A geometrically inspired category as a meaning space for natural language

Jozef Barnabas Houben

St Anne's College

University of Oxford

A dissertation submitted for the degree of

MSc in Mathematics and Foundations of Computer Science

Trinity 2017

Abstract

In this dissertation we apply the theory of conceptual spaces to the domain of shapes to develop a mathematically rigorous conceptual space providing a natural, geometrically motivated encoding of shapes that can be used in the compositional distributional model of meaning. To this end, we abstractly treat hierarchical conceptual spaces, developing the mathematical tools required to define these spaces and reason about them. In particular, we define generalisations of the notions of metrics and convexity and apply these generalisations to our conceptual space. We then define a category for these spaces and discuss some of the aspects of language that the model captures. We specialise this category to the specific domain of shapes by implementing a variant of the Marr-Nishihara model. We also introduce a model for representing interactions with prepositions.

Contents

Chapter 1

Introduction

Natural language processing is a field which has seen much excitement and grabbed a lot of attention in recent years. Tools such as personal assistance promise us easier, more comfortable lives. However, much of the development in these head-line grabbing technologies has been due to the advent of more powerful computers, and the discovery of algorithms that allow for automated learning of language tasks without giving much insight into the structure of language and human thought.

Compositional approaches to natural language have much more promise in this regard, but have simply not been as effective at producing tangible results. In 2010, Coecke et al. [CSC10] introduced a compositional distributional model which combined this qualitative compositional approach with a quantitative distributional approach. The model utilized techniques from categorical quantum mechanics, thus providing a diagrammatic way to reason about language. Moreover, the model outperformed other models in tasks such as computing sentence similarity ([GS11]). Subsequently, the model has been extended to encode notions such as entailment and ambiguity ([AC16]). Recently, interest has arisen in combining this compositional distributional approach with the theory of conceptual spaces spearheaded by Gärdenfors ([Gär00, Gär14]), and some early work in this area was done by Bolt et al. [BCG$^+$16]. In the present work, we aim to apply the theory of conceptual spaces to the domain of shapes, thus developing a mathematically rigorous conceptual space providing a natural, geometrically motivated encoding of shapes, that can be used in the compositional distributional model.

1.1 Overview

In the first part of this work, we will review existing theory. For this, some knowledge of categorical quantum mechanics and the graphical calculus is assumed. Readers who are not familiar with this material are referred to [HV17]. In chapter 2 we give an overview of the compositional distributional model of meaning. Then, in chapter 3, we discuss the aspects of the theory of conceptual spaces that are relevant for our purposes.

Then, in the second part of this dissertation, we will construct a hierarchical category for representing shapes. Chapter 4 is dedicated to discussing the notion of such spaces and developing the mathematical tools required to define these spaces and reason about them. In particular, we define generalisations of the notions of metrics and convexity and apply these generalisations to our conceptual space. In chapter 5, we define a category based on these spaces and discuss some of the aspects of language that this category captures. Then, in chapter 6 we specialise the category to the specific domain of shapes by implementing a variant of a model first developed by Marr and Nishihara ([MN78]). In that chapter we also introduce a model for representing interactions with prepositions. Finally, in chapter 7, we give an overview of our results, evaluate the strength and weaknesses of the model that was developed, and suggest avenues for further research.

Part I

Review of existing theory

Chapter 2

Compositional distributional models of meaning

Traditionally, there have been two partially orthogonal approaches to assigning meaning to natural language. On the one hand, there was the compositional approach, an introduction to which can be found in [DWP81]. This approach is based on the notion that the meaning of a sentence is a function of the meaning of its parts ([GS11]), but is solely qualitative. On the other hand, there was the distributional approach ([Sch98]) which is quantitative but lacks a compositional structure by which to accurately combine meanings in complex sentences. Under the distributional approach, the meaning of words is encoded using vectors. These vectors are typically determined by examining the context of the word in a large corpus, and determining the co-occurrence with some designated set of basis words. Under the simplest model, the vector simply expresses how many times each of these basis words occurred in the word's context (subject to some normalisation). This distributional approach has found some real world applications ([GS11]).

More recently, Coecke et al. ([CSC10]) introduced a new approach which combines these two approaches. This *compositional distributional model of meaning* is both quantitative and compositional, and has been shown to outperform other models in computing sentence similarity ([GS11]). In the first section we will describe this model, largely following the treatment in [CSC10]. In section 2.2 we will discuss how relative pronouns can be accommodated in this model, following the treatment in [SCC14]. Finally, in section 2.3 we will briefly discuss entailment and ambiguity.

2.1 Foundations

Under the model by Coecke et al., computing the meaning of a sentence is a three step process:

1. First, grammatical types are assigned to each of the words in the sentence.
2. Then the grammatical types are used to construct a morphism in a compact closed category.
3. We then feed the meanings of the individual words into this morphism to calculate the meaning of the sentence.

We will discuss each of these steps, together with the required mathematical material in a separate subsection.

2.1.1 Pregroup grammars

For the grammatical types a *Pregroup grammar* is used. A detailed overview of Pregroup grammars can be found in [Lam08]. Here we will give a brief overview of the aspects relevant for our purposes. First, we define a Pregroup.

Definition 2.1 (Partially ordered monoid)**.** *A* partially ordered monoid $(P, \leq, \cdot, 1)$ *is a partially ordered set* $(P, \leq)$ *equipped with a monoid multiplication* $- \cdot -$ *with unit* 1*, where for* $p, q, r \in P$*, if* $p \leq q$ *then we have*

$$r \cdot p \leq r \cdot q \qquad \textit{and} \qquad p \cdot r \leq q \cdot r. \tag{2.1}$$

Definition 2.2 (Pregroup)**.** *A* Pregroup $(P, \leq, \cdot, 1, (-)^l, (-)^r)$ *is partially ordered monoid together with left and right adjoints* $(-)^l$ *and* $(-)^r$ *such that for every element* $p \in P$*:*

$$p^l \cdot p \leq 1 \leq p \cdot p^l \qquad \textit{and} \qquad p \cdot p^r \leq 1 \leq p^r \cdot p. \tag{2.2}$$

We refer to p^l *as the* left adjoint *of* p*, and* p^r *as the* right adjoint *of* p*.*

To construct a Pregroup grammar we simply fix a basic set of grammatical types, define a (possibly empty) set of inequalities between them, and then freely generate the Pregroup from these basic types (it has been shown that this free Pregroup exists and is unique). In the context of Pregroup grammars, the relation $\leq$ is typically written as $\to$, and a series of inequalities is referred to as a *reduction*.

As an example, we describe a very simple grammar for declarative transitive sentences. We introduce two basic grammatical types: n for nouns, and s for declarative statements. We will impose no inequalities between these basic types. The type s we will be our sentence type. That is to say, a well-formed sentence should reduce to s. To transitive verbs we will assign the type $n^r s n^l$. This can be interpreted as follows: a transitive verb takes a noun on its right and a noun on its left, and forms with these a declarative statement. Why this interpretation is valid will become clear in a moment. With these types fixed, we can now perform step 1 of the process above for our example sentence "John likes Mary":

John	likes	Mary
n	$n^r s n^l$	n

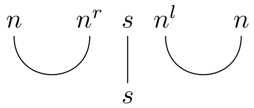
The type of the full sentence is simply the product of the types of the words, i.e. , $n \cdot n^r s n^l \cdot n$. This yields the following reduction:

$$n n^r s n^l n \to 1 s n^l n \to 1 s 1 \to s. \tag{2.3}$$

Thus the sentence reduces to s and is indeed well-formed.

2.1.2 Reductions in compact closed categories

The reduction above can be depicted graphically as follows:



(2.4)

This is obviously suggestive of a morphism in a compact closed category. To make this concrete, we take some compact closed category $\mathcal{C}$, and set up the following association between the Pregroup and $\mathcal{C}$:

- For every basic grammatical type p we pick an object P of $\mathcal{C}$.
- The monoid multiplication of the Pregroup is associated with the monoidal multiplication in $\mathcal{C}$. So if $p, q \in P$ correspond to $P, Q \in \mathrm{Obj}(\mathcal{C})$, then $p \cdot q$ corresponds to $P \otimes Q$.
- The left and right adjoints in the Pregroup correspond to duals in $\mathcal{C}$.[1]

[1] In [CSC10] each of the categories $\mathcal{C}$ that were used had self-dual objects, thus Pregroup elements and their adjoints were mapped to the same object. Here, we will discuss the more general case where objects in $\mathcal{C}$ might not be self-dual. In that case it is necessary to map adjoints to duals in order to make the diagrams work.

- The reductions $p^l \cdot p \to 1$ and $p \cdot p^r \to 1$ correspond to the cap, and $1 \to p \cdot p^l$ and $1 \to p^r \dot{p}$ to the cup.
- For each pre-imposed reduction $p \to q$ (i.e. those not generated by the free structure), we pick a morphism $P \to Q$ in $\mathcal{C}$ (where P and Q are the objects corresponding to p and q).

Note that we can regard the Pregroup as a monoidal category with left and right duals (i.e. an autonomous category), and if we do that the correspondence above is a functor preserving the monoidal and duality structure.[2] In what follows, we will take this perspective and use F to denote this functor.

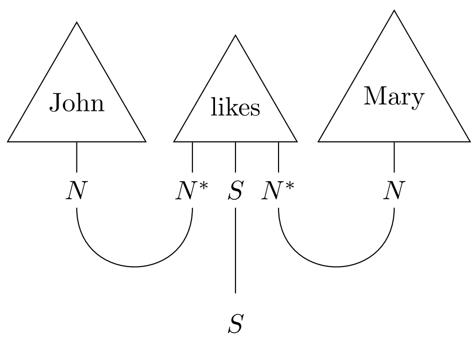
Applying this correspondence to the reduction in equation (2.3) yields exactly the morphism in $\mathcal{C}$ whose diagram is (2.4). The process of obtaining this morphism is exactly step 2 of the three step process described above.

2.1.3 From word meaning to sentence meaning

Next, for every word we pick a state in the object of $\mathcal{C}$ corresponding to its grammatical type. So in our example, for "likes" we pick a state in $F(n^r s n^l) = N^* \otimes S \otimes N^*$, where $N = F(n)$ and $S = F(s)$. This state is meant to encode the meaning of the individual word. We then apply the morphism corresponding to the grammatical reduction to these meaning-states. This is step 3 of the process above. In our example, if $|\text{John}\rangle, |\text{likes}\rangle, |\text{Mary}\rangle$ are the states representing the words in the sentence, then the meaning of the sentence is computed as:

$$F(nn^r sn^l n \leq s) \circ |\text{John}\rangle \otimes |\text{likes}\rangle \otimes |\text{Mary}\rangle \,. \tag{2.5}$$

Or, diagrammatically:



(2.6)

[2] In [CSC10] it is stated that the Pregroup is actually compact closed. This is in general not true under the usual definition of compact closed categories. Compact closure requires symmetry, but this does not hold for freely generated Pregroups.

The motivating example for $\mathcal{C}$ is **FVec**, where the states representing words are vectors. This is where the distributional aspect comes in; vectors encode the meaning of words just as in the distributional model, then these meanings are combined using a categorical morphism obtained from the compositional aspect of the model (the Pregroup grammar). However, the compositional distributional model does not require that $\mathcal{C} = \mathbf{FVec}$, it is much more general. Indeed, in [CSC10], an example using **FRel** is also worked out. In later work (e.g. [BCG$^+$16]) other compact closed categories have been substituted in order to refine the representation of meaning. This freedom is essential to what follows, since we will mainly be focussing on choosing a category that is well suited to achieving our goals. In the next chapter, we will pick this thread back up by introducing the notion of conceptual spaces and discussing how they relate to the choice of category.

2.2 Relative pronouns

In the previous section we have seen how we can combine the meanings of words using the unit and counit of a compact closed category. But of course, these categories often have much richer structure to exploit. In this section we will discuss how Frobenius algebras can be used to model relative pronouns.

Pronouns are difficult to represent in distributional models, because the context in which they occur has little bearing on their meaning; they occur in the context of a great variety of nouns and verbs. It is precisely their interaction with the other words in a sentence which is relevant. Thus the compositional distributional approach would seem like a natural approach to modelling these words. A method introduced by Sadrzadeh et al. ([SCC14]) deals with subject and object relative pronouns by using Frobenius algebras. In this section we will give an overview of the material in [SCC14].

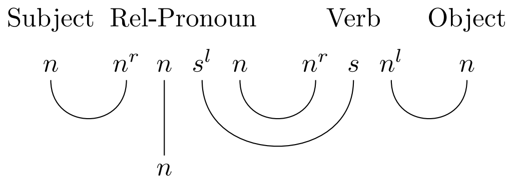
Sadrzadeh et al. write:

> The intuition behind the use of Frobenius algebras to model such cases is the following. In “book that John read”, the relative clause acts on the noun (modifies it) via the relative pronoun, which passes information from the clause to the noun. The relative clause is then discarded, and the modified noun is returned. Frobenius algebras provide the machinery for all of these operations.

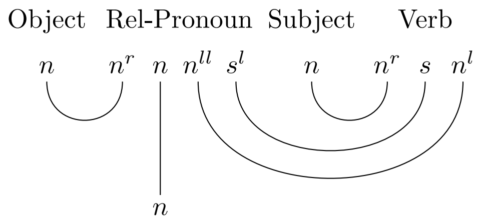
Sadrzadeh et al. assign the following Pregroup types to the subject and object relative pronouns:

$$n^r n s^l n \text{ (subject)} \qquad\qquad n^r n n^{ll} s^l \text{ (object)} \tag{2.7}$$

Using these types in a simple noun phrase, we obtain the following reductions:



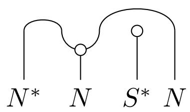
(2.8)

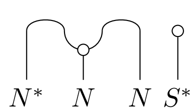


(2.9)

Note that the reduction is to n, not to s. Thus the phrase is not a full well-formed sentence, but instead a noun-phrase.

