

Chapter 4

Meanings

MEANINGS are the topic of this chapter. More precisely, it is abstract meanings that we want to characterize. Unlike what is ordinarily assumed we do not consider the structure of the space of meanings and the functions on them a matter of arbitrary convention. Like with exponents we must ask what meanings actually are and how they can be manipulated.

4.1 “Desyntactified” Meanings

The present chapter is about what meanings *are*. Given the discussion of [Section 3.7](#) we have two kinds of meanings to worry about: concrete meanings and abstract meanings. We shall for the most part consider a calculus of concrete meanings but most of the results are actually independent of which of the two we study. Though much has been made of Putnam’s dictum that meanings (that is, concrete meanings) cannot be in a speaker’s head (Putnam (1975), see also Gärdenfors (2004)), the question whether or not that is so is actually peripheral to the question we are raising, namely, what meanings are and how they can be manipulated. It threatens to focus the debate on questions of factual knowledge rather than principle. Whether or not my concept of gold is the same as that of another person and who has the right concept is a question of factual detail. What matters in this book is what kind of object that concept of mine is and how I use it; and similarly for any other person. Language is therefore subjective, I make no attempt at constructing a language for a community of speakers. Communication is effected only via common expressions and must rely on intersubjective identity (or near identity) in their meaning.

We have said that meanings are given at the outset. It therefore seems to be needless to ask what meanings are, we just look at them. However, there is a larger issue in the background that I cannot adequately treat in this book. The issue is that we cannot access concrete meanings as such; the only thing we can access is particular judgements. We have difficulties saying exactly what defines the concept “book” whereas we seem to be completely reliable in our judgement whether this or that thing is a book. And so there is a legitimate question as to whether the data we can access are the ones we actually need.

While sentences are concrete since we can make them appear on tape or on paper, meanings are not directly observable. There is a long intellectual tradition to assume

that meanings are structured (see King (2007) for a recent exposition). This position is adopted not only in philosophy but also in cognitive linguistics. Unfortunately, it is in practice hard to assess which particular structure the meaning of a given sentence has. In absence of a priori arguments the methodology should be to try to discover that structure from the given data. For it very often happens that our intuitions on meanings are obscured by our own language. What appears to be a semantic fact often enough is just a syntactic (or morphological) fact in disguise. In this way semantics is often infected with syntax. To counteract this trend I shall try to “desyntactify” meanings. (See Erdélyi Szabó, Kálmán, and Kurucz (2007) for a more radical proposal of desyntactification.) In particular, below I shall identify some traits of semantic representations that I consider of purely syntactic nature: hierarchy, order and multiplicity. Hierarchy shows up in the notion of a functional *type*; some meanings are functions that can take objects of certain lower types as arguments. This introduces an asymmetry into meanings that I claim does for the most part not exist in the meanings themselves. Order shows up in the notion of a *tuple*. Predicate logic explicates the meanings of formulae as relations, or sets of tuples. But where exactly the idea of a first member in a tuple or a second member is to be found in the actual denotation is unclear. Finally, although we can repeat a variable, we cannot repeat the same object. It follows that repetition may exist in syntax but not in semantics. We shall look at these problem areas in more detail.

Frege is one of the proponents of the idea that there are “unsaturated” expressions. For example, a function is unsaturated; it yields a value only when given an argument. The function $x^2 + 5$, in conventional notation, does not denote a number. We only get a number when we assign to x some value, say 3. Likewise, Frege argues, many words do not by themselves express a complete thought. They need certain argument places to be filled before this is the case. In this view, the phrase /Ate./ is unsaturated: it lacks a specification of the subject. Thus, only /John ate./ is complete. It is precisely this idea that has been exploited in Montague Grammar and Categorical Grammar. Both of them diagnose this as a syntactic failure that is essentially a type mismatch. Unfortunately, it is unclear whether the incompleteness of /Ate./ is at all a semantic fact. There is an alternative line of analysis, which treats meanings as intrinsically complete (that is, propositional) and instead views the unacceptability of sentences such as /Ate./ as a purely syntactic fact of English. On this view, /Ate./ means “someone was eating something”. There are several reasons why this is a better idea for natural languages. The main one is that the correspondence between semantic arguments and syntactic positions is at best weak. The notion of eating involves both a subject and an object (and a time point, for that matter). An event of eating is constituted minimally by something being eaten and someone eating it. In order to pin down the exact meaning we need to know who ate what when. As it happens, /eat/ can also be used without an object. The standard approach (even in syntactic theory) has been to assume that in this case the sentence contains an empty object. Also, there are ways to convey the same meaning and yet use a fully grammatical construction, such as /There is eating./. What is or is not obligatorily expressed in a sentence varies greatly between languages. Some languages allow the subject to be dropped, for example. Finally and relatedly, the

analogy with functions is misleading in one important respect: while the argument to the function is an object, that is, a thing, the syntactic subject does not necessarily supply one. For should we assume that /John or Mary/ denotes an object that we can feed to the verb, say in /John or Mary ate./? Similarly, /Someone ate./ contains a quantifier in subject position, something that is analysed not as an argument to the verb but rather as a functor. In my view, a syntactic argument serves to specify the identity of some object in question. This specification can be incomplete and thus the function once again lacks any specific value.

Montague has been impressed by the idea that syntactic requirements are at the heart of semantic nature and has consequently endorsed the view that meanings are objects of a typed universe of functions. To implement this we may either choose a universe of the typed λ -calculus or some version of typed combinatory logic. A type is a term of the language with a single binary symbol \rightarrow (you might want more type constructors but this does not change the argument). There is a set of basic types, for example e and t , and one formation rule: If α and β are types, so is $\alpha \rightarrow \beta$. Each type α is associated with a set M_α of denotations. It is generally required that $M_\alpha \cap M_\beta = \emptyset$ whenever $\alpha \neq \beta$. This means that every object has at most one type. Furthermore, we require

$$M_{\alpha \rightarrow \beta} := (M_\beta)^{M_\alpha} := \{f : M_\alpha \rightarrow M_\beta\}. \quad (4.1)$$

This means that we only need to fix the sets M_b for basic b .

At its core Montague Grammar uses only two modes of combination: forward application and backward application.

$$\begin{aligned} A_{>}(\langle \vec{x}, m \rangle, \langle \vec{y}, n \rangle) &= \langle \vec{x} \hat{\ } _ \hat{\ } \vec{y}, m(n) \rangle \\ A_{<}(\langle \vec{x}, m \rangle, \langle \vec{y}, n \rangle) &= \langle \vec{x} \hat{\ } _ \hat{\ } \vec{y}, n(m) \rangle \end{aligned} \quad (4.2)$$

For $A_{>}(\langle \vec{x}, m \rangle, \langle \vec{y}, n \rangle)$ to be defined m must be a function that can take n as its argument. This means that there are α and β such that m is of type $\alpha \rightarrow \beta$ and n of type α . The result is then an object of type β .

