics, on the other hand, describes the effect of what the speaker 'might have said but didn't': it requires interpreting sentences which are different from the sentence which the speaker actually uttered. At the same time, the two approaches also have a number of similarities. Both rely crucially on the concept of informativeness or entailment, and both assume that the speaker is being as informative as possible. In my approach, the maximal-informativeness assumption is behind the search for the strongest interpretation compatible with the context; in neo-Gricean pragmatics, it is what licenses the inference that stronger sentences are unassertable.

Not only are the two approaches compatible, but the resources provided by the compositional theory of lexical networks are important for neo-Gricean inference. To begin with, notice that a Horn scale is really just another example of an implicational network. Horn scales are similar to the networks discussed in this thesis, except that they express relationships between the meanings of different words, and are usually total orders, meaning all the nodes are arranged in a single long chain. To give an example, the Horn scale $\langle$*all*, *some*$\rangle$ would correspond to the following simple network:

(528) $$[\![\text{all}]\!] \xrightarrow{\quad\alpha\quad} [\![\text{some}]\!]$$

where:

- $[\![\text{all}]\!], [\![\text{some}]\!] : (Ind \to Type) \to (Ind \to Update) \to Update$
- $\alpha : \prod_{N:(Ind\to Type)} \prod_{P:(Ind\to Update)} [\![\text{all}]\!](N)(P) \to_{Update} [\![\text{some}]\!](N)(P)$

That is, $[\![\text{all}]\!]$ and $[\![\text{some}]\!]$ are quantificational determiners, and α maps the presuppositions and assertions of $[\![\text{all}]\!](N)(P)$ to the presuppositions and assertions of $[\![\text{some}]\!](N)(P)$, for any N and P.

How does one go from a scale relating a group of words to a scale relating a group of sentences? In the compositional theory of lexical networks, the answer is automatic: by knowing how arrows are preserved under composition. For instance, consider the Horn scale $\langle$*always*, *often*, *sometimes*$\rangle$. The context *John X swims* lifts the Horn scale and preserves its order, as follows:

(529)
$$\begin{tikzcd} \llbracket\text{always}\rrbracket \arrow[d] & & \llbracket\text{John always swims}\rrbracket \arrow[d] \\ \llbracket\text{often}\rrbracket \arrow[d] & \longmapsto & \llbracket\text{John often swims}\rrbracket \arrow[d] \\ \llbracket\text{sometimes}\rrbracket & & \llbracket\text{John sometimes swims}\rrbracket \end{tikzcd}$$

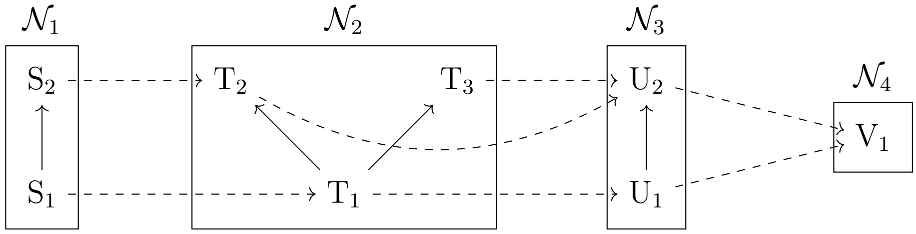
It follows that $\llbracket$John often swims$\rrbracket$ implicates $\neg\llbracket$John always swims$\rrbracket$. However, the negative context *John doesn't X swim* lifts the Horn scale in the opposite order:

(530)
$$\begin{tikzcd} \llbracket\text{always}\rrbracket \arrow[d] & & \llbracket\text{John doesn't always swim}\rrbracket \\ \llbracket\text{often}\rrbracket \arrow[d] & \longmapsto & \llbracket\text{John doesn't often swim}\rrbracket \arrow[u] \\ \llbracket\text{sometimes}\rrbracket & & \llbracket\text{John doesn't sometimes swim}\rrbracket \arrow[u] \end{tikzcd}$$

As a result, $\llbracket$John doesn't often swim$\rrbracket$ implicates $\neg\llbracket$John doesn't sometimes swim$\rrbracket$. There are also contexts which fail to preserve the Horn scale at all, such as the context *Jane said "John X swims"*: these contexts do not license any scalar implicatures whatsoever. The upshot is that the theory of Horn scales needs to be supplemented by an account of how implication is preserved under composition. The compositional theory of lexical networks developed in this thesis automatically provides such an account, because every word which is capable of acting on an implication network contains as part of its lexical entry both an action on senses and an action on arrows.