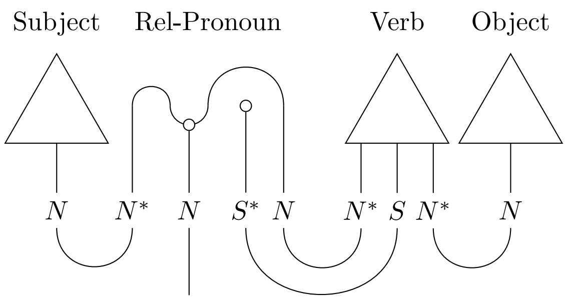
To these words we assign the following states:[3]

Subj: 

Obj: 

(2.10)

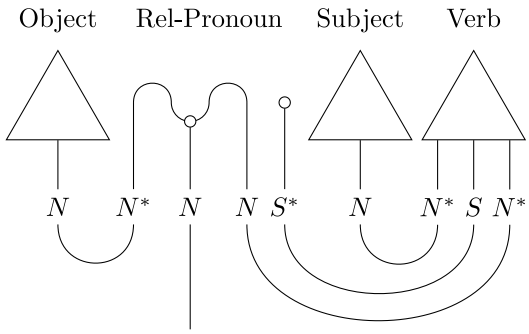
When used in a simple noun phrase the subject relative pronoun yields the following diagram:



(2.11)

[3]Again, unlike in [SCC14], we do not assume that objects in $\mathcal{C}$ are self-dual.

And for the object relative pronoun:



(2.12)

For further details and implementations using toy models, please refer to [SCC14].

2.3 Entailment and Ambiguity

While the basic model of $\mathcal{C} = \mathbf{FVec}$ based on co-occurrence is attractive for its simplicity, and has shown promise in both the compositional distributional model and traditional distributional models, it lacks the structure to capture some finer aspects of meaning. This can be remedied by choosing a different model $\mathcal{C}$ which has a richer structure. This comes at the price of simplicity, and thus likely makes it much harder to use an automated process to determine the states representing the meaning of words. In the next chapter we will take a closer look at the choice of $\mathcal{C}$ when we discuss conceptual spaces, but for now we mention two aspects of meaning which are poorly represented by vectors, and the solutions which have been proposed for them in the literature: lexical entailment and ambiguity.

Entailment (or hyponymy) refers to the notion that a word can be an instance of another word. For example, 'pigeon' is an instance of 'bird', which is an instance of 'animal'. We say that 'pigeon' *entails* 'bird' (or that 'pigeon' is a hyponym of 'bird', and 'bird' a hypernym of 'pigeon'). The vector model does not capture this notion; while the vector model can capture the fact that 'pigeon' and 'bird' are closely associated, it's not clear how to differentiate this relation from the reverse relation (hyponymy versus hypernymy), or from the relation between 'pigeon' and 'hawk'. Bankova et al. [BCLM16] describe a model based on density matrices that does capture this notion. Similar approaches were described in [BKS15] and [BSC15]. Density matrices are a concept from quantum mechanics, where they are used to represent partial knowledge of a system, and can be described in a categorical manner as *completely positive maps*. The model by Bankova et al. further allows for graded entailment, thus capturing a notion of *entailment strength*.

Similarly, vectors do not represent ambiguity well. The vector model can capture the fact that the word ‘queen’ is closely associated with ‘monarch’, ‘bee’, ‘rock band’ and ‘chess’, but it does not capture the fact that each of these associations stems from a separate meaning of ‘queen’, and that when the word is used in a sentence, generally only one of these meanings is intended. Piedeleu [Pie14] introduced a model to capture this notion, again based on density matrices. Ashoush and Coecke [AC16] combined these two approaches by using dual density operators to simultaneously represent ambiguity and entailment.

Our model, which will be introduced in subsequent chapters, naturally captures both entailment and ambiguity, but is unable to capture any grading in these notions.

Chapter 3

Conceptual spaces

While the compositional distributional model was first applied together with meaning representations drawn from distributional models and represented in $\mathcal{C} = \mathbf{FVec}$, the model is independent of the exact choice of meaning space. In the previous section we already touched upon how moving from vectors to single or dual density operators allows us to encode entailment and ambiguity. In this chapter we will take a step back and consider what we would expect from a meaning space from the point of view of cognitive science. The hope is that this will lead to a more efficient representation of meaning which naturally encodes many aspects of meaning that would otherwise have to be imposed on the meaning space in an ad-hoc manner. Such a space would be interesting for natural language processing applications, but might also be mathematically interesting in its own right. Moreover, studying such spaces and their requirements could be an interesting avenue for research in cognitive science.

The exploration of these topics has been driven by Gärdenfors, who published two books ([Gär00, Gär14]) on this subject. In these books, he introduced the concept of *conceptual spaces*, and put these in a (limited) mathematical context. In this chapter we will summarise the relevant information in these works. In subsequent chapters, we will apply these concepts in a mathematically rigorous setting.

Gärdenfors writes (section 2.1 of [Gär14]):

> Conceptual spaces are constructed out of *quality dimensions*. Examples are pitch, temperature, weight, size, and force. The primary role of the dimensions is to represent various “qualities” of objects in different domains. Some dimensions come in bundles – what I call *domains* – for example,

space (dimensions of height, width, and depth); color (hue, saturation, and brightness); taste (salt, bitter, sweet, and sour, and maybe a fifth dimension); emotion (arousal and value; [...]); and shape (dimensions not well known; [...]).

[...]

The notion of a dimension should be understood literally. It is assumed that each dimension is endowed with a topological or geometric structure.

3.1 Convexity

The geometric structure that Gärdenfors exploits is largely based on notions of *distance* (or *similarity*), and *betweenness*, and these are the notions that we will restrict ourselves to in the following chapters.

A central thesis by Gärdenfors is the following (section 2.2 of [Gär14], and also section 3.5 of [Gär00]):

> *Thesis about properties:* A *property* is a convex region in some domain.

And (section 2.2 of [Gär14]):

> Properties, as characterized by the thesis, form a special case of *concepts*. This distinction is defined by saying that a property is based on a single domain, while a concept is based on one *or more* domains.
>
> [...]
>
> Concepts are not just bundles of properties. The representation of an object category that I present in chapter 6 also includes an account of the *correlations* between the regions from different domains that are associated with the category. For example, the concept *apple* involves a very strong (positive) correlation between the sweetness in the taste domain and the sugar content in the nutrition domain and a weaker correlation between the color red and a sweet taste.

Gärdenfors defines convexity in terms of betweenness, and betweenness as a ternary relation B satisfying (section 1.6.1 of [Gär00]):

B0: If $B(a, b, c)$ then a, b, c are distinct points.

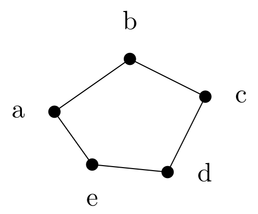


Figure 3.1: Example of a graph violating betweenness requirement B3, if the metric is hops in the graph, and betweenness is defined in terms of the metric. Based on figure 1.9 from [Gär00].

B1: If $B(a, b, c)$ then $B(c, b, a)$. In words: "If b is between a and c, then b is between c and a."

B2: If $B(a, b, c)$ then not $B(b, a, c)$. "If b is between a and c, then a is not between c and b."

B3: If $B(a, b, c$ and $B(b, c, d)$, then $B(a, b, d)$. "If b is between a and c and c is between b and d, then b is between a and d."

B4: If $B(a, b, d)$ and $B(b, c, d)$, then $B(a, b, c)$. "If b is between a and d and c is between b and d, then b is between a and c."

Gärdenfors then defines the usual notion of a metric, and notes that betweenness B can be defined in terms of a metric d as $B(a, b, c) \iff d(a, c) = d(a, b) + d(b, c)$. Gärdenfors also notes that such a notion of betweenness satisfies B1, B2 and B4, but not necessarily B3 (see figure 3.1). In the following chapters, we will define a metric for our model and then define convexity in terms of this metric. Thus, to preserve generality, we are forced to drop B3 from the list of requirements for betweenness. B0 can always be recovered by a trivial modification of B, thus for the sake of simplicity we will also drop this requirement.

A convex subset is then defined as (definition 3.3 in [Gär00]):

Definition 3.1 (Convex subset of a conceptual space)**.** *A subset C of a conceptual space S is said to be* convex *if, for all points x and y in C, all points between x and y are also in C.*

Gärdenfors argues that convexity plays a crucial part in learning. As an example of how learning based on the convexity assumption might work, he describes how a set of concepts can be created from prototypes (or first examples) of those concepts by creating a Voronoi tessellation, and then slowly adjusting this tessellation as new

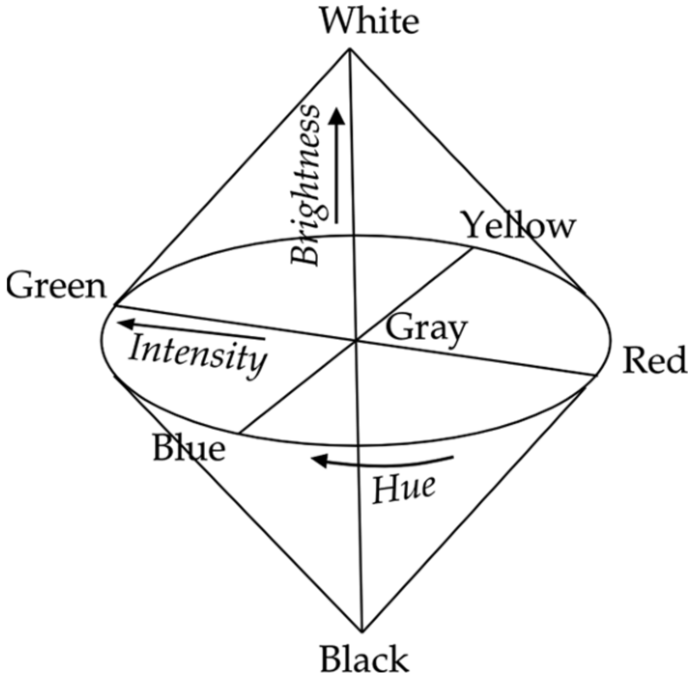


Figure 3.2: The colour spindle. Source: figure 2.1 from [Gär14]

examples are observed, while still maintaining the convexity of the concepts. For details, refer to section 2.3 of [Gär14].

3.2 Example: colour

In both [Gär00] and [Gär14], Gärdenfors uses the example of the colour domain to clarify the notions discussed above. This example is particularly helpful because the proposed domain matches very well with our experience of colour, and with the representation of colour in digital imagery. The quality dimensions in this domain are *hue*, *saturation* and *brightness*. Hue is considered to be a polar dimension, in the sense that it 'wraps around', while saturation and brightness are linear. Figure 3.2 shows the geometric intuition behind these dimensions.

This example also shows how a single domain may admit multiple choices for the notion of betweenness (or, equivalently, distance), and that the specific choice can be crucial to obtaining a natural notion of properties. In figure 3.3 two different notions of betweenness are shown. The linear notion, shown on the left, leads to a non-convex region for the property 'red', thus violating the convexity thesis. This

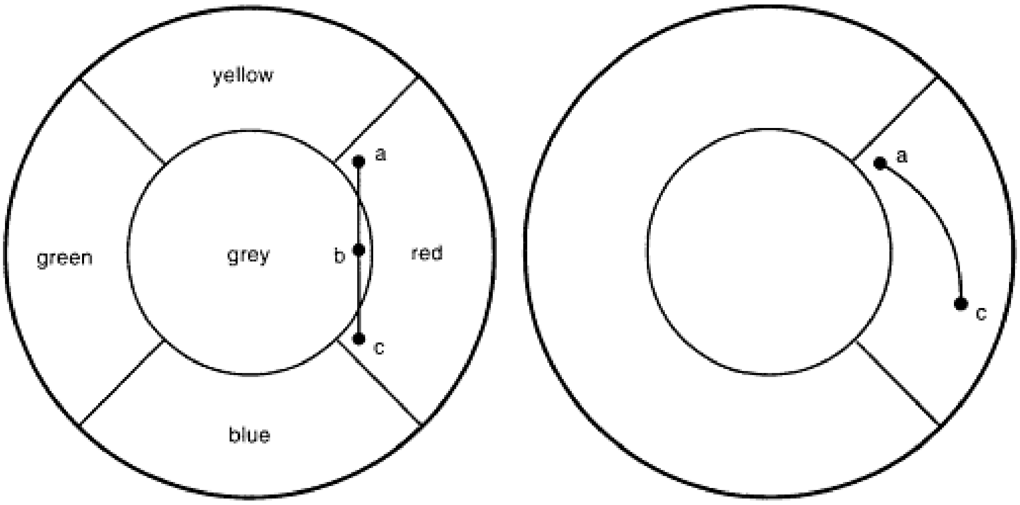


Figure 3.3: A horizontal (hue-saturation) slice of the colour spindle. Left: a linear notion of betweenness does not match with the natural interpretation of colour properties. Right: this can be remedied by using a polar notion of betweenness for the hue dimension. Source: figures 3.5 and 3.6 from [Gär00].

can be remedied by choosing a different notion of betweenness, one that respects the polar nature of the hue dimension.