Montague Grammar inherits from λ -calculus a number of traits; one is that functions cannot take several arguments simultaneously. A function can only take one argument at a time. This can be eliminated either by allowing simultaneous abstraction or by adding a pair constructor (as in the Lambek Calculus). However, linguists have supported the idea that functions take their arguments one by one. For this means that syntax is binary branching. This has been one of the central arguments in favour of Categorical Grammar. Thus, if we have a predicate with several arguments, we bring it into the desired form by “Currying”, a procedure abstracting the arguments one by one. Additionally, it assumes that when two constituents are concatenated to form a new constituent, the meaning of the result is already determined, at least in the basic calculus. Namely, if two constituents can at all be put together into a single constituent then one of them will have type $\alpha \rightarrow \beta$ and the other the type α ; the result will therefore be of type β . The idea that constituent

formation adds nothing to the meaning is also known as *lexicalism*. In this section I shall propose that rather than using functions we should use relations; and that we should also abandon lexicalism.

The idea of higher order types makes sense only if it is unequivocally clear what is argument and what is function. For if it is an intrinsic property of the meaning of a verb that it takes something as its argument there should be no doubt about this at all. Precisely this, however, has been a problematic issue for Montague Grammar. For on the one hand a singular proposition like “John is sick” is taken to be one where the verb denotes a semantic function taking the subject as its argument. On the other hand, quantified expressions have been argued to be structured in the opposite way: /everyone/ denotes a function in /Everyone is sick./. In order to avoid this mismatch, Montague decided to raise the denotation of /John/ so that it becomes a function over functions. But that was a technical manoeuvre. It was clearly not motivated from semantic considerations but rather from syntactic uniformity. From here, it is a small step towards the type changing operations, which have been used extensively in Landmann (2004). However, they threaten to undermine the idea that we have an intuitive grasp over the semantics of expressions.

Worse, it appears that the idea of the meaning of the syntactic subject as denoting the argument that is supplied to the function is generally unworkable. We can only say that the subject expression predicates *of* that argument. Modern semantics has basically adopted that latter view. However, if that is so, the whole function-argument asymmetry becomes arbitrary. And if we are free to view the subject at one moment as the argument to the verb and at another moment as the function I conclude that the distinction should be dropped altogether. Indeed, some philosophers and linguists have pursued a different semantics. One avenue is event semantics, which has been introduced to overcome not only the rigidity of the typing but also that of predicate logic itself (see Parsons (1994)). (The need to free semantics from syntactic “impositions” is also felt in Minimal Recursion Semantics (see Copestake et al. (2005)). However, the latter is driven purely by concerns of practicality and compensates for the lack of syntactic information by introducing labels. Such approaches, though widespread in computational linguistics do nothing to answer the questions that I have in mind here: namely whether semantics is independent of syntax.) Yet not everyone may be convinced. Therefore, to settle the matter we need empirical criteria. Additionally we need to see if there is a way to replace the typed universe with something else. For if there is not, then this in itself would weaken our position.

The preceding discussion can also be seen in a different light. Even if we grant that the meaning of /eat/ is a function there might be a question as to how that function is used in actual semantics. One camp holds that expressions are basically closed expressions. There are no free variables. One exponent of this view is P. Jacobson. The opposing view is that there is such a thing as free variables and there is no need to quantify them away. Proposals to this effect have been made in Kamp (1981) and Staudacher (1987), among others. The disadvantage of closed expressions is that they make pronominal reference difficult (if not impossible). (But see Jacobson (1999, 2000, 2002) for an opposing view.)

As a consequence, DRT went the opposite way, namely not to abstract away arguments but use formulae instead, with or without free variables. This however comes at a price. For if variables are no longer quantified away we must take proper care of them. There is a standard procedure to eliminate functions from predicate logic. Likewise we shall show here that an approach based on functions can be replaced by one that uses open propositions. An open proposition is a proposition that still needs certain variables to be filled. (These are exactly the “incomplete thoughts”.) Open propositions are the denotations of formulae. A formula is an expression of the form $\varphi(x_0, x_1, \dots, x_{n-1})$ of type t ($=$ truth value), where x_i , $i < n$, are variables of any type. Thus, given an assignment of objects of appropriate type to the variables this expression will yield a truth value. A notable change to previous conceptions of truth, however, is that we consider an open proposition true exactly when it has a satisfying assignment. Thus, /eat/ becomes true exactly when someone is eating something at some moment. This is opposite to the standard conception in logic where an open proposition is considered true if there is no falsifying assignment; so /eat/ would be true if everyone eats everything at every moment. In our approach free variables are inherently existential, in standard predicate logic they are inherently universal. We should note that one problem that besets the free variable approach is that the choice of the actual variable inserted matters for the interpretation of the formula. However, it is patently clear that whether we use x_8 or x_{11} is a matter of convenience. (Fine (2007) has addressed this issue and came to the conclusion that meanings are relational. I will briefly discuss his proposal in Section 4.6.) Thus we have to devise a method to interpret such formulae and manipulate them in such a way that it does not make reference to the actual names of the variables. It is often thought that algebraic semantics has provided a solution to this problem, for example in the proposal by Quine. Here, meanings are relations and there is no talk of variable names. Yet, now we need to talk about positions in a relation, which are not in semantics either. We must namely also make explicit use of substitutions based on indices (see Ben Shalom (1996)). So this does not fully answer the complaint.

There is a well-known procedure to convert all meanings into open propositions. If m is a meaning of type α , $\alpha \neq t$, then replace it with $x = m$, where x is of type α . Consequently, signs of the form $\langle \vec{x}, m \rangle$ are now replaced by signs of the form $\langle \vec{x}, x = m \rangle$. Now consider the rule of application:

$$\mathbf{A}_>(\langle \vec{x}, m \rangle, \langle \vec{y}, n \rangle) = \langle \vec{x} \hat{\ } _ \hat{\ } \vec{y}, m(n) \rangle \quad (4.3)$$

In the new semantics it becomes:

$$\mathbf{U}_>(\langle \vec{x}, u = m \rangle, \langle \vec{y}, v = n \rangle) = \langle \vec{x} \hat{\ } _ \hat{\ } \vec{y}, u = m \wedge v = n \wedge w = u(v) \rangle. \quad (4.4)$$

This is however not always satisfactory. It introduces the idea of applying m to n through the construction; and the construction still speaks of applying m to n . There is an alternative, which runs as follows.

$$\mathbf{U}_>(\langle \vec{x}, u = m(w) \rangle, \langle \vec{y}, v = n \rangle) = \langle \vec{x} \hat{\ } _ \hat{\ } \vec{y}, u = m(w) \wedge v = n \wedge w = v \rangle \quad (4.5)$$

This rule simply conjoins the two meanings and unifies certain variables. The unification, by the way, is the semantic contribution of the rule itself and cannot—on pain of reintroducing the same problematic meanings—be pushed into the meanings of the elements themselves. If $m(w)$ is a function and has to be applied then we also have to feed to $m(w)$ these additional arguments. In this way we can see to it that the generalized rule is as follows.