We might imagine a synthesis of the theory of lexical networks and the theory of Horn scales along the following lines. Given multiple lexical networks *of the same type* $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \ldots$, a Horn scale $\langle \mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \ldots \rangle$ consists of a monotone family of arrows from senses of $\mathcal{N}_i$ to senses of $\mathcal{N}_j$, for every $i < j$. For example, the following is a valid Horn scale:

(531)



Notice that the inter-network (dashed) arrows are monotone with respect to the intra-network (bold) arrows. In general, the networks on a Horn scale need not have the same number of senses: some may have many, others only one. Now suppose that the speaker utters a sentence involving the word U – the third network in the diagram. The hearer begins by searching through the U network for a sense which is compatible with the context. Suppose that the hearer discovers that U_1

is compatible with the context. She should then construct an alternative version of the sentence, replacing U_1 with T_1, since this lies directly above U_1 on the Horn scale. As in the usual neo-Gricean account, she then tries to explain why the speaker did not utter T_1: either (a) T_1 is incompatible with the context, (b) T_1 is compatible and the speaker disbelieves its assertion, or (c) T_1 is compatible and the speaker is unsure whether its assertion is true or false. Options (b) and (c) will then give rise to an implicature in the usual way.

5.4 Conclusion

In Chapter 1, I described two approaches to meaning in language: a 'downwards' approach which looks at the senses of individual words, how these are grounded in cognition and perception, and how they are connected together into networks; and an 'upwards' approach which is concerned with how words combine together to yield the truth conditions of sentences, and how sentences contribute to an ongoing discourse. The former is associated with lexical semantics and cognitive linguistics (e.g. Fillmore 1976, Lakoff 1987, Tyler & Evans 2003), whereas the latter is associated with formal semantics in the Montague tradition (e.g. Montague 1973, Partee 1986, Heim & Kratzer 1998). This thesis has been an attempt to take a small class of words – the English spatial adjectives – and look 'in both directions at once', exploring both their lexical semantics and their compositional semantics. To achieve this, it was necessary to develop a general framework within which lexical and compositional semantics can interact.

A central idea in this thesis which I take to be important for the general study of language and mind is that cognitive semantics and formal semantics are fundamentally compatible. I have argued that the view that the two fields are incompatible derives from a realist construal of formal semantics, according to which the truth conditions of a sentence are given with respect to a mind-independent model of the real world (e.g. Lewis 1972, Abbott 1997). The alternative, known as conceptualism or internalism, holds that truth conditions are given with respect to a model of the world *as conceptualized by language users* (Bach 1986, Verkuyl 1989, Jackendoff 1998). The various components of the formal model – individuals, events, times, degrees, vectors, and so on – are not mind-independent entities, but part of a shared conceptual world. On this view, the difference between formal and cognitive approaches has more to do with one's taste for formal language than any fundamental disagreement.

My hope is that in the future, formalists and cognitivists will come to take each other's concerns more seriously. Formal semantics is concerned above all with compositionality and inference. A word like *wide* can be treated as atomic because the important thing is how it contributes to a larger expression such as *wider than 10cm*, *as wide as a car*, *the widest object in the room*, and so forth. Cognitivists, on the other hand, are most concerned with the connection between language and the rest of the mind. Far from being atomic, a word like *wide* becomes an entire domain of enquiry because of the complex conceptualization which it invokes. There is a tendency for both fields to neglect precisely what the other considers to be important. It is by bearing in mind both kinds of complexity, and recognising the connections between them, that we can hope to make progress. This thesis represents a small step in the direction of such a unification.

Bibliography

Abbott, B. (1997), ‘Models, truth and semantics’, *Linguistics and Philosophy* pp. 117–138.

Abney, S. P. (1987), The English noun phrase in its sentential aspect, PhD thesis, Massachusetts Institute of Technology.

Asher, N. (2011), *Lexical Meaning in Context: A web of words*, Cambridge University Press.

Asudeh, A., Giorgolo, G., Butt, M. & King, T. H. (2012), Flexible composition for optional and derived arguments, *in* ‘Proceedings of the LFG12 Conference’, pp. 64–84.

Bach, E. (1986), Natural language metaphysics, *in* ‘Studies in Logic and the Foundations of Mathematics’, Vol. 114, Elsevier, pp. 573–595.

Barras, B., Boutin, S., Cornes, C., Courant, J., Filliatre, J.-C., Gimenez, E., Herbelin, H., Huet, G., Munoz, C., Murthy, C. et al. (1997), ‘The Coq proof assistant reference manual: Version 6.1’.

Bartsch, R. & Vennemann, T. (1974), ‘Semantic structures’, *Foundations of Language* **12**(2).