Such a notion of betweenness could arise from the following metric:

$$d_{\text{colour}}((\theta_1, s_1, v_1), (\theta_2, s_2, v_2)) = \sqrt{d_\theta(\theta_1, \theta_2)^2 + |s_1 - s_2|^2 + |v_1 - v_2|^2}, \tag{3.1}$$

where θ, s, v are the coordinates for hue, saturation and brightness, respectively. The function d_θ is the angular distance function, which gives the difference between two angles. Explicitly:

$$d_\theta(\theta_1, \theta_2) = \min\{|\theta_1 - \theta_2|, |\theta_1 - \theta_2 + 2\pi|, |\theta_1 - \theta_2 - 2\pi\}. \tag{3.2}$$

Part II

Building a hierarchical category for shapes

Chapter 4

Building hierarchical conceptual spaces

In this chapter we will develop the tools for building hierarchical conceptual spaces based on trees. Then, in the next chapter, we define a category for such conceptual spaces and show how it fits within the distributional compositional model. Finally, in chapter 6, we specialize this category to represent shapes using a variant of the Marr-Nishihara model.

To motivate this construction, we first discuss Marr and Nishihara's model in abstract terms in section 4.1. Then we give a mathematically rigorous definition of trees in section 4.2. In order to discuss similarity and convexity for such trees, we define a generalisation of metrics and convexity in section 4.3. Finally, we apply this generalisation to trees in the last section of this chapter.

4.1 Marr and Nishihara's model

In his work on conceptual spaces, Gärdenfors argues that shape is an important dimension for understanding language and learning (section 3.2.4 of [Gär14]):

> During the period between eighteen and twenty-four months, children undergo what is called a naming spurt, acquiring a substantial number of nouns for representing objects. Evidence suggests that, during this period, they also learn to extract the general shape of objects, and this abstraction improves object category learning (Son, Smith, & Goldstone, 2008; L. Smith, 2009). How the object category space develops in children

is still not well known. Some cues can be obtained from children's ability to learn nonsense words for new things (Bloom, 2000; L. Smith, 2009). There seems to be a shape bias; that is, the shape of objects seems to be the most important property in determining category membership for small children (Landau, Smith, & Jones, 1998; Smith & Samuelson, 2006).

Yet it is not clear how to represent shape in the compositional distributional model.

A first idea for representing shapes might be to simply encode the physical embodiment of a shape as a subset S of euclidean space $\mathbb{R}^3$. Such an approach has numerous advantages:

- It is conceptually very simple.
- It has a clear relation to the physical world and to sensory input.
- Constructing such representations for real-world shapes seems straight forward and might even be completely automated through techniques such as 3D scanning.

However, it also has some clear disadvantages:

- For real-world, non-mathematical objects, it is not clear with how much detail shapes should be encoded. Take a spider, for example, should its shape simply be represented as a sphere with eight legs sticking out of it? Or should we painstakingly encode the exact shape of its body and legs, the details of its eyes, its fangs, all the hairs on its legs, etc.? How do we deal with two spiders with a slightly different shape? How do we deal with entailment in the context of shapes (for example, the general shape of an insect versus the specific shape of an ant)?
- It is not clear how to define similarity and betweenness.
- When describing the shape of an object to someone, a speaker would be unlikely to describe the exact physical embodiment of that object. Instead, the speaker would likely compare the object to another object with a similar shape, or describe it as a composition of simpler geometric shapes. Thus the physical embodiment model does not mesh well with verbal descriptions of shapes. This also suggests that it doesn't mesh well with our mental representation of shapes.
- The physical embodiment representation has no concept of whole-part relationships. Thus while the model might be able to represent the shape of a spider

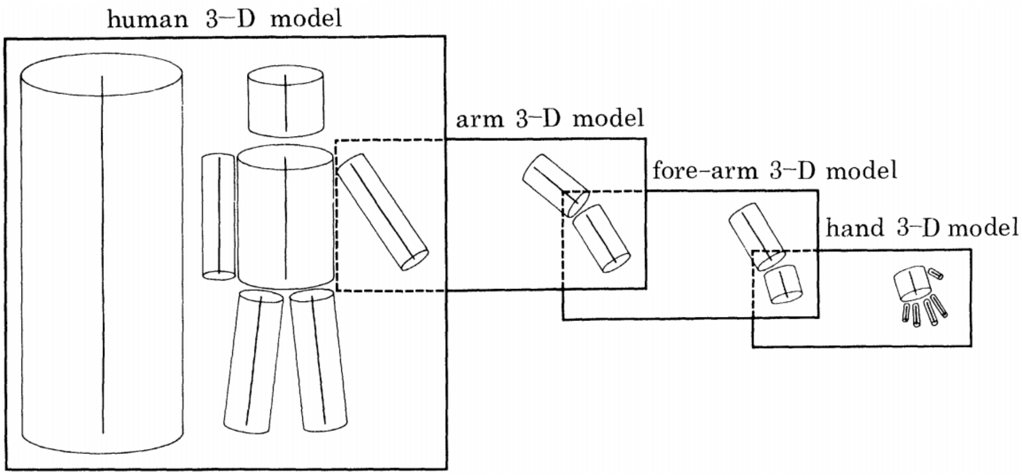


Figure 4.1: This figure shows what (part of) a tree representing a human would look like under the Marr-Nishihara model. Source: figure 3 of [MN78].

and the shape of a leg, it is unable to model the fact that a leg is a distinct part of a spider.

- The representation is purely static; it doesn't encode any information on how an object might move or be deformed. While it might seem strange to want such dynamic information to be included in the shape domain, this does seem defensible when you consider the fact that we think of humans having a certain 'shape', even though a person can deform and move their body in many different ways.

In section 6.3.1 of [Gär14], Gärdenfors suggests using a model developed by Marr and Nishihara [MN78] to represent shapes. In this model a shape is described by a tree, where each node in the tree describes a cylinder with certain dimensions, an orientation and a position. Each level describes the shape in more detail, with the children of a node representing the parts that together make up that node.

In figure 4.1 we see a depiction of what a tree representing the shape of the human body could look like. The root node simply describes a single vertical cylinder. The children of the root node describe the main parts that make up the human body: a torso, a head, two arms and two legs. Each of these parts are represented by a cylinder with a certain size, position and orientation relative to the root cylinder. Moreover, each of these parts is itself described as being made up of smaller parts, going on in a recursive fashion thus giving rise to the tree structure. For example, the arm is made up of two parts: the upper arm and the lower arm. Then the lower arm is again made up of two parts: the forearm and the hand. The hand, in turn, is again made up of several parts: the palm of the hand, a thumb and four fingers.

Contrasting this model to the simple model of physical embodiment, we see that it has many advantages over that model:

- The level of detail for a shape is explicitly encoded in a Marr-Nishihara tree; higher trees encode a higher level of detail in their deeper levels. Thus it is possible to compare two shape representations even if they do not have the same amount of detail: we can simply compare deeper and deeper levels of both trees until we exhaust the detail in one of the trees. This also allows us to deal with entailment: shallower trees represent general concepts, and deeper trees represent instances of such concepts. So an insect might have a certain general shape, represented by a shallow tree, and then specific insects, such as ants, are represented by deeper trees that agree with the insect tree in the shallower levels. This notion will be explored further in the next section when we define the subtree relation, and again in chapter 5.

- While similarity and betweenness are not clear cut, the hierarchical structure of Marr-Nishihara trees does seem to make defining such concepts more tractable. And in fact, we will develop mathematically precise formulations of these notions in the remainder of this chapter.

- This model seems to mesh well with how shapes are verbally described; a shape is represented by a (recursive) composition of primitive shapes. Moreover, Marr and Nishihara specifically constructed this model to (attempt to) mirror our mental model of shapes [MN78].

- Whole-part relationships are explicitly encoded in Marr-Nishihara trees.

It also has some disadvantages:

- This model is much more complicated than the physical embodiment model. This makes it harder to argue about.

- It's not clear how the Marr-Nishihara tree representing the shape of an arbitrary object should be determined. Marr and Nishihara do touch upon this problem in [MN78], and suggest that it might inferred from a picture of the object. It is conceivable that this process could be automated by using a database of captioned images, though this would certainly be very challenging.

- Marr and Nishihara developed this model primarily to represent the shapes of animals. It is not clear that the model generalises well to non-biological shapes. In particular, using cylinders as the primitive may not be well suited to describing the many somewhat cubic man-made objects that fill our lives. One way of solving this problem is by choosing a different primitive or using several different primitives together.
- The model is inherently three dimensional, even though certain shapes are clearly not (circles, for example), and even some real world objects (such as a piece of paper or a string) are thought of as having a one or two dimensional shape. Again, this problem can be solved by choosing one or two dimensional primitives.
- Like the physical embodiment model, this model is purely static.

In section 6.3.2 of [Gär14], Gärdenfors also mentions a more intricate model which does explicitly include information about how an object can be deformed. However, with an appropriate choice of metric and betweenness, this information can simply be encoded as a convex region in the space of Marr-Nishihara trees.

In light of the above, the Marr-Nishihara model seems like a good starting point for building a conceptual space for shapes. With that in mind, we will spend the remainder of this chapter on developing the mathematical tools that we will need in order to reason about trees.

4.2 Defining trees

We now give a mathematically rigorous definition for trees as we will use them in the remainder of this work. In the first section we will define a structure which allows us to label children of a node in a tree using consecutive natural numbers. Then we define the actual tree structure, parametrising over a property space P: nodes in the tree will carry an element of P. This element represents the properties of the node. Finally, we define an order on trees.

4.2.1 Finite labelled subsets

We define a structure akin to sets, but with the elements being labelled, thus permitting duplicates. We use bounded natural numbers as the labels to enforce finiteness.

This also gives us an ordering of the labels.

Definition 4.1 (Finite labelled subset)**.** *Let X be a set and let S be a function*

$$S : \mathbb{N} \to X \cup \{\varepsilon\}, \tag{4.1}$$

Where, ε is some designated symbol such that $\varepsilon \notin X$. We say that S is a finite labelled subset *of X, and write $S \sqsubseteq X$, if S satisfies:*

$$\exists n_0 \in \mathbb{N}\colon \big(\forall n \in \mathbb{N}, n < n_0\colon\ S(n) \neq \varepsilon\big) \wedge \big(\forall n \in \mathbb{N}, n \geq n_0\colon\ S(n) = \varepsilon\big). \tag{4.2}$$

We refer to n_0 as the size *of S, and write $\#(S) = n_0$. We write $x \in S$ if there is some $n \in \mathbb{N}$ such that $S(n) = x$. Note that $\{x \,|\, x \in S\}$ is a finite set. In particular, its cardinality is bounded by n_0. We write $\mathbb{P}(X)$ for the set of all finite labelled subsets of X.*

Note that finite labelled subset could equivalently be called a finite sequence over a set.

Definition 4.2 (Empty labelled set)**.** *For any set X, we use the symbol $\emptyset$ to denote the* empty labelled set*, defined as:*

$$\begin{aligned} &\emptyset : \mathbb{N} \to X, \\ &\forall n \in \mathbb{N}\colon\ \emptyset(n) = \varepsilon. \end{aligned} \tag{4.3}$$

Note that $\emptyset \sqsubseteq X$.

Definition 4.3 (Set builder notation for finite labelled subsets)**.** *Let X be a set, $f : \mathbb{N} \to X$ a function and n_0 a natural number. Then we write*

$$S = \{n \mapsto f(n) \,|n \leq n_0\} \tag{4.4}$$

for the finite labelled subset of X defined by

$$S(n) = \begin{cases} f(n) & \text{if } n < n_0, \\ \varepsilon & \text{otherwise.} \end{cases} \tag{4.5}$$

Note that $S \sqsubseteq X$ and $\#(S) = n_0$. Similarly, for $x \in X$ the singleton labelled set of x, written as

$$S = \{0 \mapsto x\} \tag{4.6}$$

is defined by

$$S(n) = \begin{cases} x & \text{if } n = 0, \\ \varepsilon & \text{otherwise.} \end{cases} \tag{4.7}$$

And a useful transformation on such structures:

Definition 4.4 (Reduction function)**.** *The* reduction function ρ *transforms a finite labelled subset S of X into a different finite labelled subset of X defined by:*

$$\rho(S)(n) = S(n+1). \tag{4.8}$$

Note that if $\#(S) \geq 1$, then $\#(\rho(S)) = \#(S) - 1$, and otherwise $\#(\rho(S)) = 0$.

4.2.2 Finite trees

We are now ready to formally define trees. We will restrict ourselves to finite trees. In order to rigorously define finite trees, we first define sets of non-empty trees with finite width and an upper bound on their height, and then define the full set of trees as the union of these sets, together with the empty tree.

Definition 4.5. *Let P be a set. Then the set* $\mathrm{Tree}_1(P)$ *of non-empty finite trees of height at most* 1*, having properties in P, is defined as:*

$$\mathrm{FTree}_1(P) = \big\{(p, \emptyset) \,|\, p \in P\big\}. \tag{4.9}$$

Definition 4.6. *Let P be a set, and $n \geq 2$ an integer. Then the set* $\mathrm{Tree}_n(P)$ *of non-empty finite trees of height at most n, having properties in P, is defined as:*

$$\mathrm{FTree}_n(P) = \mathrm{FTree}_{n-1}(P) \cup \big\{(p, C) \,|\, p \in P,\, C \sqsubseteq \mathrm{FTree}_{n-1}(P)\big\}. \tag{4.10}$$

Definition 4.7 (Finite trees)**.** *Let P be a set. Then* the set $\mathrm{FTree}(P)$ of finite trees having properties in P *is defined as:*

$$\mathrm{FTree}(P) = \{\varepsilon\} \cup \bigcup_{n=1}^{\infty} \mathrm{Tree}_n(P). \tag{4.11}$$

We refer to ε as the empty tree*.*

Depending on the perspective we want to take, we use the terms node and tree interchangeably, with the former emphasising the node properties of the tree, and the latter emphasising the structure of the tree.