$$U_{>}^{ij}(\langle \vec{x}, \varphi(\vec{u}) \rangle, \langle \vec{y}, \chi(\vec{v}) \rangle) = \langle \vec{x} \hat{\ } \sqcup \hat{\ } \vec{y}, \varphi(\vec{u}) \wedge \chi(\vec{v}) \wedge u_i = v_j \rangle \quad (4.6)$$

Eliminating the equation we can alternatively write

$$U_{>}^{ij}(\langle \vec{x}, \varphi(\vec{u}) \rangle, \langle \vec{y}, \chi(\vec{v}) \rangle) = \langle \vec{x} \hat{\ } \sqcup \hat{\ } \vec{y}, \varphi(\vec{u}) \wedge [u_i/v_j]\chi(\vec{v}) \rangle. \quad (4.7)$$

Thus we have the following result: the meaning of a complex constituent is a conjunction of the meaning of its parts with some fixed open formula. This is a welcome result. For it says that every meaning is propositional and merging two constituents is conjunction—modulo the addition of some more constraints.

The standard rendering in predicate logic suffers from defects, too. Consider the meaning of /eat/ again. It has, as we agreed, three slots: that of the subject, the object and the time point. When we want to specify any one of the arguments we must know which one that is. If we want to say who is eating we must be able to connect the subject expression with the appropriate subject slot in the predicate. In predicate logic this mechanism is ensured through a linear notation. That there is eating of a sandwich by Steven at noon today is rendered in relational notation as follows.

$$\text{eat}(\text{Steven}, x, 12 : 00) \wedge \text{sandwich}(x) \quad (4.8)$$

Recall that we agreed to read this existentially: it means that there is a value, say $s1$, for x that is a sandwich and such that Steven eats it at 12:00. The order of the three arguments, “Steven”, “ x ” and “12:00” is syntactic: the linear alignment in the formula allows to assign them a particular slot in the relation. One may disagree and claim that it is not the notation that achieves this but rather the denotation: /eat/ denotes a three place relation, which in turn is a set of triples. If this is so then we must ask what reality there is to these triples. In predicate logic, it turns out, they have *no reality*. Compare the following pieces of notation:

$$p(x, y, z) \quad p(\langle x, y, z \rangle) \quad (4.9)$$

On the left we have a ternary predicate p and three arguments. On the right we have a unary predicate p being applied to a single argument, the triple $\langle x, y, z \rangle$. Standard models for predicate logic do not assume that triples exist. It is true that the interpretation of relation symbols is given in the form of sets of tuples but these objects are not part of the domain. Technically, it is possible to install a domain for such tuples; however, that seems to be a mere technical trick we are pulling. The fundamental question to be asked is namely what makes the arguments come in that particular order as opposed to another. I do not know of any reason to put the subject first. But what is the significance of being the first member in the sequence anyway? I know

of no answer to that question. At best, the significance is not objective but rather an artefact of the way we code meanings in predicate logic; this in turn is simply an effect of the language we speak. I am sure that speakers of an OSV language would use a different encoding. But what difference would that make in terms of the meaning as opposed to the encoding? In fact, Dixon (1994) translates Dyirbal verbs in active morphology by their passive counterparts in English. Mel'čuk (1988) goes one step further and says that in Dyirbal the syntactic subject is the object of the corresponding English verb.

Now, if it is possible to systematically exchange the first and the second position in the predicate logic encoding, then we know that what counts is not the actual position. Rather, what is first in one notation is second in the other and vice versa. Thus, if the meanings had these positions in them it should not be possible to exchange the positions in this way. This avenue is to be explored. Suppose we have a language just like English except that in transitive constructions all objects and subjects are exchanged. Such a language is not so outlandish: it would be the consistent ergative counterpart of English. Call this language *Erglish*. Thus, for Dixon, Dyirbal is Erglish though with somewhat different pronunciation. The question is: to what extent can semantics tell the difference between English and Erglish? The answer is: it precisely depends on whether it can tell the difference between being subject and being object. Unless there is a semantic difference, these languages look semantically exactly the same. It therefore appears that if subjects and objects are different we ought to define our semantic rules in terms of this semantic difference rather than using arbitrary labels.

Kit Fine has argued in Fine (2000) that from a metaphysical point of view we should better renounce the idea of a positionalist view of relations. The calculus of concepts below is an attempt to provide such an account. (For a more sophisticated theory of relations see Leo (2010).) It will do more than this, as we believe there is more to the problem. Ultimately, we want to say that a property is true not of a sequence (as in predicate logic) nor of a multiset but rather of a *set* of objects under a particular way of relating the members to a slot. This means that we shall also eliminate *repetitions* in the sequence. It will follow that the concept of self-loving is different from the concept of loving someone else in that the first is unary and the second is binary.

4.2 Predicate Logic

Standard semantic theories assume that meanings are adequately described using predicate logic, first or higher order. Therefore, in this section I shall describe two semantics for (many sorted) predicate logic. This section does not introduce predicate logic as an interpreted language; we leave that topic to [Section 5.1](#). In this section we shall concentrate on standard predicate logic and clarify the basic terminology and definitions.

We assume that basic objects are sortal; we have, for example, *objects*, *time points*, *degrees*, *events*, *situations*, *regions*, *worlds*, *truth values* and so on. For each

sort we assume that the meanings associated with it come from a particular set. Thus we assume that we have a primitive set S of **sorts**. Each sort $s \in S$ is interpreted by a set M_s . Thus we have a family of sets $\mathcal{M} := \{M_s : s \in S\}$. Standardly, it is assumed that the sets M_s and M_t are disjoint whenever $s \neq t$. A **relational type** is a member of S^* , that is, it is a string of sorts. For a relational type \vec{s} , an **object of type** \vec{s} is an element of the set $M_{\vec{s}}$, which is defined inductively as follows.

$$\begin{aligned} M_{\langle \rangle} &:= \{\emptyset\} \\ M_{(s)} &:= M_s \\ M_{\vec{s}-t} &:= M_{\vec{s}} \times M_t \end{aligned} \tag{4.10}$$

Finally, a **relation of type** \vec{s} is a set of objects of type \vec{s} . The type $\langle \rangle$ is of special importance. It corresponds to the set $\{\emptyset\}$. This set has two subsets: $0 := \emptyset$ and $1 := \{\emptyset\}$. These sets will function as our truth values: 1 is for “true” and 0 for “false”. This is achieved by somewhat unorthodox means. A predicate is true in a model if it has a satisfying tuple (see Definition 4.1). Otherwise it is false. Thus, it is true if its extension is not empty and false otherwise. So, denotations of predicates of type \vec{s} are subsets of $M_{\vec{s}}$. Applied to $\vec{s} = \langle \rangle$ this gives the desired correspondence.