Barwise, J. & Cooper, R. (1981), Generalized quantifiers and natural language, *in* ‘Philosophy, Language, and Artificial Intelligence’, Springer, pp. 241–301.

Bekki, D. (2014), Representing anaphora with dependent types, *in* ‘International Conference on Logical Aspects of Computational Linguistics’, Springer, pp. 14–29.

Biederman, I. (1987), ‘Recognition-by-components: A theory of human image understanding.’, *Psychological Review* **94**(2), 115.

Bierwisch, M. (1967), ‘Some semantic universals of German adjectivals’, *Foundations of Language* pp. 1–36.

Bierwisch, M. (1989), The semantics of gradation, *in* ‘Dimensional Adjectives: Grammatical structure and conceptual interpretation’, Springer, pp. 71–261.

Blutner, R. (2000), ‘Some aspects of optimality in natural language interpretation’, *Journal of Semantics* **17**(3), 189–216.

Boldini, P. (2000), 'Formalizing context in intuitionistic type theory', *Fundamenta Informaticae* **42**(2), 105–127.

Bove, A., Dybjer, P. & Norell, U. (2009), A brief overview of Agda – a functional language with dependent types, *in* 'International Conference on Theorem Proving in Higher Order Logics', Springer, pp. 73–78.

Breitholtz, E. (2020), Enthymematic reasoning and pragmatics, *in* 'Enthymemes and Topoi in Dialogue', Brill, pp. 16–36.

Breitholtz, E. & Cooper, R. (2011), Enthymemes as rhetorical resources, *in* 'Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2011)', Citeseer, pp. 149–157.

Bresnan, J., Asudeh, A., Toivonen, I. & Wechsler, S. (2015), *Lexical-Functional Syntax*, John Wiley & Sons.

Bresnan, J. W. (1973), 'Syntax of the comparative clause construction in English', *Linguistic inquiry* **4**(3), 275–343.

Brugman, C. & Lakoff, G. (1988), Cognitive topology and lexical networks, *in* 'Lexical Ambiguity Resolution', Elsevier, pp. 477–508.

Carlson, G. N. (1977), 'A unified analysis of the English bare plural', *Linguistics and Philosophy* **1**(3), 413–457.

Carlson, L., Regier, T. & Covey, E. (2003), 'Defining spatial relations: Reconciling axis and vector representations', *Representing Direction in Language and Space* pp. 111–131.

Carnap, R. (1952), 'Meaning postulates', *Philosophical studies* **3**(5), 65–73.

Chatzikyriakidis, S. & Luo, Z. (2014), Using signatures in type theory to represent situations, *in* 'JSAI International Symposium on Artificial Intelligence', Springer, pp. 172–183.

Chatzikyriakidis, S. & Luo, Z. (2017*a*), *Modern Perspectives in Type-Theoretical Semantics*, Springer.

Chatzikyriakidis, S. & Luo, Z. (2017*b*), On the interpretation of common nouns: Types versus predicates, *in* 'Modern Perspectives in Type-Theoretical Semantics', Springer, pp. 43–70.

Chatzikyriakidis, S. & Luo, Z. (2020), *Formal Semantics in Modern Type Theories*, John Wiley & Sons.

Chierchia, G. (2004), 'Scalar implicatures, polarity phenomena, and the syntax/pragmatics interface', *Structures and Beyond* **3**, 39–103.

Clark, H. H. (1973), Space, time, semantics, and the child, *in* 'Cognitive Development and Acquisition of Language', Elsevier, pp. 27–63.

Cobreros, P. (2011), ‘Paraconsistent vagueness: a positive argument’, *Synthese* **183**(2), 211–227.

Coleman, L. & Kay, P. (1981), ‘Prototype semantics: The English verb lie’, *Language* **57**, 26–44.

Connolly, A. C., Fodor, J. A., Gleitman, L. R. & Gleitman, H. (2007), ‘Why stereotypes don’t even make good defaults’, *Cognition* **103**(1), 1–22.

Cooper, R. (2005), ‘Records and record types in semantic theory’, *Journal of Logic and Computation* **15**(2), 99–112.

Cooper, R. (2012), ‘Type theory and semantics in flux’, *Handbook of the Philosophy of Science* **14**(2012), 271–323.

Cooper, R. (2016), *Type Theory and Language: From perception to linguistic communication*, Draft of book chapters available from https://sites. google. com/site/typetheorywithrecords/drafts.

Cooper, R., Dobnik, S., Lappin, S. & Larsson, S. (2015), Probabilistic type theory and natural language semantics, *in* ‘Linguistic Issues in Language Technology’, Vol. 10.