We make a few auxiliary definitions

Definition 4.8 (Properties and children)**.** *For a non-empty tree (p, C), we refer to p as the* properties of the root node*, and write* $\mathrm{Prop}\big((p, C)\big) = p$*, and we refer to C as the* set of children of the root node*, and write* $\mathrm{Children}\big((p, C)\big) = C$*.*

Definition 4.9 (Height). *Let P be a set. We define a height function* height : $\mathrm{FTree}(P) \to \mathbb{N}$ *as follows:*

- *The height of the empty tree is 0:*

$$\mathrm{height}(\varepsilon) = 0. \tag{4.12}$$

- *The height of a tree with no children is 1:*

$$\mathrm{height}\big((p, \emptyset)\big) = 1, \tag{4.13}$$

  for all $p \in P$.

- *Otherwise, the height of a tree is 1 more than the maximum height of its children:*

$$\mathrm{height}\big((p, C)\big) = 1 + \max_{t \in C} \mathrm{height}(t). \tag{4.14}$$

Definition 4.10 (Size). *Let P be a set. We define a size function* size : $\mathrm{FTree}(P) \to \mathbb{N}$ *as follows:*

- *The size of the empty tree is 0:*

$$\mathrm{size}(\varepsilon) = 0. \tag{4.15}$$

- *Otherwise, the size of a tree is 1 plus the sum of the size of its children:*

$$\mathrm{size}\big((p, C)\big) = 1 + \sum_{t \in C} \mathrm{size}(t). \tag{4.16}$$

Definition 4.11 (Node path and depth). *A* node path *p is a finite labelled subset of $\mathbb{N}$. Let P be a set and $t \in \mathrm{FTree}(P)$ a tree. We write $t(p)$ for the* node indexed by p in t, *given by:*

- *If $\#(p) = 0$, then $t(p) = t$.*

- *Otherwise, if $t = \varepsilon$, then $t(p) = \varepsilon$.*

- *Otherwise, t is of the form (p, C) and $t(p)$ is given by the node indexed by $\rho(p)$ in $C(p(0))$:*

$$t(p) = C(p(0))(\rho(p)). \tag{4.17}$$

The depth *of a node n in a tree t is the size of the smallest node path that indexes it.*

Definition 4.12 (Level)**.** *Let P be a set. We define a level-size function* $\text{ls} : \mathbb{N} \times \text{FTree}(P) \to \mathbb{N}$ *as follows:*

- *For any n, the size of level n of the empty tree is 0:*

$$\text{ls}(n, \varepsilon) = 0. \tag{4.18}$$

- *The size of level 0 of a non-empty tree is 1:*

$$\text{ls}\big(0, (p, C)\big) = 1. \tag{4.19}$$

- *Otherwise, the size of level n of a non-empty tree is the sum of the sizes of levels $n-1$ of the children of the tree:*

$$\text{ls}\big(n, (p, C)\big) = \sum_{t \in C} \text{ls}(n-1, t). \tag{4.20}$$

We define a level function $\text{level} : \mathbb{N} \times \text{FTree}(P) \to \mathbb{P}(\text{FTree}(P))$ *as follows:*

- *For any n, level n of the empty tree is the empty labelled set:*

$$\text{level}(n, \varepsilon) = \emptyset. \tag{4.21}$$

- *Level 0 of a non-empty tree is a singleton containing that tree:*

$$\text{level}(0, t) = \{0 \mapsto t\}. \tag{4.22}$$

- *Otherwise, level n of a non-empty tree (p, C) is a labelled set of trees S where for $m \in \mathbb{N}$, $S(m)$ is given by:*
  - *If there exists an $i \in \mathbb{N}$ such that $l = \sum_{j=0}^{i-1} \text{ls}(n-1, C(j))$ satisfies $l \leq m < l + \text{ls}(n-1, C(i))$:*

$$S(m) = \text{level}(n-1, C(i)). \tag{4.23}$$

  - *Otherwise:*

$$S(m) = \varepsilon. \tag{4.24}$$

Intuitively, the level of a tree is the labelled set of all nodes at a certain depth, where the labelling is from left to right. Note that $\#(\text{level}(n, t)) = \text{ls}(n, t)$*.*

Definition 4.13 (Root, leaf and internal node)**.** *The* root *of a tree is the top-level node of the tree (which is equivalent to the entire tree). Note that the root is exactly the node indexed by $\emptyset$. A* leaf *is a node in a tree with no children (i.e. of the form $(p, \emptyset)$). An* internal node *is a node that is not a leaf.*

4.2.3 Ordering trees

It will be useful to be able to compare trees. To this end, we define a subtree relation.

Definition 4.14 (Subtree)**.** *Given two trees $s, t \in \mathrm{FTree}(P)$ for some set P, we say that s is a* subtree *of t, and write $s \leq t$ if one of the following holds:*

(ST.1) *$s = \varepsilon$; or*

(ST.2) *s is of the form $(p, \emptyset)$ and t is of the form (q, B), where $p = q$; or*

(ST.3) *s is of the form (p, A) and t is of the form (q, B), where $p = q$, $\#(A) = \#(B)$, and*

$$\forall n \in N\colon\ A(n) \leq B(n). \tag{4.25}$$

Intuitively, this means that when $s \leq t$, the tree s can be obtained from t by removing all of the children of some of the nodes in t (or discarding the tree entirely). Note that this is different from the usual graph-theoretic definition of subtrees. The motivation for this definition within the context of conceptual spaces is that we want $s \leq t$ to roughly mean 's is a less detailed (or more general) version of t'. In this context, the empty tree ε is considered the most general. For non-empty trees we allow the structure of t to continue where s has a leaf. On the other hand, where s has an internal node, this node must exactly match the corresponding node in t, except that the children of the node in t may be more detailed. Thus we explicitly disallow corresponding internal nodes to have a different number of children.

So, if we represents concepts like 'insect', 'ant', and 'spider' by Marr-Nishihara trees, then we would expect to have the following relationships:

- $|\mathrm{insect}\rangle \leq |\mathrm{ant}\rangle$, because 'insect' is more general than 'ant'. This is witnessed by the fact that the Marr-Nishihara tree of 'ant' is more detailed than that of 'insect'. The tree for 'insect' would be quite shallow, representing only the overall structure of three body parts, one pair of antennae and six legs. The tree for 'ant' would be much deeper, with the top levels having the same structure, but then going on to deeper levels describing the exact morphology of ants.
- $|\mathrm{insect}\rangle \not\leq |\mathrm{spider}\rangle$, because 'insect' is not more general than 'spider': a spider is not an insect. This is witnessed, for example, by the fact a spider has eight legs, while an insect has six. Thus the Marr-Nishihara tree of 'insect' cannot be extended to the tree of 'spider'; at some point we would have to add additional children (representing the extra legs) to an existing node.

- $|\text{ant}\rangle \not\leq |\text{spider}\rangle$, because neither is a more general version of the other. Both these concepts would be represented by fairly deep trees, which already disagree in the higher levels.

It should be clear that this relation is related to the concept of entailment. This will be discussed further in section 5.3.

We note that the relation defined above is a partial order:

Theorem 4.15. *For any set P, the relation $\leq \subseteq \mathrm{FTree}(P) \times \mathrm{FTree}(P)$ as defined in definition 4.14 is a partial order.*

Proof. We prove reflexivity, transitivity and anti-symmetry by induction on the maximum height h of the trees.

For $h = 0$ the only possible tree is ε and all the required properties hold trivially.

Now let $h \geq 1$ and assume that reflexivity, transitivity and anti-symmetry of $\leq$ hold for all trees of height at most $h - 1$. Note that for any tree of height at most h, the children of the root will have height at most $h - 1$, thus by the induction hypothesis we can apply reflexivity, transitivity and anti-symmetry to such trees.

For reflexivity, let t be a tree of height at most h. If $t = \varepsilon$, we apply the previous case. Otherwise, t can be written as (p, C) for some $p \in P$ and set of children C. We then see that case (ST.3) of definition 4.14 applies: for each n we have $C(n) \leq C(n)$ by the induction hypothesis.

For transitivity, let r, s, t be trees of height at most h, and assume that $r \leq s$ and $s \leq t$.

- If $r = \varepsilon$ then we can immediately apply case (ST.1) to $r \leq t$.
- Otherwise, if r is of the form $(p, \emptyset)$, then s is of the form (p, B). Thus either case (ST.2) or case (ST.3) applies to $s \leq t$ and we conclude that t is of the form (p, C). Thus case (ST.2) applies to $r \leq t$.
- Finally, if r is of the form (p, A) for non-empty A, then s is of the form (p, B) for non-empty B. Thus case (ST.3) applies to $s \leq t$, and t is of the form (p, C) for non-empty C. We have

$$\forall n \in N\colon\ A(n) \leq B(n) \text{ and } B(n) \leq C(n). \tag{4.26}$$

By the induction hypothesis, we can use transitivity on the trees in the above equation. This gives us:

$$\forall n \in N\colon\ A(n) \leq C(n) \tag{4.27}$$

We also have $\#(A) = \#(B) = \#(C)$. Thus case (ST.3) applies $r \leq t$.

For anti-symmetry, let s, t be trees of height at most h, and assume that $s \leq t$ and $t \leq s$.

- If $s = \varepsilon$ then $t \leq s$ implies $t = \varepsilon$ through case (ST.1), and thus $s = t$.
- If s is of the form $(p, \emptyset)$ then case (ST.2) applies and thus t is of the form (p, B). Then $t \leq s$ implies that either case (ST.2) applies and $B = \emptyset$, in which case we're done, or case (ST.3) applies and $\#(B) = \#(\emptyset) = 0$, which again implies that $B = \emptyset$.
- If s is of the form (p, A) for non-empty A then case (ST.3) applies to $s \leq t$ and thus t is of the form (p, B) for non-empty B. Thus case (ST.3) must also apply to $t \leq s$. So we have:

$$\forall n \in \mathbb{N}\colon\ A(n) \leq B(n) \text{ and } B(n) \leq A(n). \tag{4.28}$$

Applying the induction hypothesis we conclude

$$\forall n \in \mathbb{N}\colon\ A(n) = B(n). \tag{4.29}$$

Thus $A = B$ and $s = t$.

We conclude that by the principle of induction, reflexivity, transitivity and anti-symmetry hold for all trees of finite height. Thus $\leq$ is a partial order on $\mathrm{FTree}(P)$. □

We can also define a meet:

Definition 4.16 (Meet of trees)**.** *Let P be a set. We define a meet function $\wedge$: $\mathrm{FTree}(P) \times \mathrm{FTree}(P) \to \mathrm{FTree}(P)$. For $s, t \in \mathrm{FTree}(P)$, the meet $s \wedge t$ is given by:*

- *If $s = \varepsilon$ or $t = \varepsilon$, then $s \wedge t = \varepsilon$.*
- *If $s = (p, A)$ and $t = (q, B)$, and $p \neq q$, then $s \wedge t = \varepsilon$.*
- *If $s = (p, A)$ and $t = (q, B)$, $p = q$, $\#(A) = \#(B)$ and*

$$\forall n \in \mathbb{N}, n < \#(A)\colon\ A(n) \wedge B(n) \neq \varepsilon, \tag{4.30}$$

then

$$s \wedge t = \Big(p, \big\{n \mapsto A(n) \wedge B(n) \,|\, n \leq \#(A)\big\}\Big) \tag{4.31}$$

- *Otherwise,* $s = (p, A)$ *and* $t = (q, B)$ *with* $p = q$*, and* $s \wedge t = (p, \emptyset)$*.*

Theorem 4.17. *The meet defined above is indeed an order-theoretic meet, and thus* $\leq$ *gives rise to a meet semi-lattice.*

Proof. It is obvious from the definition of the meet that for any trees $s, t \in \text{FTree}(P)$, we have $m = s \wedge t \leq s, t$. To show that m is the maximum among such elements, we proceed by induction on the maximum height h of m.

If $h = 0$, then $m = \varepsilon$. Thus either $s = \varepsilon$, $t = \varepsilon$, or $s = (p, A)$ and $t = (p, Q)$ with $p \neq q$. In each of these cases the only subtree of both s and t is ε, thus the result holds.

Now let $h \geq 1$ and assume that if $s \wedge t$ has height at most $h - 1$, then it is the maximum simultaneous subtree of s and t. If $m = \varepsilon$ the previous case applies. Otherwise, m is of the form (p, A), and thus s is of the form (p, B) and t is of the form (p, C). If $A = \emptyset$ then one of the following must be true:

- $\#(B) \neq \#(C)$. In this case, the maximum simultaneous subtree of s and t is clearly $(p, \emptyset) = m$, so the result holds.
- $\#(B) = \#(C)$, but there is an $n \in N$ with $n < \#(B)$ such that $B(n) \wedge C(n) = \varepsilon$. This implies that $B(n) = \varepsilon$, $C(n) = \varepsilon$, or $\text{Prop}(B(n)) \neq \text{Prop}(C(n))$. The first two cases contradict the fact that $\#(B) = \#(C)$, so we can assume the latter case. This implies that the maximum simultaneous subtree of $B(n)$ and $C(n)$ is ε. Since ε is not a valid child, this forces $m = (p, \emptyset)$, so the result holds.