I also mention that functions are treated basically as relations; a function of type $\langle s_0, s_1, \dots, s_n \rangle$ is interpreted as follows. Its arguments are of sort s_i , $i < n$ and the value is of sort s_n . It is known that we can eliminate functions from a first-order signature (see Monk (1976)) and so for simplicity we shall assume that there are no functions.

A **first-order (sortal) signature** over a set S of sorts is a pair $\tau = \langle \text{Rel}, \tau \rangle$ such that Rel is a finite set, the set of **relation symbols** and $\tau : \text{Rel} \rightarrow S^*$ an assignment of relational types to relation symbols. All signatures will be finite. The alphabet of PL_τ consists in the following symbols

1. variables x_i^s , where $i \in \mathbb{N}$ and $s \in S$;
2. relation symbols R of type $\tau(R)$;
3. propositional connectives $\wedge, \vee, \rightarrow, \neg$;
4. for each $i \in \mathbb{N}$ and each sort $s \in S$ the quantifiers $\exists x_i^s$ and $\forall x_i^s$.

PL_τ is infinite even if τ is finite. This will require creating a new type, that of an *index*. Indices are generated from a finite alphabet. From these symbols we can form formulae in the following way:

1. If $\vec{s} = \tau(R)$ and \vec{x} is a sequence of variables of type \vec{s} then $R(\vec{x})$ is an **atomic formula**.
2. If φ and χ are formulae, so are $\neg\varphi, \varphi \wedge \chi, \varphi \vee \chi$ and $\varphi \rightarrow \chi$.
3. If φ is a formula and x_s^i a variable then $(\exists x_s^i)\varphi$ and $(\forall x_s^i)\varphi$ is a formula.

Notice that formulae have no type (or, more accurately, are all of the same type). For each $s \in S$ there is an identity $=^s$, which we normally write $=$. Identity is sortal; $x_i^t =^s x_j^u$ is true only if $t = u = s$ (that is, if the sorts are identical). A τ -**structure** is a pair $\mathcal{M} = \langle \mathcal{M}, \mathcal{I} \rangle$, where $\mathcal{M} = \{M_s : s \in S\}$ and for every relation symbol

$R, \mathcal{I}(R)$ is a relation of type $\tau(R)$ over \mathcal{M} , that is, $\mathcal{I}(R) \subseteq M_{\tau(R)}$. An **assignment** into \mathcal{M} or a **valuation** is defined as a function β from the set of variables into $\bigcup \mathcal{M} := \bigcup_{s \in S} M_s$ such that for every $s \in S$: $\beta(x_i^s) \in M_s$. The pair $\langle \mathcal{M}, \beta \rangle$ is called a τ -**model**. Ordinarily, a formula $\varphi(x_0, x_1, \dots, x_{n-1})$ with variables x_i of type s_i is interpreted as a relation of type $\vec{s} := \langle s_0, s_1, \dots, s_{n-1} \rangle$. We shall take a detour via the assignments. Write $[\varphi]_{\mathcal{M}}$ for the set of assignments making a formula φ true. It is defined inductively. For a given assignment β , write $\beta' \sim_{x_i^s} \beta$ if for all $t \neq s$ and all $j \neq i$: $\beta'(x_j^t) = \beta(x_j^t)$. Ass denotes the set of all assignments.

$$\begin{aligned}
[R(\vec{y})]_{\mathcal{M}} &:= \{\beta : \langle \beta(y_0), \beta(y_1), \dots, \beta(y_{n-1}) \rangle \in \mathcal{I}(R)\} \\
[\neg\varphi]_{\mathcal{M}} &:= \text{Ass} - [\varphi]_{\mathcal{M}} \\
[\varphi \wedge \chi]_{\mathcal{M}} &:= [\varphi]_{\mathcal{M}} \cap [\chi]_{\mathcal{M}} \\
[\varphi \vee \chi]_{\mathcal{M}} &:= [\varphi]_{\mathcal{M}} \cup [\chi]_{\mathcal{M}} \\
[\varphi \rightarrow \chi]_{\mathcal{M}} &:= (\text{Ass} - [\varphi]_{\mathcal{M}}) \cup [\chi]_{\mathcal{M}} \\
[(\exists x_i^s)\varphi]_{\mathcal{M}} &:= \{\beta : \text{there is } \beta' \sim_{x_i^s} \beta : \beta' \in [\varphi]_{\mathcal{M}}\} \\
[(\forall x_i^s)\varphi]_{\mathcal{M}} &:= \{\beta : \text{for all } \beta' \sim_{x_i^s} \beta : \beta' \in [\varphi]_{\mathcal{M}}\}
\end{aligned} \tag{4.11}$$

This formulation makes predicate logic amenable to the treatment of this book. Standardly, however, one prefers a different formulation. Let β be a valuation and φ a formula. Then say that φ is **true in \mathcal{M} under the assignment β** and write $\langle \mathcal{M}, \beta \rangle \models \varphi$, if $\beta \in [\varphi]_{\mathcal{M}}$. This notion is defined inductively by

$$\begin{aligned}
\langle \mathcal{M}, \beta \rangle \models \varphi(\vec{x}) &:\Leftrightarrow \beta(\vec{x}) \in \mathcal{I}(R) \\
\langle \mathcal{M}, \beta \rangle \models \neg\varphi &:\Leftrightarrow \text{not } \langle \mathcal{M}, \beta \rangle \models \varphi \\
\langle \mathcal{M}, \beta \rangle \models \varphi \wedge \chi &:\Leftrightarrow \langle \mathcal{M}, \beta \rangle \models \varphi \text{ and } \langle \mathcal{M}, \beta \rangle \models \chi \\
\langle \mathcal{M}, \beta \rangle \models \varphi \vee \chi &:\Leftrightarrow \langle \mathcal{M}, \beta \rangle \models \varphi \text{ or } \langle \mathcal{M}, \beta \rangle \models \chi \\
\langle \mathcal{M}, \beta \rangle \models \varphi \rightarrow \chi &:\Leftrightarrow \langle \mathcal{M}, \beta \rangle \not\models \varphi \text{ or } \langle \mathcal{M}, \beta \rangle \models \chi \\
\langle \mathcal{M}, \beta \rangle \models (\exists y)\varphi &:\Leftrightarrow \text{for some } \beta' \sim_y \beta : \langle \mathcal{M}, \beta' \rangle \models \varphi \\
\langle \mathcal{M}, \beta \rangle \models (\forall y)\varphi &:\Leftrightarrow \text{for all } \beta' \sim_y \beta : \langle \mathcal{M}, \beta' \rangle \models \varphi
\end{aligned} \tag{4.12}$$

For a formula φ the set of **free variables**, $\text{fr}(\varphi)$, is defined as follows.