Cooper, R. & Ginzburg, J. (2015), ‘Type theory with records for natural language semantics’, *The Handbook of Contemporary Semantic Theory* pp. 375–407.

Coquand, T. (1986), The calculus of constructions, PhD thesis, INRIA.

Corver, N. F. M. (1991), The syntax of left branch extractions, PhD thesis, Tilburg University.

Cresswell, M. J. (1976), The semantics of degree, *in* ‘Montague Grammar’, Elsevier, pp. 261–292.

Dalrymple, M., Kanazawa, M., Mchombo, S. & Peters, S. (1994), What do reciprocals mean?, *in* ‘Semantics and Linguistic Theory’, Vol. 4, pp. 61–78.

Dirven, R. & Taylor, J. (1986), *The conceptualisation of vertical space in English: the case of "tall"*, LAUD.

Dorst, L., Fontijne, D. & Mann, S. (2009), *Geometric Algebra for Computer Science: An object-oriented approach to geometry*, Morgan Kaufmann Publishers Inc.

Dowty, D. R., Wall, R. & Peters, S. (2012), *Introduction to Montague Semantics*, Vol. 11, Springer.

Dummett, M. et al. (1991), *The Logical Basis of Metaphysics*, Harvard University Press.

Durrell, M. (1988), ‘Zu einigen Deutschen und Englischen Dimensionsadjektiven. Eine vergleichende Analyse’, *Deutscher Wortschatz. Lexikologischen Studien* pp. 93–115.

Eschenbach, C., Tschander, L., Habel, C. & Kulik, L. (2000), Lexical specifications of paths, *in* ‘Spatial Cognition II’, Springer, pp. 127–144.

Evans, V. (2006), *Cognitive Linguistics*, Edinburgh University Press.

Evans, V. (2013), *Language and Time: A cognitive linguistics approach*, Cambridge University Press.

Faller, M. (2000), ‘Dimensional adjectives and measure phrases in vector space semantics’, *Formalizing the Dynamics of Information.* pp. 151–170.

Farkas, D. F. & Kiss, K. É. (2000), ‘On the comparative and absolute readings of superlatives’, *Natural Language & Linguistic Theory* **18**(3), 417–455.

Farmer, W. M. (2008), ‘The seven virtues of simple type theory’, *Journal of Applied Logic* **6**(3), 267–286.

Fauconnier, G. (1975), ‘Polarity and the scale principle’, *Proceedings of the Chicago Linguistic Society* .

Fernández, R. & Larsson, S. (2014), Vagueness and learning: A type-theoretic approach, *in* ‘Proceedings of the Third Joint Conference on Lexical and Computational Semantics (SEM 2014)’, pp. 151–159.

Field, H. (2003), ‘No fact of the matter’, *Australasian Journal of Philosophy* **81**(4), 457–480.

Fillmore, C. J. (1976), Frame semantics and the nature of language, *in* ‘Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech’, Vol. 280, pp. 20–32.

Fillmore, C. J. (1997), *Lectures on Deixis*, CSLI Publications.

Fine, K. (1975), ‘Vagueness, truth and logic’, *Synthese* **30**(3), 265–300.

Fodor, J. & Lepore, E. (1996), ‘The red herring and the pet fish: Why concepts still can’t be prototypes’, *Cognition* **58**(2), 253–270.

Gazdar, G. (1980), ‘Pragmatics, implicature, presupposition and logical form’, *Critica* **12**(35).

Geckeler, H. (1997), ‘Réflexions sur le champ lexical adjectival des dimensions spatiales du Français’, *L’Organisation Lexicale et Cognitive des Dimensions Spatiales et Temporelles: actes d’EUROSEM 1996* pp. 95–106.

Goguen, J. A. (1969), ‘The logic of inexact concepts’, *Synthese* pp. 325–373.

Goy, A. (2002), Grounding meaning in visual knowledge, *in* 'Spatial Language', Springer, pp. 121–145.

Greimas, A. J. (1966), 'Sémantique structurale', *Recherche de Méthode* .

Grice, H. P. (1975), Logic and conversation, *in* 'Speech Acts', Brill, pp. 41–58.

Grimshaw, J. (1991), Extended projection, Master's thesis, Brandeis University.

Groenendijk, J. & Stokhof, M. (1991), 'Dynamic predicate logic', *Linguistics and Philosophy* pp. 39–100.

Grzegorczykowa, R. (1997), *Projekt syntezy badań porównawczych w zakresie nazw wymiarów.*, Stockholms Univ. Slaviska institutionen.