Otherwise, A is non-empty. This means that $\#(B) = \#(C)$ and that for each $n \leq \#(B)$ we have $A(n) = B(n) \wedge C(n) \neq \varepsilon$. By the induction hypothesis, we have that for each such n, the meet $B(n) \wedge C(n)$ is the maximum simultaneous subtree of $B(n)$ and $C(n)$. Let $r \in \text{FTree}(P)$ such that $r \leq s, t$. Then one of the following must be true:

- $r = \varepsilon$. Then $r \leq m$ and the result holds.
- $r = (p, \emptyset)$. Then $r \leq m$ and the result holds.
- $r = (p, R)$ for non-empty R. Then $\#(R) = \#(B), \#(C)$, and for all $n \in \mathbb{N}$ with $n \leq \#(R)$, we have $R(n) \leq B(n), C(n)$. But since $A(n) = B(n) \wedge C(n)$ is the

maximum simultaneous subtree of $B(n)$ and $C(n)$, this implies that $R(n) \leq A(n)$. Thus $r \leq m$ and the result holds.

Thus, by the principle of induction, meets of any height are indeed the greatest maximum simultaneous subtree, and thus the order-theoretic meet. We conclude that $\leq$ gives rise to a meet semi-lattice. □

4.3 Generalised metrics and convexity

In this section we introduce a generalisation to the usual notion of metric spaces, and use this generalisation to define convexity. We then apply this to trees in the next section. Compare this approach to the approach taken by Bolt et al. in [BCG+16], where conceptual spaces were equipped with a linear structure which was then used to define convexity.

First, we define a distance space, which will take the role of the real numbers in the definition of generalised metrics:

Definition 4.18 (Distance space). *A* distance space $(D, \leq, +, 0)$ *is a set D equipped with a binary relation $\leq$, binary operation $+$ and a designated element 0 such that:*

(DS.1) $(D, +)$ *is an abelian group with* 0 *as its identity; and*

(DS.2) $(D, \leq)$ *is a totally ordered set; and*

(DS.3) the order is translation-invariant with respect to the group operation:

$$\forall a, b, c \in D\colon\ a \leq b \implies a + c \leq b + c. \tag{4.32}$$

When confusion is unlikely to arise, we will refer to a distance space $(D, \leq, +, 0)$ simply as D, leaving the structure implied. When we want to make the structure explicit, we will write $D = (D, \leq, +, 0)$ and then use D as a shorthand for the entire structure.

Given two distance spaces we can construct the product of these spaces:

Definition 4.19 (Product distance space). *Let $X = (X, \leq_X, +_X, 0_X)$ and $Y = (Y, \leq_Y, +_Y, 0_Y)$ be distance spaces. Then the* product distance space $X \times Y$ *is given by:*

- *The set* $X \times Y$ *equipped with*
- *total order* $\leq_{X\times Y}$ *given by* $(x_1, y_1) \leq_{X\times Y} (x_2, y_2)$ *if* $x_1 \neq x_2$ *and* $x_1 \leq_X x_2$*, or* $x_1 = x_2$ *and* $y_1 \leq_Y y_2$*; and*
- *binary operation* $+_{X\times Y}$ *given by* $(x_1, y_1) +_{X\times Y} (x_2, y_2) = (x_1 +_X x_2, y_1 +_Y y_2)$*; and*
- *the group identity* $0_{X\times Y} = (0_X, 0_Y)$*.*

Note that in general $X \times Y$ will be a very different object from $Y \times X$, due to the asymmetry in the order relation. The following lemma shows that this definition is valid.

Lemma 4.20. *Let* X *and* Y *be distance spaces. Then* $X \times Y$ *is a distance space.*

Proof. The group of $X \times Y$ is simply the direct product of the groups of X and Y, thus it is obvious that (DS.1) holds.

For (DS.2) we check antisymmetry, transitivity and totality.

For antisymmetry, suppose that $(x_1, y_1) \leq (x_2, y_2)$ and $(x_2, y_2) \leq (x_1, y_1)$. If $x_1 \neq x_2$ then the assumption implies that $x_1 \leq x_2$ and $x_2 \leq x_1$, thus by antisymmetry of $\leq$ on X, we conclude that $x_1 = x_2$, arriving at a contradiction. Thus $x_1 = x_2$, and the assumption implies that $y_1 \leq y_2$ and $y_2 \leq y_1$. Thus by antisymmetry of $\leq$ on Y, we conclude that $y_1 = y_2$. Thus $(x_1, y_1) = (x_2, y_2)$, as required.

For transitivity, suppose that $(x_1, y_1) \leq (x_2, y_2)$ and $(x_2, y_2) \leq (x_3, y_3)$. Note that we have $x_1 \leq x_2$ and $x_2 \leq x_3$, thus $x_1 \leq x_3$. If $x_1 \neq x_2$ or $x_2 \neq x_3$ then $x_1 \neq x_3$ (this follows from antisymmetry on X) and thus $(x_1, y_1) \leq (x_3, y_3)$. If instead $x_1 = x_2 = x_3$, then $y_1 \leq y_2$ and $y_2 \leq y_3$, thus $y_1 \leq y_3$ and thus again $(x_1, y_1) \leq (x_3, y_3)$, as required.

For totality, note that for $(x_1, y_1), (x_2, y_2)$, either $x_1 = x_2$ or $x_1 \neq x_2$. In the former case, totality follows from the totality on X, in the latter case it follows from totality on Y. Thus totality is satisfied, as required. This shows that we indeed have a total order, satisfying (DS.2).

Finally, for (DS.3), let $(x_1, y_1), (x_2, y_2), (x_3, y_3) \in X \times Y$ and suppose that $(x_1, y_1) \leq (x_2, y_2)$. Then either $x_1 \neq x_2$ or $x_1 = x_2$. In the former case, we have $x_1 \leq x_2$ and thus $x_1 + x_3 \leq x_2 + x_3$. We also have $x_1 + x_3 \neq x_2 + x_3$, by invertibility of x_3. Thus, in the former case $(x_1, y_1) + (x_3, y_3) \leq (x_2, y_2) + (x_3, y_3)$. In the latter case we have

$y_1 \leq y_2$, and thus $y_1 + y_3 \leq y_2 + y_3$. We also have $x_1 + x_3 = x_2 + x_3$, and thus $(x_1, y_1) + (x_3, y_3) \leq (x_2, y_2) + (x_3, y_3)$, as required. Thus (DS.3) is satisfied. □

The proof above is the only situation where invertibility of distance space elements is required. Thus condition (DS.1) could be replaced by the following, weaker requirements:

- $(D, +)$ is a commutative monoid with identity 0; and
- for any $x, y, z \in D$, if $x \neq y$, then $x + z \neq y + z$.

We also have the following lemma:

Lemma 4.21. *Let X and Y be distance spaces, and let $x_1, x_2 \in X$ and $y_1, y_2 \in Y$. If $x_1 \leq x_2$ and $y_1 \leq y_2$, then $(x_1, y_1) \leq (x_2, y_2)$ in the product space $X \times Y$.*

Proof. If $x_1 \neq x_2$ the result holds by $x_1 \leq x_2$. Otherwise, the result holds by $y_1 \leq y_2$. □

With this definition of a distance space, the definition of a generalised metric is obvious:

Definition 4.22 (Generalised metric)**.** *Let X be a set and $(D, \leq, +, 0)$ a distance space. A* generalised metric *is a function $d : S \times S \to D$ such that for any $x, y, z \in S$:*

(GM.1) $d(x, y) \geq 0$ *(non-negativity); and*

(GM.2) $d(x, y) = 0 \iff x = y$ *(identity of indiscernibles); and*

(GM.3) $d(x, y) = d(y, x)$ *(symmetry); and*

(GM.4) $d(x, z) \leq d(x, y) + d(y, z)$ *(triangle inequality).*

We say that (X, d) is a generalised metric space*.*

Note that $D = \mathbb{R}$ with the usual ordering and addition is a distance space, and generalised metrics to this distance space are exactly the usual metrics.

We can also define a product metric:

Definition 4.23 (Product metric)**.** *Let (X, d_X) and (Y, d_Y) be generalised metric spaces. Then define $(X, d_X) \times (Y, d_Y) = (X \times Y, d_X \times d_Y)$.*

The following lemma shows that this definition is valid.

Lemma 4.24. *Let (X, d_X) and (Y, d_Y) be generalised metric spaces. Then $(X, d_X) \times (Y, d_Y)$ is a generalised metric space.*

Proof. It is immediately obvious that (GM.1), (GM.2) and (GM.3) hold.

For (GM.4) (the triangle inequality), let $(x_1, y_1), (x_2, y_2), (x_3, y_3) \in X \times Y$. By the triangle inequality on X and Y, we have $d_X(x_1, x_3) \leq d_X(x_1, x_2) + d_X(x_2, x_3)$ and $d_Y(y_1, y_3) \leq d_Y(y_1, y_2) + d_Y(y_2, y_3)$. Thus, by lemma 4.21, we have

$$\begin{aligned} d_X \times d_Y((x_1, y_1), (x_3, y_3)) &= (d_X(x_1, x_3), d_Y(y_1, y_3)) \\ &\leq (d_X(x_1, x_2) + d_X(x_2, x_3), d_Y(y_1, y_2) + d_Y(y_2, y_3)) \\ &= d_X \times d_Y((x_1, y_1), (x_2, y_2)) + d_X \times d_Y((x_2, y_2), (x_3, y_3)). \end{aligned} \tag{4.33}$$

□

The notion of metric convexity carries over to generalised metrics. Note that metric convexity is distinct from te usual notion of convexity. First we define bridges and betweenness:

Definition 4.25 (Bridge and betweenness)**.** *Let (X, d) be a generalised metric space, and let $x, z \in X$. We define the* bridge from x to z *as:*

$$\text{Bridge}(x, z) = \{y \in X \mid d(x, z) = d(x, y) + d(y, z)\}, \tag{4.34}$$

and we say that $y \in X$ is between x and z *if $y \neq x, z$ and $y \in \text{Bridge}(x, z)$.*

Then a convex subset is simply defined as a set containing bridges between its elements:

Definition 4.26 (Metric convexity)**.** *Let (X, d) be a generalised metric space, and let $S \subseteq X$. We say that S is* convex *if for any $x, y \in S$:*

$$\text{Bridge}(x, y) \subseteq S. \tag{4.35}$$

Note that this definition agrees with definition 3.1.

As one would hope, products of convex subsets are convex:

Theorem 4.27. *Let (X, d_X) and (Y, d_Y) be generalised metric spaces, and let $S_X \subseteq X$ and $S_Y \subseteq Y$ be convex. Then $S_X \times S_Y \subseteq X \times Y$ is convex.*

Proof. Let $(x_1, y_1), (x_3, y_3) \in S_X \times S_Y$, and let $(x_2, y_2) \in \text{Bridge}((x_1, y_1), (x_3, y_3))$. Then

$$\begin{aligned} d_X \times d_Y((x_1, y_1), (x_3, y_3)) = d_X \times d_Y((x_1, y_1), (x_2, y_2)) \\ + d_X \times d_Y((x_2, y_2), (x_3, y_3)). \end{aligned} \tag{4.36}$$

This implies $d_X(x_1, x_3) = d_X(x_1, x_2) + d_X(x_2, x_3)$ and $d_Y(y_1, y_3) = d_Y(y_1, y_2) + d_Y(y_2, y_3)$. Thus $x_2 \in \text{Bridge}(x_1, x_3)$ and $y_2 \in \text{Bridge}(y_1, y_3)$. Therefore, by convexity of S_X and S_Y, $x_2 \in S_X$ and $y_2 \in S_Y$. Thus $(x_2, y_2) \in S_X \times S_Y$. We conclude that $\text{Bridge}((x_1, y_1), (x_3, y_3)) \subseteq S_X \times S_Y$. Thus $S_X \times S_Y$ is convex. □

4.4 A metric for trees

In order to consider notions of similarity and convexity for hierarchical conceptual spaces, we need to define a metric on trees. We will develop a metric based on the following principles:

(a) Structural differences are more important than differences in just node properties.

(b) Differences higher up in a tree are more important than differences further down.

In section 4.4.1 we will define the distance space that will represent structural differences, and introduce a few other concepts that will be required for what follows. Next, in section 4.4.2 we define a structural metric (ignoring node properties), and show that it is a valid generalised metric if the property space is trivial. Finally, in section 4.4.3, we extend the definition of the structural metric to a full tree metric, which also takes node properties into account.

4.4.1 Preliminary definitions

First, we define a structural distance space, for expressing structural distances between trees. The intuition behind the construction is that we have an infinite list of integers, each integer representing the distance due to a specific level of the trees. The order relation is defined such that deeper levels are only considered if there is no difference in the shallower levels. This corresponds to principle (b) above. The addition operation will be used to recursively define distance by summing over the children of a node.

Definition 4.28 (Structural distance space)**.** *The* structural distance space D_S *is given by:*

- *The set* $D_S = \{x : \mathbb{N} \to \mathbb{Z}\}$ *of functions from the natural numbers to the integers.*
- *The lexicographical order relation* $\leq$ *characterised as follows: Let* $x, y \in D_S$*. If* $x = y$ *then certainly* $x \leq y$*, so we can assume that* $x \neq y$*. Let* $i \in \mathbb{N}$ *be the smallest number such that* $x(i) \neq y(i)$*. If* $x(i) < y(i)$*, then* $x \leq y$*; otherwise* $y \leq x$*.*
- *The addition operation* $+$ *given by pointwise addition:* $(x + y)(i) = x(i) + y(i)$*.*
- *The identity* 0_S *defined by* $0_S(i) = 0$*.*

It is easily checked that this definition indeed satisfies the requirements of a distance space.