$$\begin{aligned}
\text{fr}(R(\vec{y})) &:= \{y_i : i < \text{length}(\tau(R))\} \\
\text{fr}(\neg\varphi) &:= \text{fr}(\varphi) \\
\text{fr}(\varphi \wedge \chi) &:= \text{fr}(\varphi) \cup \text{fr}(\chi) \\
\text{fr}(\varphi \vee \chi) &:= \text{fr}(\varphi) \cup \text{fr}(\chi) \\
\text{fr}(\varphi \rightarrow \chi) &:= \text{fr}(\varphi) \cup \text{fr}(\chi) \\
\text{fr}((\exists y)\varphi) &:= \text{fr}(\varphi) - \{y\} \\
\text{fr}((\forall y)\varphi) &:= \text{fr}(\varphi) - \{y\}
\end{aligned} \tag{4.13}$$

Proposition 4.1 (Coincidence Lemma) *Let β and β' be valuations such that for all $y \in \text{fr}(\varphi)$ $\beta(y) = \beta'(y)$. Then $\langle \mathcal{M}, \beta \rangle \models \varphi$ iff $\langle \mathcal{M}, \beta' \rangle \models \varphi$. Alternatively, $\beta \in [\varphi]_{\mathcal{M}}$ iff $\beta' \in [\varphi]_{\mathcal{M}}$.*

A **theory** (or **deductively closed set**) in the signature τ is a set of formulae $T \subseteq L_{\tau}$ such that

for every formula φ and every formula χ : if $\varphi \rightarrow \chi \in T$ and $\varphi \in T$, then $\chi \in T$.

There is a calculus for predicate logic, whose nature we shall not elucidate (however, see Monk (1976) or Rautenberg (2006)). It defines in syntactic terms a relation $\Delta \vdash \varphi$ between sets Δ of formulae and a single formula φ . If $\Delta \vdash \varphi$, we say that φ is **derivable** from Δ . With respect to this calculus, we say that T is **consistent** if for $\perp := (\exists x_0^s) \neg (x_0^s = x_0^s)$ (any choice of s) we do *not* have $T \vdash \perp$.

Theorem 4.1 (Completeness of Predicate Logic) *For every consistent theory T there is a model \mathcal{M} and a valuation β such that for all $\delta \in T$: $\langle \mathcal{M}, \beta \rangle \models \delta$.*

An alternative to sets of assignments are finitary relations. Since this gets us closer to our final interpretation (via concepts), let us see how this approach might go. We assume that we have a slightly different enumeration of the variables as before. Instead of enumerating the variables of each sort separately, we enumerate *all* variables in one infinite list. The set of variables of all sorts is $\text{Var} := \{x_i : i \in \mathbb{N}\}$. Each of the x_i has its sort, s_i , which we leave implicit in the notation. For every formula φ we define the meaning to be a relation $(\varphi)_{\mathcal{M}}$. Before we specify the precise nature of this relation we shall introduce an idea by Kleene. Let the syntactic objects be pairs (φ, \vec{x}) , where φ is a formula and \vec{x} a sequence of variables. Then we let its denotation be the set of all tuples \vec{a} of the same type as \vec{x} such that there is a valuation that satisfies φ and sends x_i to a_i . For example, $(x_0 + x_1 = x_3, x_0)$ is a syntactic object and denotes over the natural numbers the set $\{(i) : i \in \mathbb{N}\}$; $(x_0 + x_1 = x_3, x_0, x_3)$ is a syntactic object and it denotes the set $\{(i, j) : i \leq j\}$. Finally, $(x_0 + x_1 = x_3, x_0, x_3, x_1)$ denotes the set $\{(i, j, k) : i + k = j\}$. Call a syntactic object (φ, \vec{x}) **complete** if every free variable of φ is in \vec{x} . (We may or may not disallow repetition of variables.) It is possible to give a compositional semantics for complete syntactic objects (see the exercises).

The problem with predicate logic is that our strings are not pairs of formulae and variables. But there is in fact no need to assume that. Namely, all we need to assume is a canonical linear order on the variables. We then assume that the meaning of the formula φ is what the meaning of (φ, \vec{x}) is, where \vec{x} is a specific set containing the set of free variables of φ in canonical order. The sequence we choose here is $\langle x_0, x_1, \dots, x_{n-1} \rangle$ where x_{n-1} is the highest free variable of φ . (Notice that the variables x_i with $i < n - 1$ need not occur free in φ .) Thus the relation codes the assignment of the first n variables x_i , $i < n$, in the following way. For a valuation β we define the partialization $\beta_n := \beta \upharpoonright \text{Var}_n$, where $\text{Var}_n = \{x_i : i < n\}$ for some n . We translate the valuation γ into a sequence

$$(\beta_n)^\heartsuit := \langle \beta_n(x_i) : i < n \rangle \in \mathbf{X}_{i < n} M_{s_i}. \quad (4.14)$$

Let $\ell(\varphi)$ be the largest number such that $x_{\ell(\varphi)-1} \in \text{fr}(\varphi)$. Then put

$$\llbracket \varphi \rrbracket_{\mathcal{M}} := \{(\beta_{\ell(\varphi)})^\heartsuit : \beta \in [\varphi]_{\mathcal{M}}\}. \quad (4.15)$$

Clearly,

$$\llbracket \varphi \rrbracket_{\mathcal{M}} \subseteq \mathbf{X}_{i < n} M_{s_i}. \quad (4.16)$$

Now, instead of defining $\llbracket \varphi \rrbracket_{\mathcal{M}}$ via the set of satisfying valuations we can also give an inductive definition. Let $R^{\rightarrow k}$ be the expansion of R to a k -ary relation. This is defined as follows. (a) $R^{\rightarrow 0} := R$. (b) If k is less than the length of R then $R^{\rightarrow k+1} := R$. (c) If k is at least the length of R then $R^{\rightarrow k+1} := (R^{\rightarrow k}) \times M_{s_k}$, where s_k is the sort of x_k . For a tuple \vec{a} let $[i : b]\vec{a}$ denote the result of replacing a_i by b . $\vec{a} \cdot b$ denotes \vec{a} with b added at the end. Given a relation R of length n , put

$$\mathbf{C}_i.R := \begin{cases} R & \text{if } i \geq n, \\ \{\vec{a} : \text{there is } b \in M_{s_i} \text{ such that } \vec{a} \cdot b \in R\} & \text{if } i = n - 1, \\ \{\vec{a} : \text{there is } b \in M_{s_i} \text{ such that } [i : b]\vec{a} \in R\} & \text{else.} \end{cases} \quad (4.17)$$

Notice that in case $i = n - 1$ the relation gets contracted. Cylindrification yields a relation of length $n - 1$ in this case. Finally, let Ω^k be the total relation of length k .