Hazout, I. (1995), 'Comparative ellipsis and logical form', *Natural Language & Linguistic Theory* **13**(1), 1–37.

Heim, I. (1982), The semantics of definite and indefinite noun phrases, PhD thesis, University of Massachusetts Amherst.

Heim, I. (1983), 'File change semantics and the familiarity theory of definiteness', *Semantics: Critical Concepts in Linguistics* pp. 108–135.

Heim, I. (1985), 'Notes on comparatives and related matters', *Ms., University of Texas, Austin* .

Heim, I. (1995), 'Notes on superlatives', *Ms., MIT* .

Heim, I. (2000), Degree operators and scope, *in* 'Semantics and Linguistic Theory', Vol. 10, pp. 40–64.

Heim, I. & Kratzer, A. (1998), *Semantics in Generative Grammar*, Blackwell, Oxford.

Hendriks, P. & De Hoop, H. (2001), 'Optimality theoretic semantics', *Linguistics and Philosophy* **24**(1), 1–32.

Herskovits, A. (1987), *Language and Spatial Cognition*, Cambridge University Press.

Horn, L. (1984), 'Toward a new taxonomy for pragmatic inference: Q-based and R-based implicature', *Meaning, Form, and Use in Context: Linguistic applications* **11**, 42.

Horn, L. R. (1972), *On the semantic properties of logical operators in English*, University of California, Los Angeles.

Hyde, D. & Colyvan, M. (2008), 'Paraconsistent vagueness: Why not?', *The Australasian Journal of Logic* **6**.

Izvorski, R. (1995), A solution to the subcomparative paradox, *in* 'Proceedings of WCCFL', Vol. 14, pp. 203–219.

Jackendoff, R. (1975), ‘Morphological and semantic regularities in the lexicon’, *Language* pp. 639–671.

Jackendoff, R. (1983), *Semantics and Cognition*, MIT press.

Jackendoff, R. (1989), ‘What is a concept, that a person may grasp it?’, *Mind & language* **4**(1-2), 68–102.

Jackendoff, R. (1996), ‘Conceptual semantics and cognitive linguistics’, *Cognitive Linguistics* **7**(1), 93–129.

Jackendoff, R. (1998), ‘Why a conceptualist view of reference? A reply to Abbott’, *Linguistics and Philosophy* pp. 211–219.

Jackendoff, R. (2002), *Foundations of Language: Brain, meaning, grammar, evolution*, Oxford University Press.

Jackendoff, R. & Aaron, D. (1991), ‘Review of ‘More than Cool Reason: A field guide to poetic metaphor”, *Language* **67**(2), 320–328.

Janssen, T. M. V. & Zimmermann, T. E. (2021), Montague Semantics, *in* E. N. Zalta, ed., ‘The Stanford Encyclopedia of Philosophy’, Summer 2021 edn, Metaphysics Research Lab, Stanford University.

Johnson, M. (1987), *The Body in the Mind: The bodily basis of meaning, imagination, and reason*, University of Chicago Press.

Kamp, H. (1975), Two theories about adjectives, *in* ‘Meaning and the Dynamics of Interpretation’, Brill, pp. 225–261.

Kamp, H. (1980), A theory of truth and semantic representation, *in* ‘Formal Semantics: The essential readings’, Blackwell, Oxford, pp. 189–222.

Kamp, H. & Partee, B. (1995), ‘Prototype theory and compositionality’, *Cognition* **57**(2), 129–191.

Kamp, H., Van Genabith, J. & Reyle, U. (2011), Discourse representation theory, *in* ‘Handbook of Philosophical Logic’, Springer, pp. 125–394.

Karttunen, L. (1974), ‘Presupposition and linguistic context’, *Theoretical Linguistics* **1**(1-3), 181–194.

Katz, J. J. & Fodor, J. A. (1963), ‘The structure of a semantic theory’, *Language* **39**(2), 170–210.

Katzir, R. (2007), ‘Structurally-defined alternatives’, *Linguistics and Philosophy* **30**(6), 669–690.

Kennedy, C. (1999), *Projecting the Adjective: The syntax and semantics of gradability and comparison*, Routledge.

Kennedy, C. (2007), 'Vagueness and grammar: The semantics of relative and absolute gradable adjectives', *Linguistics and Philosophy* **30**(1), 1–45.

Kennedy, C. & McNally, L. (2005), 'Scale structure, degree modification, and the semantics of gradable predicates', *Language* pp. 345–381.

Klein, E. (1980), 'A semantics for positive and comparative adjectives', *Linguistics and Philosophy* **4**(1), 1–45.