Note that D_S can be alternatively described as the infinite product of integer distance spaces:

$$D_S = \bigtimes_{i \in \mathbb{N}} (\mathbb{Z}, \leq, +, 0). \tag{4.37}$$

We also define the following function on the structural distance space:

Definition 4.29 (Shift function)**.** *The* shift function $\sigma : D_S \to D_S$ *is given by:*

$$\sigma(x)(i) = \begin{cases} 0 & \text{if } i = 0, \\ x(i-1) & \text{otherwise,} \end{cases} \tag{4.38}$$

for all $x \in D_S$ *and* $i \in \mathbb{N}$*.*

This function will be useful when reasoning recursively about distances.

Some properties of the shift function:

Lemma 4.30. *The shift function satisfies the following properties:*

- σ *is linear:* $\sum_{d \in D} \sigma(d) = \sigma\left(\sum_{d \in D} d\right)$*, for all* $D \subseteq D_S$*; and*
- σ *is strictly increasing:* $\sigma(x) \leq \sigma(y) \iff x \leq y$*, for all* $x, y \in D_S$*.*

Proof. Follows immediately from the definition. □

Finally, we will use a Kronecker delta-like notation to designate specific elements $\delta_k \in D_S$, for $k \in \mathbb{N}$:

$$\delta_k(i) = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{otherwise.} \end{cases} \tag{4.39}$$

4.4.2 The structural metric

We now define the structural metric on finite trees:

Definition 4.31 (Structural metric)**.** *For P a set, the* structural metric $d_S : \mathrm{FTree}(P) \times \mathrm{FTree}(P) \to D_S$ *is given by:*

(SM.1) If both trees are empty:

$$d_S(\varepsilon, \varepsilon) = 0_S. \tag{4.40}$$

(SM.2) If one of the trees is empty:

$$d_S(\varepsilon, (p, C)) = d_S((p, C), \varepsilon) = \delta_0 + \sum_{t \in C} \sigma\big(d_S(\varepsilon, t)\big). \tag{4.41}$$

(SM.3) Otherwise:

$$d_S((p, A), (q, B)) = \sum_{n=0}^{m-1} \sigma\big(d_S(A(n), B(n))\big), \tag{4.42}$$

where $m = \max(\#(A), \#(B))$.

The metric essentially counts, per node, the number of nodes that are missing from one tree or the other, summing these values per level.

The following theorem shows that the metric defined above is a generalised metric when the property space is trivial.

Theorem 4.32. *For $P = \{*\}$, the structural metric d_S is a generalised metric.*

Proof. It is immediately obvious that (GM.1) and (GM.3) hold. For the other conditions, we proceed by induction on the maximum height h of the trees.

For $h = 0$, the only tree of height at most h is ε. Both (GM.2) and (GM.4) are trivially satisfied.

Now let $h \geq 1$, and assume that (GM.2) and (GM.4) hold for trees of height at most $h - 1$. Note that for any tree of height at most h, the children of the root will have height at most $h-1$, thus by the induction hypothesis, we can apply the identity of indiscernibles and the triangle equality to such trees.

Identity of indiscernibles (GM.2):

If $x = y = \varepsilon$, the condition is trivially satisfied in the same way as for $h = 0$.

If one of x and y is ε, and the other is of the form $(*, C)$, then case (SM.2) of definition 4.31 applies, and we have $d_S(x, y) = \delta_0 + \sigma(\ldots) \neq 0_S$, and $x \neq y$, thus the condition is satisfied.

For the final case, we have $x = (*, A)$ and $y = (*, B)$ and can assume without loss of generality that $\#(A) \leq \#(B)$. Thus case (SM.3) applies. Note that all the terms in the sums are non-negative by property (GM.1). If $x \neq y$, then $A \neq B$, so there is some $n \in \mathbb{N}$ such that $A(n) \neq B(n)$, and we can assume that $n < \max(\#(A), \#(B))$. This implies that the sum in equation (4.42) is positive, thus $d_S(x, y) > 0_S$ as required. If, instead, $x = y$, then for all $n \in \mathbb{N}$, $A(n) = B(n)$. Thus the sum is 0_S (again, by the induction hypothesis), and we have $d_S(x, y) = 0_S$, as required.

Triangle inequality (GM.4):

First, note that for any $a = (*, A)$ and $b = (*, B)$, $(d_S(a, b))(0) = 0$, and $(d_S(a, b))(1) = |\#(A) - \#(B)|$.

If x, y and z are not all distinct, the triangle equality certainly holds (by property (GM.2)). Therefore we can assume that x, y and z are all distinct.

If $y = \varepsilon$ then $d_S(x, y) + d_S(y, z) = \delta_0 + \delta_0 + \sigma(\ldots)$, thus $(d_S(x, y) + d_S(y, z))(0) = 2$. On the other hand, $(d_S(x, z))(0) = 0$ (since neither x or z is equal to ε). So $d_S(x, z) < d_S(x, y) + d_S(y, z)$, and the triangle equality is satisfied.

Suppose $x = \varepsilon$, $y = (*, B)$ and $z = (*, C)$. Then the distance $d_S(y, z)$ is calculated via case (SM.3). So, with $m = \max(\#(B), \#(C))$,

$$
\begin{aligned}
d_S(x,y) + d_S(y,z) &= \delta_0 + \sum_{t \in B} \sigma\big(d_S(\varepsilon, t)\big) + \sum_{n=0}^{m-1} \sigma\Big(d_S\big(B(n), C(n)\big)\Big) \\
&= \delta_0 + \sum_{n=0}^{m-1} \Big[\sigma\Big(d_S\big(\varepsilon, B(n)\big) + d_S\big(B(n), C(n)\big)\Big)\Big] \\
&\geq \delta_0 + \sum_{n=0}^{m-1} \sigma\Big(d_S\big(\varepsilon, C(n)\big)\Big) \\
&= \delta_0 + \sum_{t \in C} \sigma\big(d_S(\varepsilon, t)\big) = d_S(x,z),
\end{aligned}
\tag{4.43}
$$

as required. Note that we used the induction hypothesis to apply the triangle equality term-wise.

The case with $z = \varepsilon$ is symmetric to the case with $x = \varepsilon$.

For the final case, let $x = (*, A)$, $y = (*, B)$ and $z = (*, C)$. Then case (SM.3) applies to both $d_S(x,y)$ and $d_S(y,z)$. Let $m = \max(\#(A), \#(B))$ and $m' = \max(\#(B), \#(C))$, then we have:

$$
d_S(x,y) + d_S(y,z) = \sum_{n=0}^{m-1} \sigma\big(d_S(A(n), B(n))\big) + \sum_{n'=0}^{n'-1} \sigma\big(d_S(B(n), C(n))\big). \tag{4.44}
$$

Setting $m^* = \max(m, m')$, we can rewrite this to:

$$
d_S(x,y) + d_S(y,z) = \sum_{n=0}^{m^*-1} \sigma\big(d_S(A(n), B(n)) + d_S(B(n), C(n))\big). \tag{4.45}
$$

We then use the induction hypothesis to apply the triangle equation term-wise:

$$
d_S(x,y) + d_S(y,z) \geq \sum_{n=0}^{m^*-1} \sigma\big(d_S(A(n), C(n))\big) = d_S(x,z), \tag{4.46}
$$

as required.

We conclude that by the principle of induction both the identity of indiscernibles and the triangle inequality hold for all trees of finite height. Thus d_S is a generalised metric on FTree($\{*\}$). □

4.4.3 The tree metric

When the property space is not trivial, the structural metric will clearly fail to be a generalised metric, since identity of indiscernibles (GM.2) is violated. However, we can extend this metric to trees with non-trivial property spaces with a minor modification. To that end, we first define a restriction on the sets P that we will use as property spaces:

Definition 4.33 (Property space)**.** *A* property space *is a set P equipped with a generalised metric $d_P : P \times P \to D_P$.*

Definition 4.34 (Tree metric)**.** *Let P be a property space with generalised metric $d_P : P \times P \to D_P$. Then define*

$$D_{P^*} = \bigtimes_{i \in \mathbb{N}} D_P. \tag{4.47}$$

Now the tree metric $d_T[d_P] : \mathrm{FTree}(P) \times \mathrm{FTree}(P) \to D_S \times D_{P^*}$*, written simply as d_T when the property metric is clear from context, is given by:*

(TM.1) If both trees are empty:

$$d_T(\varepsilon, \varepsilon) = (0_S, 0_P). \tag{4.48}$$

(TM.2) If one of the trees is empty:

$$d_T(\varepsilon, (p, C)) = d_S((p, C), \varepsilon) = (\delta_0, 0_P) + \sum_{t \in C} \sigma_T\big(d_T(\varepsilon, t)\big). \tag{4.49}$$

(TM.3) Otherwise:

$$d_T((p, A), (q, B)) = (0_S, d_P(p, q)) + \sum_{n=0}^{m-1} \sigma_T\big(d_T(A(n), B(n))\big), \tag{4.50}$$

where $m = \max(\#(A), \#(B))$.

Here, $\sigma_T : D_S \times D_{P^*} \to D_S \times D_{P^*}$ is defined analogously to σ from definition 4.29:

$$\sigma_T(x, y)(i) = \begin{cases} (0, 0_P) & \text{if } i = 0, \\ (x, y)(i-1) & \text{otherwise,} \end{cases} \tag{4.51}$$

where for $(x, y) \in D_S \times D_{P^*}$ we write $(x, y)(i) := (x(i), y(i))$.

Theorem 4.35. *For (P, d_P) a generalised metric property space, the tree metric $d_T[d_P]$ is a generalised metric.*

Proof. This is a trivial modification of the proof for theorem 4.32. □

Note that the distance space product $D_S \times D_{P^*}$ yields an order in which D_{P^*} is only considered if there are no differences in D_S (see definition 4.19). This corresponds to principle (a) above. Principle (b) (differences higher up are more important) for structural differences is preserved from the structural metric, and for properties it is guaranteed by the product definition of D_{P^*}. Thus the metric d_T satisfies both principles.

Chapter 5

A category of trees

In this chapter we define a category of trees, and then explore the properties of states in this category when interpreted as a representation of meaning. In section 5.1 we define the category **FTree**(P) for property space P. In section 5.2 we discuss how the representation of concepts differs from the representation of concrete objects. Finally, in section 5.3 we discuss how entailment and ambiguity are encoded in our category.

5.1 The category FTree(P)

Definition 5.1 (Category of trees). *Let P be a property space. Then the* category **FTree**(P) of finite trees with properties in P *is the category with:*

- *Objects: the sets $I = \{\varepsilon\}$ and* $\mathrm{FTree}(P)$*, and any finite Cartesian product of these.*
- *Morphisms: for A, B an object of the category, the morphisms between A and B are the relations between A and B.*
- *Composition: the usual composition of relations; if $R : A \to B$ and $S : B \to C$, then*

$$S \circ R : A \to C := \{(a, c) \in A \times C \,|\, \exists b \in B\colon\ (a, b) \in R \wedge (b, c) \in S\}. \tag{5.1}$$

Note that the metric and order defined on trees can be extended to all the objects of **FTree**(P) by using the product metric of definition 4.23 and the usual product order. The unit carries a trivial structure.

Also note that $\mathbf{FTree}(P)$ is a full subcategory of **Rel**. Unsurprisingly then, $\mathbf{FTree}(P)$ admits much the same structure as **Rel**. In particular, we define the compact closed structure on $\mathbf{FTree}(P)$ exactly as for **Rel**. Each time, we omit the proofs of correctness for these structures since they follow directly from the correctness of these structures for **Rel**.

Theorem 5.2. *Let P be a property space. Then $\mathbf{FTree}(P)$ admits a monoidal structure with the monoidal functor given by:*

$$\begin{aligned}\otimes : \mathbf{FTree}(P) \times \mathbf{FTree}(P) &\to \mathbf{FTree}(P),\\ (A, B) &\mapsto A \times B,\\ (R : A \to B, S : X \to Y) &\mapsto R \times S : A \times X \to B \times Y.\end{aligned} \tag{5.2}$$

The monoidal unit is given by $I = \{\varepsilon\}$, the associator has components

$$\begin{aligned}\alpha_{A,B,C} : (A \otimes B) \otimes C &\to A \otimes (B \otimes C),\\ ((a, b), c) &\sim (a, (b, c)),\end{aligned} \tag{5.3}$$

the left unitor has components

$$\begin{aligned}\lambda_A : I \otimes A &\to A,\\ (\varepsilon, a) &\sim a,\end{aligned} \tag{5.4}$$

and the right unitor has components

$$\begin{aligned}\rho_A : A \otimes I &\to A,\\ (a, \varepsilon) &\sim a.\end{aligned} \tag{5.5}$$

Theorem 5.3. *Let P be a property space. Then $\mathbf{FTree}(P)$ admits a symmetric structure with the braiding given by*

$$\begin{aligned}\sigma_{A,B} : A \otimes B &\to B \otimes A,\\ (a, b) &\sim (b, a).\end{aligned} \tag{5.6}$$

Theorem 5.4. *Let P be a property space. Then $\mathbf{FTree}(P)$ has a duality structure where every object is its own dual and the unit and counit are given by:*

$$\begin{aligned}\eta_A : I &\to A \otimes A,\\ \varepsilon &\sim (a, a) \text{ for all } a \in A,\end{aligned} \tag{5.7}$$

and

$$\begin{aligned}\epsilon_A : I &\to A \otimes A,\\ (a, a) &\sim \varepsilon \text{ for all } a \in A.\end{aligned} \tag{5.8}$$

To avoid confusion between the empty tree ε and the counit ϵ, we will typically just refer to 'the counit', or draw a cup, while continuing to use ε for the empty tree.