$$\begin{aligned} \llbracket R(x_{i_0}, \dots, x_{i_{n-1}}) \rrbracket_{\mathcal{M}} &:= \{\vec{a} : \langle a_{i_0}, \dots, a_{i_{n-1}} \rangle \in \mathcal{I}(R)\} \\ \llbracket \neg \varphi \rrbracket_{\mathcal{M}} &:= \Omega^{\ell(\varphi)} - \llbracket \varphi \rrbracket_{\mathcal{M}} \\ \llbracket \varphi \wedge \chi \rrbracket_{\mathcal{M}} &:= \llbracket \varphi \rrbracket_{\mathcal{M}}^{\rightarrow \ell(\chi)} \cap \llbracket \chi \rrbracket_{\mathcal{M}}^{\rightarrow \ell(\varphi)} \\ \llbracket \varphi \vee \chi \rrbracket_{\mathcal{M}} &:= \llbracket \varphi \rrbracket_{\mathcal{M}}^{\rightarrow \ell(\chi)} \cup \llbracket \chi \rrbracket_{\mathcal{M}}^{\rightarrow \ell(\varphi)} \\ \llbracket \varphi \rightarrow \chi \rrbracket_{\mathcal{M}} &:= \left(\Omega^{\ell(\chi)} - \llbracket \varphi \rrbracket_{\mathcal{M}}^{\rightarrow \ell(\chi)} \right) \cup \llbracket \chi \rrbracket_{\mathcal{M}}^{\rightarrow \ell(\varphi)} \\ \llbracket (\exists x_i) \varphi \rrbracket_{\mathcal{M}} &:= \mathbf{C}_i.\llbracket \varphi \rrbracket_{\mathcal{M}} \end{aligned} \quad (4.18)$$

Example 4.1 It is worthwhile to mention a few facts about how we intend to use this for natural language. First, we assume that the denotation of expressions is a relation of some sort. To make this come about, we must eliminate all functions and constants. This technique is known (see Monk (1976)). We show some cases. The denotation of /John/ is the set of things being identical to John; we can represent this by the formula $x = j$, where j is the constant denoting John. There is no saturation; merge corresponds to conjunction. The sentence ‘‘John left.’’ contains two pieces whose meaning we can paraphrase as ‘‘someone is John’’ and ‘‘someone left’’. The syntagma adds the meaning that the two people are the same. \odot

In order to implement the previous idea it is necessary to revise our notion of satisfaction.

Definition 4.1 We write $\mathcal{M} \models \varphi(\vec{x})$ and say that $\varphi(\vec{x})$ is **true** in \mathcal{M} if there is some valuation β such that $\langle \beta(x_i) : i < n \rangle \in \langle \varphi \rangle_{\mathcal{M}}$.

For comparison we shall say a few words about the type theoretic interpretation chosen by Montague. Instead of using “flat” types (which we call sorts) he introduces a hierarchy as follows (compare also Section 4.1). A **functional type** is (a) either a basic type or (b) a sequence $\rightarrow s_0 s_1$ where s_0 and s_1 are functional types. We use variables α, β to denote functional types and also write $\alpha \rightarrow \beta$ rather than using Polish Notation, to keep within the standard notation. We associate with $\alpha \rightarrow \beta$ the set of all functions from M_α to M_β . Montague uses e for *objects* and t for *truth values*. A relational type $\langle s_0, s_1, \dots, s_{n-1} \rangle$ is coded as the functional type

$$s_0 \rightarrow (s_1 \rightarrow (\dots \rightarrow (s_{n-1} \rightarrow t))) \quad (4.19)$$

This allows to dispense with the original “flat” types.

Exercise 4.1 Prove the Coincidence Lemma (Proposition 4.1).

Exercise 4.2 Spell out a compositional approach to the semantics of complete syntactic objects. (You may consult Section 5.1 on this but the solution should be clear anyhow.)

Exercise 4.3 Show that there is no compositional semantics for syntactic objects in general. (So, dropping the completeness requirement will not work.)

Exercise 4.4 Give an example to show why the semantics $\langle \varphi \rangle_{\mathcal{M}}$ cannot simply be based on the pairs (φ, \vec{x}) where \vec{x} is exactly the set of free variables of φ in canonical order.

4.3 Concepts

Standard semantic theories assume that meanings are adequately described using predicate logic, first or higher order. In this section, however, I shall sketch a different theory of meaning, which is based on *concepts*. A concept is a set of relations that are in some sense variants of each other. A relation is a variant of another relation if it can be obtained either by permutation of its arguments or by contracting or expanding it. A precise definition is as follows.

Let $\vec{s} = \langle s_0, s_1, \dots, s_{n-1} \rangle$ be a type and $\pi : n \rightarrow n$ be a permutation. Then $\pi(\vec{s}) := \langle s_{\pi(0)}, s_{\pi(1)}, \dots, s_{\pi(n-1)} \rangle$ is a **permutation** of \vec{s} . If $t \in S$ then $\vec{s} \cdot t$ is an **expansion** of \vec{s} . Given a relation R of type \vec{s} , define

$$\pi[R] := \{ \pi(\vec{x}) : \vec{x} \in R \}. \quad (4.20)$$

This is a relation of type $\pi(\vec{s})$. A relation R' is said to be a **permutation** of R if and only if it is of the form $\pi[R]$ for some permutation π . Furthermore, let

$$E(R) := \{ \langle x_0, x_1, \dots, x_{n-1}, x_{n-1} \rangle : \langle x_0, x_1, \dots, x_{n-1} \rangle \in R \}. \quad (4.21)$$

This is a relation of type $\vec{s} \cdot s_{n-1}$. A relation R' is said to be a **diagonal expansion** of R if and only if it has the form $E(R)$. Finally, set

$$P_t(R) := R \times M_t. \quad (4.22)$$

This is a relation of type $\vec{s} \cdot t$. A relation is said to be a **product expansion of R** (with type t) if and only if it has the form $P_t(R)$.

Definition 4.2 R' is an **immediate variant** of R if and only if R' is either a permutation of R or R' is a diagonal expansion of R or R is a diagonal expansion of R' or R' is a product expansion of R or R is a product expansion of R' . R' is a **variant** of R if there is a series $\langle R_i : i < n + 1 \rangle$ such that $R_0 = R$, $R_n = R'$ and for each $i < n$, R_{i+1} is an immediate variant of R_i . We write $R \sim R'$ if R' is a variant of R .

The relation of variance is an equivalence relation. It is clearly transitive and reflexive (choose $n = 0$ in the definition), and it is symmetric because it is the transitive and reflexive closure of a symmetric relation.