Klein, E. (1982), 'The interpretation of adjectival comparatives', *Journal of Linguistics* **18**(1), 113–136.

Krahmer, E. & Piwek, P. (1999), Presupposition projection as proof construction, *in* 'Computing Meaning', Springer, pp. 281–300.

Krifka, M. (1998), The origins of telicity, *in* 'Events and Grammar', Springer, pp. 197–235.

Kripke, S. A. (2009), 'Presupposition and anaphora: Remarks on the formulation of the projection problem', *Linguistic Inquiry* **40**(3), 367–386.

Ladusaw, W. (1980), On the notion affective in the analysis of negative-polarity items, *in* 'Formal Semantics: The essential readings', Blackwell, Oxford, p. 470.

Ladusaw, W. A. (1979), Polarity Sensitivity as Inherent Scope Relations., PhD thesis, The University of Texas at Austin.

Lafrenz, P. G. (1983), 'Zu den semantischen Strukturen der Dimensionsadjektive in der deutschen Gegenwartssprache'.

Lakoff, G. (1987), *Women, Fire, and Dangerous Things: What categories reveal about the mind*, University of Chicago Press.

Lakoff, G. & Johnson, M. (1980), *Metaphors We Live By*, University of Chicago Press.

Lang, E. (1989), The semantics of dimensional designation of spatial objects, *in* 'Dimensional adjectives: Grammatical structure and conceptual interpretation', Springer, pp. 263–417.

Lang, E. (2001), 'Spatial dimension terms', *Language Typology and Language Universals. An International Handbook* **2**, 1251–1275.

Larsson, S. (2015), 'Formal semantics for perceptual classification', *Journal of Logic and Computation* **25**(2), 335–369.

Lassiter, D. (2017), *Graded Modality: Qualitative and quantitative perspectives*, Oxford University Press.

Lerner, J. & Pinkal, M. (1995), Comparative ellipsis and variable binding, *in* 'Semantics and Linguistic Theory', Vol. 5, pp. 222–236.

Levinson, S. C. (1996), ‘Frames of reference and molyneux’s question: Crosslinguistic evidence’, *Language and Space* **109**, 169.

Levinson, S. C. (2003), *Space in Language and Cognition: Explorations in cognitive diversity*, Cambridge University Press.

Levinson, S. C., Stephen, C. & Levinson, S. C. (2000), *Presumptive Meanings: The theory of generalized conversational implicature*, MIT press.

Lewis, D. (1972), General semantics, *in* ‘Semantics of Natural Language’, Springer, pp. 169–218.

Lewis, D. (1979), Scorekeeping in a language game, *in* ‘Semantics from Different Points of View’, Springer, pp. 172–187.

Lewis, D. (1982), ‘Logic for equivocators’, *Noûs* **16**(3), 431–441.

Linde-Usiekniewicz, J. (2002), *Dimension Terms in Polish*, Warszawa: Wydział Polonistyki Uniwersytetu Warszawskiego.

Link, G. (1980), The logical analysis of plurals and mass terms: A lattice-theoretical approach, *in* ‘Formal Semantics: The essential readings’, Blackwell, Oxford, p. 147.

Ludlow, P. & Segal, G. (2004), On a unitary semantical analysis for definite and indefinite descriptions, *in* M. Reimer & A. Bezuidenhout, eds, ‘Descriptions and Beyond’, Oxford University Press, pp. 420–437.

Luo, Z. (1999), ‘Coercive subtyping’, *Journal of Logic and Computation* **9**(1), 105–130.

Luo, Z. (2010), Type-theoretical semantics with coercive subtyping, *in* ‘Semantics and Linguistic Theory’, Vol. 20, pp. 38–56.

Luo, Z. (2012), ‘Formal semantics in modern type theories with coercive subtyping’, *Linguistics and Philosophy* **35**(6), 491–513.

Luo, Z. (2014), Formal semantics in modern type theories: Is it model-theoretic, proof-theoretic, or both?, *in* ‘International Conference on Logical Aspects of Computational Linguistics’, Springer, pp. 177–188.

Lyons, J. (1977), *Semantics, Volume I*, Cambridge University Press.

Marr, D. (1982), *Vision: A computational investigation*, WH Freeman.

Marr, D. & Nishihara, H. K. (1978), ‘Representation and recognition of the spatial organization of three-dimensional shapes’, *Proceedings of the Royal Society of London. Series B. Biological Sciences* **200**(1140), 269–294.

Martin-Löf, P. (1984), *Intuitionistic Type Theory*, Bibliopolis Naples.