Corollary 5.5. *Let P be a property space. Then* **FTree**(P) *is a symmetric monoidal category with duals. Thus* **FTree**(P) *is a compact closed category.*

5.2 Concepts versus concrete objects

In the remainder of dissertation we will make a distinction between *concepts*[4] and *concrete objects*. A concept is an abstract notion to which any given concrete object may or may not conform. Examples of concepts include 'insect', 'human', and 'furniture', while a concrete object could be 'that spider', 'John', 'our table'.

If we think of the concept-concrete object-distinction in terms of detail, it becomes a little clearer how this distinction might take shape in our conceptual space. A concrete object has a maximum level of detail, representing our knowledge about that object. Adding extra detail to the representation of a concrete object corresponds to gaining more knowledge about that object and should thus be considered a fundamental change in our mental model of the object. On the other hand, we do not generally think of an object in all of its detail; a person who owns a football probably has some fairly extensive knowledge of the details of the ball: the pattern of the leather patches, the specific shape of the stitching, etc.; however, when he tells a friend he will bring the football with him on his holiday trip, his mental model of the ball is probably limited to a simple sphere (at least as far as shape is concerned). Thus forgetting details, or generalising, is something we constantly do for concrete objects.

In contrast, a concept has a minimum level of detail, representing the properties that any object conforming to that concept should, at a minimum, have. Removing detail from the representation of a concept represents a fundamental change in the mental model of that concept, since it corresponds to generalising that concept. On the other hand, filling in extra details corresponds to instantiating that concept and is something we are constantly doing.

This suggests that concrete objects should be represented by states which are in some sense closed under 'generalising', while concepts should be represented by states

[4]While Gärdenfors made a distinction between concepts and properties (see chapter 3.1), these two notions coincide in our case, since our conceptual space covers only a single domain.

which are in some sense closed under 'specialising'. In this section we develop a mathematically rigorous model for this way of thinking.

5.2.1 Closed sets

Definition 5.6 (Upward closed). *Let $(P, \leq)$ be a partially ordered set, and let $S \subseteq P$. We say that S is* upward closed *if:*

$$\forall x \in S, y \in P \colon\ x \leq y \implies y \in S. \tag{5.9}$$

Definition 5.7 (Basis of an upward closed set). *Let $(P, \leq)$ be a partially ordered set, and let $S \subseteq P$ be an upward closed set. Let $B \subseteq S$. We say that B is a* basis *of S and write $S = \uparrow B$ if:*

$$\forall x \in P \colon\ x \in S \iff (\exists b \in B \colon\ b \leq x)\,. \tag{5.10}$$

And similarly:

Definition 5.8 (Downward closed). *Let $(P, \leq)$ be a partially ordered set, and let $S \subseteq P$. We say that S is* downward closed *if:*

$$\forall x \in S, y \in P \colon\ y \leq x \implies y \in S. \tag{5.11}$$

Definition 5.9 (Basis of a downward closed set). *Let $(P, \leq)$ be a partially ordered set, and let $S \subseteq P$ be a downward closed set. Let $B \subseteq S$. We say that B is a* basis *of S and write $S = \downarrow B$ if:*

$$\forall x \in P \colon\ x \in S \iff (\exists b \in B \colon\ x \leq b)\,. \tag{5.12}$$

We say that a set is *closed* if it is either upward or downward closed. Note that this notion of closedness is unrelated to the topological notion.

5.2.2 Convexity of closed sets

In this section, we will prove that closed sets with convex bases are convex. We first prove the following theorem which relates bridges of trees to bridges of their subtrees:

Theorem 5.10. *Let P be a property space and let $r, t, x, z \in \mathrm{FTree}(P)$. Suppose that $r \leq x$ and $t \leq z$. Then for any $y \in \mathrm{Bridge}(x, z)$ there exists an $s \in \mathrm{Bridge}(r, t)$ such that $s \leq y$.*

Proof. We proceed by induction on $h = \max(\text{height}(r), \text{height}(t))$.

For $h = 0$, $r = t = \varepsilon$ and the result holds by always setting $s = \varepsilon$.

Now let $h \geq 1$ and assume the result holds for all $h' < h$. If either of r or t is ε, we can again set $s = \varepsilon$. Otherwise, we can write $r = (p_r, R)$ and $t = (p_t, T)$, and assume without loss of generality that $\#(R) \leq \#(T)$. Let $y \in \text{Bridge}(x, z)$ arbitrarily. If either x or z is equal to ε, then either r or t is equal to ε, contradicting our assumption. Similarly, $y = \varepsilon$ implies that either x or z is equal to ε, thus giving a contradiction. Thus we can write $x = (p_r, X)$, $y = (p_y, Y)$, and $z = (p_t, Z)$. Note that $p_y \in \text{Bridge}(p_r, p_t)$.

If $\#(T) = 0$, then $s = (p_y, \emptyset)$ satisfies the requirements. Otherwise, note that $d_T(x, z)(1) = (|\#(X) - \#(Z)|, d)$ for some $d \in D_P$. From $y \in \text{Bridge}(x, z)$ we know that $d_T(x, y)(1)+d_T(y, z)(1) = d_T(x, z)(1)$, and thus $|\#(X)-\#(Y)|+|\#(Y)-\#(Z)| = |\#(X)-\#(Z)|$. We conclude that $\#(X) \leq \#(Y) \leq \#(Z)$. Moreover, since $\#(T) > 0$, we have $\#(T) = \#(Z)$. Similarly, either $\#(R) = 0$, or $\#(R) = \#(X)$. Thus we have $\#(R) \leq \#(Y) \leq \#(Z)$. Now, for each $0 \leq n < \#(Y)$, we have:

- $R(n) \leq X(n)$,
- $T(n) \leq Z(n)$, and
- $Y(n) \in \text{Bridge}(X(n), Z(n))$.

Since $\max(\text{height}(R(n)), \text{height}(T(n))) < \max(\text{height}(r), \text{height}(t)) = h$, we can apply the induction hypothesis. So, for every n, there is a tree $S(n) \in \text{Bridge}(R(n), T(n))$ such that $S(n) \leq Y(n)$. We then simply construct s as:

$$s = (p_Y, \{n \mapsto S(n) \mid n < \#(Y)\}). \tag{5.13}$$

Clearly $s \leq y$ and $s \in \text{Bridge}(r, t)$. This completes the proof. □

The symmetric case also holds:

Theorem 5.11. *Let P be a property space and let $r, t, x, z \in \text{FTree}(P)$. Suppose that $x \leq r$ and $z \leq t$. Then for any $y \in \text{Bridge}(x, z)$ there exists an $s \in \text{Bridge}(r, t)$ such that $y \leq s$.*

Proof. The argument is symmetric to the argument in the proof of theorem 5.10. □

These theorems have some useful corollaries:

Corollary 5.12. *Let P be a property space and let $r, x, y, z \in \mathrm{FTree}(P)$. Suppose that $r \leq x, z$ and $r \nleq y$. Then $d_T(x, z) < d_T(x, y) + d_T(y, z)$.*

Proof. We use the contrapositive of theorem 5.10 with $r = t$. For clarity, we state theorem 5.10 in predicate logic:

$$\begin{aligned} &\forall r, t, x, y, z \in \mathrm{FTree}(P)\colon \\ &\qquad (r \leq x \wedge t \leq z) \implies (y \in \mathrm{Bridge}(x, z) \implies \exists s \in \mathrm{Bridge}(r, t)\colon\ s \leq y)\,. \end{aligned} \tag{5.14}$$

The statement $r \nleq y$ implies

$$\neg\left(\exists s \in \mathrm{Bridge}(r, r)\colon\ s \leq y\right). \tag{5.15}$$

Using the contrapositive of the consequent of the top-level implication in equation (5.14), we get:

$$\neg(y \in \mathrm{Bridge}(x, z)). \tag{5.16}$$

This implies that $d_T(x, z) < d_T(x, y) + d_T(y, z)$, as required. □

We also get a convexity result for closed sets of trees:

Corollary 5.13. *Let P be a property space and let $S \subseteq \mathrm{FTree}(P)$. Suppose that S is upward or downward closed with B a basis. Then if B is convex, so is S.*

And finally, the main result of this section:

Theorem 5.14. *Let P be a property space, A an object of* $\mathbf{FTree}(P)$*, and φ a state of A. Suppose that the set S represented by φ is upward or downward closed with B a basis. Then if B is convex, so is S.*

Proof. We can write $A \simeq \bigotimes_{i=1}^{n} \mathrm{FTree}(P)$ for some $n \in \mathbb{N}$.

Suppose that S is upward closed with convex basis B. For $1 \leq i \leq n$, let $\pi_i : S \to \mathrm{FTree}(P)$ be the i-th projection operator. Let $x, z \in S$. Then there are $r, t \in B$ such that $r \leq x$ and $t \leq z$. Let $y \in \mathrm{Bridge}(x, z)$ arbitrarily. Then for every $1 \leq i \leq n$, we have:

- $\pi_i(y) \in \mathrm{Bridge}(\pi_i(x), \pi_i(z))$; and
- $\pi_i(r) \leq \pi_i(x)$, and $\pi_i(t) \leq \pi_i(z)$.

Thus by theorem 5.10, for every $1 \leq i \leq n$ there is an $s_i \in \text{Bridge}(\pi_i(r), \pi_i(t))$ such that $s_i \leq \pi_i(y)$. Then $s = (s_1, \ldots, s_n) \in \text{Bridge}(r, t)$. Since B is convex, this implies that $s \in B$. We also have $s \leq y$, and thus conclude that $y \in S$.

Thus the set S is convex.

The argument for the downward closed case is symmetric. $\square$

5.2.3 Closed sets as a model for concepts and concrete objects

In the context of our conceptual space, an upward closed set essentially models the possibility of arbitrary specialisation, while a downward closed set models the possibility of complete generalisation. Based on our discussion earlier in this section, this suggests that upward and downward closed sets are a good model for concepts and concrete objects, respectively.

However, for concrete objects the possibility of complete generalisation seems like too strong an assumption. While it is true that any concrete object could be thought of in most general terms, i.e. 'just a thing', when actually discussing an object or thinking about it, a minimum level of detail is quickly established. When a speaker tells someone else that 'Alice went into the kitchen', both speakers have now formed a mental model of the kitchen that, at the very least, allows for occupation by a person. Generalising the kitchen to just some object which may or may not allow people to go into it is no longer a valid mental manipulation, since it would make it impossible to faithfully represent the information that has been communicated.

We conclude that when concrete objects interact with other parts of a sentence, that object's potential for generalisation may be diminished. In the above example this interaction happened through the preposition 'in'. This will be considered in more detail when we discuss the role of prepositions in our model in chapter 6. Another example of how this interaction might occur is through being acted upon by a concept; in the sentence 'the kitchen is a room', the concrete object 'the kitchen' is acted upon by the concept 'room'. The result is that the potential for generalising 'the kitchen' is limited to only those generalisations which are still an instance of 'room'.

5.3 Entailment and ambiguity

In section 4.2.3 we discussed how the subtree relation can witness entailment of concepts: $|\text{insect}\rangle \leq |\text{ant}\rangle$, because 'insect' is more general than 'ant'. By representing concepts as upward closed sets instead of single trees, this entailment relation is captured by $\supseteq$ rather than $\leq$: $\uparrow|\text{insect}\rangle \supseteq \uparrow|\text{ant}\rangle$. And in fact, this relation is more powerful, since it also allows us to represent entailment with respect to the property space: we can have $\uparrow|\text{rectangle}\rangle \supseteq \uparrow|\text{square}\rangle$, where the inclusion is not due to $|\text{rectangle}\rangle$ having a shallower tree structure than $|\text{square}\rangle$, but due to representing a larger region in the property space.

In section 3.1 we discussed the thesis by Gärdenfors that concepts are convex regions. Then, in section 5.2.2 we showed that we can ensure that upward closed sets are convex by restricting ourselves to convex bases. Finally, in section 5.2.3 we argued that upward closed sets are a good model for concepts. Thus we might be inclined to represent concepts as upward closed sets with convex bases. However, by allowing ourselves to use non-convex sets, for both concepts and concrete objects, we can encode ambiguity.

Consider the word 'queen'. Depending on the context, this word might refer to the concept of a monarch, a specific monarch, the band *Queen*, a chess piece (either conceptually or as a concrete object), or something completely different. In our model, we would first determine whether the word is being used conceptually or concretely, and then represent the word as the union of the representations of all the specific conceptual or concrete meanings. Then, the correct meaning will (hopefully) be resolved by the interaction of the word with other parts of a sentence.

Contrasting our approach with the one by Ashoush and Coecke [AC16], we note that in our model representations of entailment and ambiguity arise naturally from the construction of the conceptual space, whereas Ashoush and Coecke create a separate structure to represent these aspects using a model that is otherwise unable to capture them. On the other hand, the approach by Ashoush and Coecke allows for different gradations of entailment and ambiguity, while our model is unable to capture such nuance. This could be especially problematic when a word has a large number of obscure meanings.