Example 4.2 Let $S := \{\ell, n\}$, $M_\ell := \{a, b, c\}$ and $M_n := \{0, 1\}$. The relation $R = \{\langle a, 0 \rangle, \langle b, 1 \rangle\}$ is of type $\langle \ell, n \rangle$. It has a nonidentical permutation $R' = \{\langle 0, a \rangle, \langle 1, b \rangle\}$. This is also known as the **converse** of R and written R^\smile . The diagonal expansion of R is $E(R) := \{\langle a, 0, 0 \rangle, \langle b, 1, 1 \rangle\}$. The diagonal expansion of R' is $E(R') = \{\langle 0, a, a \rangle, \langle 1, b, b \rangle\}$. \clubsuit

Even though the diagonal expansion repeats only the last column, R has many more variants. Write

$$E_i(R) := \{\langle x_0, x_1, \dots, x_{n-1}, x_i \rangle : \langle x_0, x_1, \dots, x_{n-1} \rangle \in R\}. \quad (4.23)$$

Then $E_i(R)$ is a variant of R . Namely, let $\pi = (i \ n - 1)$ (see [Appendix A](#) for notation) and $\pi' = (i \ n)$. These are the permutations that exchange the items number i and $n - 1$ in the case of π and i and n in the case of π' . Then

$$E_i(R) = \pi'[E(\pi[R])]. \quad (4.24)$$

We say that R' is a **generalized diagonal expansion** of R if $R' = E_i(R)$ for some i . Likewise, the generalized product expansion is defined by

$$P_t^i(R) := \{\langle x_0, x_1, \dots, x_{n-1}, x_n \rangle : \langle x_0, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n \rangle \in R, x_i \in M_t\}. \quad (4.25)$$

Notice the following. The identity relation of type $\langle s, s \rangle$ is defined as

$$\{\langle x, x \rangle : x \in M_s\}. \quad (4.26)$$

This is a diagonal expansion of type s of the total relation M_s of type $\langle s \rangle$. This in turn is a product expansion of the relation $M_{\langle \rangle} = \{\emptyset\} = 1$. Thus the identity relation

is a variant of the “true” relation. This has consequences we shall look at in more detail later.

Definition 4.3 A **concept** is a set of relations of the form $\llbracket R \rrbracket := \{R' : R' \sim R\}$. Concepts are denoted by small Gothic letters: \mathfrak{c} , \mathfrak{d} . For a set M (or a structure \mathcal{M}), the set of concepts over M (\mathcal{M}) is denoted by $\text{Conc}(M)$ ($\text{Conc}(\mathcal{M})$).

Notice that this is well-defined since variance is an equivalence relation. In principle we should write $\llbracket R \rrbracket_{\mathcal{M}}$ since the concept depends on the structure; however, I shall mostly drop the reference to the structure since it will always be clear from the context. There are two special concepts: the *verum* concept, denoted by \mathfrak{t} and the *falsum* concept, denoted by \mathfrak{f} . We have

$$\mathfrak{t} := \llbracket \{\emptyset\} \rrbracket, \quad \mathfrak{f} := \llbracket \emptyset \rrbracket. \quad (4.27)$$

We employ the following convention. For a set M we take M to be the same as $1 \times M$, where $1 = \{\emptyset\}$. Thus, if M_s is the domain of elements of type s , since M_s and $1 \times M_s$ count as the same, the set (= relation) M_s is a variant of 1. This is to be kept in mind. $M^1 = \{\langle x \rangle : x \in M\}$ is technically different from M but considered here the same object.

Example 4.3 Let us look at a universe consisting in just one element, a . The concept generated by the empty relation is of course just the set $\{\emptyset\}$. This is the falsum concept. The verum concept is the concept of the form $\mathfrak{t} = \llbracket \{\emptyset\} \rrbracket$. These are the only concepts. Let R be a nonempty relation. Then it has the form $\{\langle a, a, \dots, a \rangle\}$. Any two such sets are variants of each other. For example, $\{\langle a, a, a \rangle\}$ is a variant of $\{\langle a, a \rangle\}$ (being both a diagonal and a product expansion), which in turn is a variant of $\{\langle a \rangle\}$. The latter is a variant of 1. Thus, every nonempty relation is a variant of every other nonempty relation but not a variant of the empty relation. So, $\text{Conc}(\{a\}) = \{\mathfrak{t}, \mathfrak{f}\}$. \otimes

Example 4.4 We shall describe the concepts over a two element universe $M := \{a, b\}$ (only one sort, with extension M). We shall only look at concepts generated by at most binary relations. The zeroary relations are \emptyset and $\{\emptyset\}$, generating the concepts \mathfrak{t} and \mathfrak{f} . The unary relations are \emptyset , $\{\langle a \rangle\}$, $\{\langle b \rangle\}$, $M = \{\langle a \rangle, \langle b \rangle\}$. The first and the last are variants of zeroary relations, so we effectively have only two new members, $\{\langle a \rangle\}$ and $\{\langle b \rangle\}$. Next we turn to binary relations. Here is a list of all 16:

$$\begin{array}{ll} R_1 := \emptyset & R_9 := \{\langle a, b \rangle, \langle b, a \rangle\} \\ R_2 := \{\langle a, a \rangle\} & R_{10} := \{\langle a, b \rangle, \langle b, b \rangle\} \\ R_3 := \{\langle a, b \rangle\} & R_{11} := \{\langle b, a \rangle, \langle b, b \rangle\} \\ R_4 := \{\langle b, a \rangle\} & R_{12} := \{\langle a, a \rangle, \langle a, b \rangle, \langle b, a \rangle\} \\ R_5 := \{\langle b, b \rangle\} & R_{13} := \{\langle a, a \rangle, \langle a, b \rangle, \langle b, b \rangle\} \\ R_6 := \{\langle a, a \rangle, \langle a, b \rangle\} & R_{14} := \{\langle a, a \rangle, \langle b, a \rangle, \langle b, b \rangle\} \\ R_7 := \{\langle a, a \rangle, \langle b, a \rangle\} & R_{15} := \{\langle a, b \rangle, \langle b, a \rangle, \langle b, b \rangle\} \\ R_8 := \{\langle a, a \rangle, \langle b, b \rangle\} & R_{16} := \{\langle a, a \rangle, \langle a, b \rangle, \langle b, a \rangle, \langle b, b \rangle\} \end{array} \quad (4.28)$$

R_1 and R_{16} are variants of \emptyset and $\{\emptyset\}$, respectively. R_2 and R_5 are diagonal expansions of $\{\langle a \rangle\}$ and $\{\langle b \rangle\}$, respectively. R_3 and R_4 are permutations of each other. R_6 is $\{\langle a \rangle\} \times M$, so it is a variant of $\{\emptyset\}$; R_7 is a permutation of R_6 . R_8 is the identity on M , hence in turn a variant of verum. R_9 is symmetric; it generates a concept different from the previous. R_{10} and R_{11} are diagonal expansions of unary relations. R_{12} , R_{13} and R_{15} are essentially new, while R_{14} is a variant of R_{13} . Thus, up to variance, there are only six relations: R_3 , R_6 , R_9 , R_{12} , R_{13} and R_{15} . \otimes

Notice that the empty set is the empty n -ary relation for every n . It thus plays multiple roles. This is not so for concepts. The empty concept has length 0 (see below for a definition). The empty binary relation generates the empty concept, just as any other empty relation, since they are the same set.