Miller, G. A. & Johnson-Laird, P. N. (1976), *Language and Perception*, Belknap Press.

Moltmann, F. (1992), The empty element in comparatives, *in* 'Proceedings of NELS', Vol. 23, pp. 319–333.

Montague, R. (1973), The proper treatment of quantification in ordinary English, *in* 'Approaches to Natural Language', Springer, pp. 221–242.

Moore, K. E. (2006), 'Space-to-time mappings and temporal concepts', *Cognitive Linguistics* **17**(2), 199–244.

Norvig, P. (1989), Building a large lexicon with lexical network theory, *in* 'Proceedings of the IJCAI Workshop on Lexical Acquisition'.

Norvig, P. & Lakoff, G. (1987), Taking: A study in lexical network theory, *in* 'Annual Meeting of the Berkeley Linguistics Society', Vol. 13, pp. 195–206.

O'Keefe, J. (1996), The spatial prepositions in English, vector grammar, and the cognitive map theory, *in* 'Language and Space', MIT Press, Cambridge, MA.

O'Keefe, J. & Nadel, L. (1978), *The Hippocampus as a Cognitive Map*, Oxford: Clarendon Press.

Osherson, D. N. & Smith, E. E. (1981), 'On the adequacy of prototype theory as a theory of concepts', *Cognition* **9**(1), 35–58.

Paradis, C. (2001), 'Adjectives and boundedness', *Cognitive Linguistics* **12**, 47–64.

Partee, B. H. (1986), *Noun Phrase Interpretation and Type-Shifting Principles*, De Gruyter.

Piñón, C. J. (1993), Paths and their names, *in* 'Chicago Linguistics Society', Vol. 29, pp. 287–303.

Pustejovsky, J. (1995), *The Generative Lexicon*, MIT press.

Radden, G. (2003), 'The metaphor TIME AS SPACE across languages', *Zeitschrift für interkulturellen Fremdsprachenunterricht* **8**(2).

Rakhilina, E. V. (2000), 'Kognitivnyj analizpredmetnyx imen: Semantika i socetaemost [A cognitive analysis of common nouns: Semantics and combinability]', *Moskva: Russkie slovari* .

Ranta, A. (1994), *Type-theoretical grammar*, Oxford University Press.

Reiter, R. (1980), 'A logic for default reasoning', *Artificial Intelligence* **13**(1-2), 81–132.

Rosch, E. (1975), 'Cognitive representations of semantic categories.', *Journal of Experimental Psychology: General* **104**(3), 192.

Rosch, E. (1999), 'Principles of categorization', *Concepts: Core readings* **189**.

Rosch, E. H. (1973), 'Natural categories', *Cognitive psychology* **4**(3), 328–350.

Sauerland, U. (2004), 'Scalar implicatures in complex sentences', *Linguistics and Philosophy* **27**(3), 367–391.

Schmidt, L. A., Goodman, N. D., Barner, D. & Tenenbaum, J. B. (2009), How tall is tall? Compositionality, statistics, and gradable adjectives, *in* 'Proceedings of the 31st Annual Conference of the Cognitive Science Society', Citeseer, pp. 2759–2764.

Schwarzschild, R. (2005), 'Measure phrases as modifiers of adjectives', *Recherches Linguistiques de Vincennes* pp. 207–228.

Sellars, W. (1963), 'Philosophy and the scientific image of man', *Science, Perception and Reality* **2**, 35–78.

Seuren, P. A. (1994), Discourse semantics, *in* 'The Encyclopedia of Language and Linguistics (vol. 2)', Pergamon Press, pp. 982–993.

Sharvit, Y. & Stateva, P. (2002), 'Superlative expressions, context, and focus', *Linguistics and Philosophy* pp. 453–504.

Shimotori, M. (2013), Conceptual contrasts: A comparative semantic study of dimensional adjectives in Japanese and Swedish, PhD thesis, Umeå Universitet.

Smith, E. E., Osherson, D. N., Rips, L. J. & Keane, M. (1988), 'Combining prototypes: A selective modification model', *Cognitive Science* **12**(4), 485–527.

Spang-Hanssen, E. (1990), 'La sémantique des adjectifs spatiaux', *Revue Romane* **25**(2), 292–309.

Stalnaker, R. (1974), 'Pragmatic presuppositions', *Semantics and Philosophy* .

Stolz, C. (1996), Spatial dimensions and orientation of objects in Yucatec Maya, PhD thesis, Bielefeld University.

Sundholm, G. (1989), 'Constructive generalized quantifiers', *Synthese* pp. 1–12.