Chapter 6

A hierarchical category for shapes

Up until now we have treated the category $\mathbf{FTree}(P)$ in abstract terms, building and justifying the mathematical machinery required for this category, but without specifying it in detail. We are now ready to define a category **Shape** which can be used as a conceptual space representing shapes in a distributional model of meaning. We define **Shape** as a specific instance of $\mathbf{FTree}(P)$ by fixing a property space P.

In section 6.1 we construct a property space to implement the Marr-Nishihara model. Then, in section 6.2 we extend this property space by adding preposition attachment points.

6.1 Node properties for the Marr-Nishihara model

We now implement the Marr-Nishihara model as described in section 4.1. Unfortunately, this will involve a number of ad-hoc poorly justified choices. This is a result of the fact that these choices should be made based upon experimental evidence, or at the very least based upon arguments from cognitive science or linguistics, and not mathematical arguments. We therefore settle for making some poorly justified choices and suggest that future researchers from those fields reconsider the issue at a later point.

As mentioned in section 4.1, a node has several properties: dimensions, orientation and position. We will interpret the orientation and position as relative to the parent node. For the root node, we will simply choose to set the position to the origin and define orientation relative to the observer. Similarly, dimensions should be considered relative, since (absolute) size is not an aspect of shape. Thus we will arbitrarily choose to give the root node a volume of 1.

As argued in section 4.1, the Marr-Nishihara model was primarily designed to describe the shape of animals. To make the model more well-suited for the many somewhat cubic man-made objects, we deviate from the Marr-Nishihara model by using rectangular cuboids as our primitive. Thus our nodes have three dimensions: width, depth and height. This choice of primitive makes Cartesian coordinates a natural choice for describing the relative position of nodes, again deviating from Marr and Nishihara's choice of cylindrical coordinates. For the relative orientation Marr and Nishihara use spherical coordinates to describe the orientation and length of an axis. We have absorbed the length of the axis in the dimensions of the primitive, and require an extra angle because we have lost a degree of symmetry by switching from cylinders to cuboids. Thus we will describe the orientation using an Euler angle system, consisting of three angles. While there are many different conventions for the exact use of Euler angles, such details won't really be relevant to our discussion, thus we leave them unspecified.

Besides defining the information on the nodes, defining a property space also involves defining a metric on this information. For the dimensions and position a normal euclidean metric seems appropriate. For the orientation we use an approach similar to the one used for defining a polar metric on colour in equation (3.1):

$$d_{\text{orientation}}\big((\varphi_1, \theta_1, \psi_1), (\varphi_2, \theta_2, \psi_2)\big) = \sqrt{d_\theta(\varphi_1, \varphi_2)^2 + d_\theta(\theta_1, \theta_2)^2 + d_\theta(\psi_1, \psi_2)^2}, \quad (6.1)$$

where d_θ is the same function as in equation (3.1).

We will combine these three metrics using the product metric of definition 4.23. We do need to choose in what order we combine the metrics, since this order matters. We will consider the relative position to be more important than the dimensions, which is more important than the relative orientation.

This gives us the following definition:

Definition 6.1 (Marr-Nishihara property space)**.** *The Marr-Nishihara property space is given by:*

- *The set $M = P \times D \times O$, where $P = \mathbb{R}^3$ encodes relative position, $D = (\mathbb{R}^+)^3$ encodes the dimensions, and $O = [0, 2\pi)^3$ encodes relative orientation.*

- *The metric* $d_M = d_{position} \times d_{dimension} \times d_{orientation} : M \to \mathbb{R}^3$*, where* $d_{position}$ *and* $d_{dimension}$ *are the usual euclidean metrics, and* $d_{orientation}$ *is defined in equation* (6.1)*. Each of these three functions are (normal) metrics, and thus generalised metric, so their product is a generalised metric.*

6.2 Decorating trees with preposition attachment points

As an example of how to exploit the hierarchical structure of the model we're constructing, we equip our model with preposition attachment points. We will extend our property space to represent such attachment points on nodes, and define attachment relations that exploit these attachment points.

Definition 6.2 (Preposition property space)**.** *Let* P *be a (finite) set of prepositions. Then the property space* A *for preposition attachment points is given by the set* $\mathcal{P}(P)$*, together with the following metric:*

$$d_A(Q, R) = \sum_{p \in P} d_p(Q, R), \tag{6.2}$$

where for $p \in P$*,* d_p *is given by:*

$$d_p(Q, R) = \begin{cases} 0 & \textit{if } p \in Q \wedge p \in R; \textit{ or } p \notin Q \wedge p \notin R, \\ 1 & \textit{otherwise.} \end{cases} \tag{6.3}$$

It is easily checked that d_A *is indeed a metric for* $\mathcal{P}(P)$*.*

We combine this with the Marr-Nishihara property space, letting the Marr-Nishihara structure take precedence over the preposition attachment point structure in the metric:

Definition 6.3 (Property space for shapes)**.** *The property space for shapes is given by the set* $S = M \times \mathcal{P}(P)$ *with the metric* $d_{Shape} = d_M \times d_A$*.*

Definition 6.4 (Category of shapes)**.** *The category* **Shape** *is defined as* **FTree**(S)*.*

And we define the preposition attachment relations:

Definition 6.5 (Preposition attachment relation)**.** *Let* $p \in P$ *be a preposition. Then the preposition attachment relation* A_p *for* p *is a relation from* $\mathrm{FTree}(S) \times \mathrm{FTree}(S)$ *to* $\mathrm{FTree}(S)$*. Given a tree* s *(the subject) and* o *the (object), it gives:*

- *If* $o = \epsilon$*:* $\emptyset$.
- *If* o *is of the form* $((m, a), C)$*, define*

$$X = \bigcup_{t \in C} \{((m, a), C[t'/t]) \mid t' \in A_p(s, t)\}, \tag{6.4}$$

where $C[t'/t]$ *means, the finite labelled subset obtained by replacing* t *with* t' *in* C*.*[5] *If* $p \in a$ *then* $A_p(s, o)$ *gives:*

$$\Big\{\big((m, a), C \cup \{s\}\big)\Big\} \cup X, \tag{6.5}$$

where $C \cup \{s\}$ *appends* s *to the end of* C*. Otherwise, if* $p \notin a$*, then* $A_p(s, o)$ *gives:*

$$X. \tag{6.6}$$

In words:

- Empty trees yield an empty set: there are no valid attachments.
- For non-empty trees, recurse over the children, collecting all valid attachments in a set. For each child t, the valid attachments of s in t are calculated, and then for each of those valid attachments we yield a node where the child has been replaced with that attachment. Additionally, if the root node itself allows attachment, yield one additional element where s has been attached under the root.

Thus $A_p(s, o)$ yields all the trees where s has been attached under exactly one node of o that allowed attachment for the preposition p.

As an example, suppose that in the course of a conversation the word 'table' has been used to mean either a desk or a coffee table. The desk has a drawer and the coffee table does not. Thus the representation of the desk contains attachment points for the preposition 'in', while the coffee table does not. The phrase 'the pen is in the table' could then be calculated as:

$$A_{\text{in}}(|\text{pen}\rangle, |\text{table}\rangle), \tag{6.7}$$

where $|\text{table}\rangle = \downarrow\{|\text{coffee table}\rangle, |\text{desk}\rangle\}$. This calculation would yield all the elements of $|\text{table}\rangle$ which have an 'in' attachment point, with (an element of) $|\text{pen}\rangle$ added to the node containing that attachment point. This yields a mental model which is different from the mental model of $|\text{table}\rangle$ in the following ways:

[5] With slight abuse of notation since t might occur multiple times in C; this notation is used because it is clearer than the index-based notation.

- The mental model now describes an arrangement consisting not just of the desk, but also of the pen sitting in the desk.
- Only elements originating from $|\text{desk}\rangle$ remain; the word ‘table’ has been disambiguated (see also section 5.3).
- The potential for generalisation has been diminished: we no longer allow ourselves to generalise to a model where the desk having a drawer is not part of the description (see also section 5.2.3).

Chapter 7

Concluding remarks

In this dissertation we have given an overview of some of the existing theory in the field of compositional distributional models of meaning and conceptual spaces. We then went on to define a hierarchical conceptual space that can be used to represent shape and have shown how it might fit into the compositional distributional model of meaning. In the process we defined generalisations of the usual notions of metrics and convexity and proved a number of results for these notions. In this chapter we briefly discuss some of the strengths and weaknesses of the model we developed, and suggest further avenues of research.

As argued in section 4.1, the Marr-Nishihara-based model constructed in this dissertation has several advantages, including explicit encoding of the level of detail, explicit encoding of whole-part relationships, and a clear correspondence to the way shapes are usually described in speech. Moreover, our model includes a notion of convexity that arises from the structure of the conceptual space, via a generalised metric defined using the structure of the space. We contrast this with the approach by Bolt et al. in [BCG$^+$16], where the convex structure was more ad-hoc and a notion of distance was not defined. Thus it can be argued that the model developed in this dissertation is closer to what was suggested by Gärdenfors in his work on conceptual spaces.

We have shown how our model encodes entailment, as well as more subtle notions of generalisation and specialisation. Moreover, our model is also able to encode ambiguity, and provides mechanisms for disambiguation. These notions arise naturally from the structure of the conceptual space. We contrast this with the approach used by Ashoush and Coecke in [AC16], as well as similar approaches, where the structure required to encode these notions was imposed externally.

On the other hand, the models like the one by Ashoush and Coecke allow for gradations of entailment and/or ambiguity to be encoded, where our model is not able to encode such subtleties. Moreover, our model is fairly complex to work with, both conceptually and computationally.

Another specific shortcoming of our model is the fact that nodes of trees are ordered. This is not desirable in a conceptual space describing shape: the components that make up a shape have no ordering other than the hierarchy already encode elsewhere in the model. The ordering is purely an artefact of the specific definition of trees. However, modifying the definition to use unlabelled sets of trees leads to problems with duplicate nodes, and makes the definitions and proofs involving the tree metric or suborder relation needlessly complicated. An obvious solution to this problem might be to work with isomorphism classes of trees, where isomorphisms are functions that act on trees purely by permuting children of a node. However, this approach is not compatible with the present approach to convexity, since such isomorphism classes are non-convex sets. Another possible solution is to have some normalisation condition on the trees to make sure the ordering is not arbitrary. While such a normalisation condition can be developed quite easily (by defining a total order on the node properties), it is hard to make relations compatible with such a convention, and certainly makes relations much harder to deal with.

In spite of the shortcomings of the model, we believe that the concept of hierarchical conceptual spaces introduced in this dissertation is a fruitful area for future research. Most of the material in this dissertation is quite general with respect to the specifics of the conceptual space being encoded. Thus the tree based model could easily be adopted for other conceptual spaces. The concept of decorating trees also seems promising. In this model the decorations have been used for resolving prepositions. It is not hard to see how a similar approach might be used for resolving references. This would be especially powerful if the model could be adopted to represent (and interact with) context.

Another interesting topic of research would be how to move more of the structure of the conceptual space into a categorical context. In this dissertation, most of the structure is described explicitly in quite a non-categorical manner. Remedying this would likely make the model more general, thus making it easier to apply to other problems. Additionally, it would be interesting from a purely theoretical point of view. In a similar vein, it might be beneficial to use a richer structure then **Rel** to bring the conceptual space into the compositional distributional model. This might, for example, allow us to encode graded entailment and ambiguity.

Bibliography

[AC16] Daniela Ashoush and Bob Coecke. Dual density operators and natural language meaning. *arXiv preprint arXiv:1608.01401*, 2016.

[BCG+16] Josef Bolt, Bob Coecke, Fabrizio Genovese, Martha Lewis, Daniel Marsden, and Robin Piedeleu. Interacting conceptual spaces. *arXiv preprint arXiv:1608.01402*, 2016.

[BCLM16] Desislava Bankova, Bob Coecke, Martha Lewis, and Daniel Marsden. Graded entailment for compositional distributional semantics. *CoRR*, abs/1601.04908, 2016.

[BKS15] Esma Balkir, Dimitri Kartsaklis, and Mehrnoosh Sadrzadeh. Sentence entailment in compositional distributional semantics. *arXiv preprint arXiv:1512.04419*, 2015.

[BSC15] Esma Balkir, Mehrnoosh Sadrzadeh, and Bob Coecke. Distributional sentence entailment using density matrices. In *International Conference on Topics in Theoretical Computer Science*, pages 1–22. Springer, 2015.

[CSC10] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *CoRR*, abs/1003.4394, 2010.

[DWP81] David R Dowty, Robert E Wall, and Stanley Peters. Introduction. In *Introduction to Montague Semantics*, pages 1–13. Springer, 1981.

[Gär00] Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2000.

[Gär14] Peter Gärdenfors. *The geometry of meaning: Semantics based on conceptual spaces*. MIT Press, 2014.

[GS11] Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404. Association for Computational Linguistics, 2011.

[HV17] Chris Heunen and Jamie Vicary. *Categorical Quantum Mechanics: An Introduction*. Department of Computer Science, University of Oxford, 2017.

[Lam08] Joachim Lambek. From word to sentence. *Polimetrica, Milan*, 2008.

[MN78] David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London B: Biological Sciences*, 200(1140):269–294, 1978.

[Pie14] Robin Piedeleu. *Ambiguity in categorical models of meaning*. PhD thesis, University of Oxford Master's thesis, 2014.

[SCC14] Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. The frobenius anatomy of word meanings I: subject and object relative pronouns. *CoRR*, abs/1404.5278, 2014.

[Sch98] Hinrich Schütze. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123, 1998.