Svenonius, P. & Kennedy, C. (2008), Northern norwegian degree questions and the syntax of measurement, *in* 'Phases of interpretation', De Gruyter Mouton, pp. 133–162.

Szabolcsi, A. (1986), Comparative superlatives, *in* 'Papers in Theoretical Linguistics', MIT Working Papers in Linguistics, pp. 245–266.

Tanaka, R. (2014), 'A proof-theoretic approach to generalized quantifiers in dependent type semantics', *Proceedings of the ESSLLI 2014 Student Session* pp. 140–151.

Tanaka, R., Mineshima, K. & Bekki, D. (2017), 'Factivity and presupposition in dependent type semantics', *Journal of Language Modelling* **5**.

Tye, M. (1994), 'Sorites paradoxes and the semantics of vagueness', *Philosophical Perspectives* **8**, 189–206.

Tyler, A. & Evans, V. (2001), 'Reconsidering prepositional polysemy networks: The case of over', *Language* pp. 724–765.

Tyler, A. & Evans, V. (2003), *The Semantics of English Prepositions: Spatial scenes, embodied meaning, and cognition*, Cambridge University Press.

UFP (2013), *Homotopy Type Theory: Univalent Foundations of Mathematics*, `https://homotopytypetheory.org/book`, Institute for Advanced Study.

Vandeloise, C. (1988), Length, width, and potential passing, *in* 'Topics in Cognitive Linguistics', John Benjamins, p. 403.

Vandeloise, C. (1993), 'The role of resistance in the meanings of thickness', *Leuvense Bijdragen* **82**(1), 29–47.

Verkuyl, H. J. (1989), 'Aspectual classes and aspectual composition', *Linguistics and Philosophy* **12**(1), 39–94.

Verkuyl, H. & Zwarts, J. (1992), 'Time and space in conceptual and logical semantics: the notion of path'.

Vogel, A. (2004), Swedish Dimensional Adjectives, PhD thesis, Stockholm University.

von Stechow, A. (1984), 'Comparing semantic theories of comparison', *Journal of Semantics* **3**(1-2), 1–77.

Weydt, H. & Schlieben-Lange, B. (1998), 'The meaning of dimensional adjectives: Discovering the semantic process', *Lexicology* **4**(2), 199–236.

Wheeler, S. C. (1972), 'Attributives and their modifiers', *Noûs* pp. 310–334.

Williamson, T. (2002), *Vagueness*, Routledge.

Williamson, T. & Simons, P. (1992), 'Vagueness and ignorance', *Proceedings of the Aristotelian Society, Supplementary Volumes* **66**, 145–177.

Winter, Y. (1996), What does the strongest meaning hypothesis mean?, *in* 'Semantics and Linguistic Theory', Vol. 6, pp. 295–310.

Winter, Y. (2001*a*), Measure phrase modification in vector space semantics, *in* 'Proceedings of WCCFL', Vol. 20, Citeseer, pp. 607–620.

Winter, Y. (2001*b*), 'Plural predication and the strongest meaning hypothesis', *Journal of Semantics* **18**(4), 333–365.

Winter, Y. (2005), 'Cross-categorial restrictions on measure phrase modification', *Linguistics and Philosophy* **28**(2), 233–267.

Wittgenstein, L. (1958), *Philosophical Investigations*, Blackwell, Oxford.

Zadeh, L. A. (1975), 'Fuzzy logic and approximate reasoning', *Synthese* **30**(3-4), 407–428.

Zeevat, H. (2000), 'The asymmetry of optimality theoretic syntax and semantics', *Journal of Semantics* **17**(3), 243–262.

Zwarts, J. (1997), 'Vectors as relative positions: A compositional semantics of modified PPs', *Journal of Semantics* **14**(1), 57–86.

Zwarts, J. (2003), Vectors across spatial domains: From place to size, orientation, shape and parts, *in* 'Representing Direction in Language and Space', Oxford University Press, pp. 39–68.

Zwarts, J. (2004), Competition between word meanings: The polysemy of (a)round, *in* 'Proceedings of Sinn und Bedeutung', Vol. 8, pp. 349–360.

Zwarts, J. (2005), 'Prepositional aspect and the algebra of paths', *Linguistics and Philosophy* **28**(6), 739–779.

Zwarts, J. & Verkuyl, H. (1994), 'An algebra of conceptual structure; an investigation into Jackendoff's conceptual semantics', *Linguistics and Philosophy* **17**(1), 1–28.

Zwarts, J. & Winter, Y. (2000), 'Vector space semantics: A model-theoretic analysis of locative prepositions', *Journal of Logic, Language and Information* **9**(2), 169–211.