It is to be noted that the identity concept is nothing but verum, a welcome consequence of the calculus. For it is the diagonal relation $\Delta_M := \{\langle a, a \rangle : a \in M\}$. This set is the diagonal expansion of M , which is a product expansion of 1. Hence identity is a variant of 1 and therefore generates the concept ι . This reflects the fact that self-identity is trivially true of everything. To say that an object is identical to itself is to issue a mere triviality. For this it does not matter whether or not we take identity to be sortal. For example, the sortal diagonal $\Delta_s := \{\langle a, a \rangle : a \in M_s\}$ is a diagonal expansion of M_s , which is an expansion of 1.

Let us now investigate the structure of the concept space somewhat.

Definition 4.4 The **length** of a relation R is the length of any member of R . Let \mathfrak{c} be a concept. A relation $R \in \mathfrak{c}$ is **minimal** in \mathfrak{c} if it is of minimal length among all members of \mathfrak{c} . The **length** of \mathfrak{c} is the length of any minimal member of \mathfrak{c} . The length of \mathfrak{c} is denoted by $\ell(\mathfrak{c})$.

Minimal relations obviously exist; moreover, they are in an important sense unique. For the purpose of the next proof, say in a relation R column i is **independent** if for every tuple $\vec{a} \in R$ and $b \in M_s$ of the appropriate sort s we have $[i : b]\vec{a} \in R$. Say that column i is a replica of column j if the columns have the same sort s and for every tuple $\vec{a} \in R$ we have $a_i = a_j$.

Proposition 4.2 Let R and R' be minimal members of a concept \mathfrak{c} . Then R is a permutation of R' .

Proof We assume here that M_s has at least two members for each sort. (This just eliminates trivial cases; for a one-element set is always redundant in a minimal member.) Let R be a minimal relation of length n . Call an n -**sequence** a sequence \vec{o} over the set $\{\star_0, \star_1, \dots, \star_{n-1}\} \cup \{o_s : s \in S\}$. \vec{o} is **full** if every \star_i , $i < n$, occurs at least once. For each $s \in S$ choose some $y_s \in M_s$. Let \vec{o} be of length k . Given $\vec{a} \in R$ we can assign an element $\vec{o}(\vec{a})$ as follows.

$$\vec{o}(\vec{a}) = \langle o_0(\vec{a}), o_1(\vec{a}), \dots, o_{k-1}(\vec{a}) \rangle \quad (4.29)$$

where

$$o_i(\vec{a}) := \begin{cases} a_j & \text{if } o_i = \star_j, \\ y_s & \text{if } o_i = \circ_s. \end{cases} \quad (4.30)$$

By induction, we shall assign an n -sequence to all variants R' of R . These sequences will be full, as can easily be checked. Moreover, inductively it is checked that \vec{o} is an embedding of R into R' (fullness is essential here). When R' is minimal, its sequence is of length n , consisting in the \star_i in some permutation. So R' is a permutation of R . And this then concludes the proof.

R is assigned the sequence $\langle \star_0, \star_1, \dots, \star_{n-1} \rangle$. Assume that R' has the sequence \vec{o} and that R'' is an immediate variant of R' . Case 1. R'' is a permutation of R' via π . Assign to R'' the sequence $\pi(\vec{o})$. If \vec{o} is full then so is $\pi(\vec{o})$. The map $\pi(\vec{o})$ is an embedding. Case 2. R'' is a diagonal expansion of R' . Then assign to R'' the sequence $\vec{o} \cdot o$, where o is the last member of \vec{o} . If \vec{o} is full so is the sequence $\vec{o} \cdot o$. Case 3. R'' is a product expansion of R' by sort s . Then assign to R'' the sequence $\vec{o} \cdot \circ_s$. If \vec{o} is full so is $\vec{o} \cdot \circ_s$. Case 4. R' is a product expansion of R'' by sort s . Two cases need to be considered. The first is that $\vec{o} = \vec{m} \cdot \circ_s$. Then assign \vec{m} to R'' . The second case is where the last member is \star_i for some $i < n$. This case never arises. For either \vec{m} contains \star_i and then the last column is a replica, contradicting its independence. Or \vec{m} does not contain \star_i . And this would mean that the i th column of R is independent of the other columns. In other words, R would not be minimal. Contradiction. So, \vec{m} is full and defines an embedding. Case 5. R' is a diagonal expansion of R'' . Then either $\vec{o} = \vec{m} \cdot \star_i$ for some $i < n$ or $\vec{o} = \vec{m} \cdot \circ_s$ for some $s \in S$. R'' will be assigned the sequence \vec{m} in both cases. \vec{m} is also full, since the last member of \vec{o} also occurs in \vec{m} if it is of the form \star_i . Suppose the last member is \star_i for some $i < n$. Again, being an expansion of R'' the last column is either independent of the other columns (which would contradict the minimality of R) or it repeats some other column of R'' , say column h . Then o_h is either \circ_s or \star_j . In the second case the j th column of R would be a replica of the i th column, so R is not minimal, unless $j = i$. But then \vec{m} is full. In the first case column k is independent of the i th column of R and so cannot depend on the last column. \square

The proof reveals that the concept allows to define the generating relation up to a permutation on condition that the generating relation is nonreducible, that is, cannot be obtained from another relation by expansion.

Lemma 4.1 *Let R, R' be minimal members of \mathfrak{c} . If $R \subseteq R'$ then $R = R'$.*

Proof Suppose that $R \subseteq R'$. By the previous theorem, $R' = \pi[R]$ for some permutation π . So, $R \subseteq \pi[R]$. From this we derive $\pi^i[R] \subseteq \pi^{i+1}[R]$ for any i and by transitivity, $R \subseteq \pi^i[R]$ for any i . Now, since there is a k such that π^k is the identity, we can also derive $\pi^{k-1}[R] \subseteq \pi^k[R] = R$ and reasoning backwards establish that $\pi^i[R] \subseteq R$ for all $i < k$. It follows that $R' = \pi[R] \subseteq R$. \square

We can use this to define the type of a concept. Suppose \mathfrak{c} is a concept and that $R \in \mathfrak{c}$ is minimal. Then R has a type \vec{s} . This is a sequence. It defines a multiset $\S(\vec{s})$ in the following way: the sort s is contained in $\S(\vec{s})$ exactly as many times as it is contained in \vec{s} . Thus we say that $\S(\vec{s})$ is the **type** of \mathfrak{c} .

We define the following subsumption relation on concepts.

$$c \leq d :\Leftrightarrow (\forall R \in c)(\exists S \in d)(R \subseteq S) \quad (4.31)$$

Notice that $R \subseteq S$ means that the relations are of same length and type. It turns out that just one pair of sets is sufficient to establish an order between the concepts.

Lemma 4.2 $c \leq d$ if and only if there is $R \in c$ and $S \in d$ such that $R \subseteq S$.

Proof From left to right is clear. So assume that there exists $R \in c$ and $S \in d$ such that $R \subseteq S$. Let π be a permutation. Then $\pi[R] \subseteq \pi[S]$. Also, $R \times M \subseteq S \times M$ and $E(R) = E(S)$. So for any permutation and expansion of R there is a corresponding set in d . If however R is itself an expansion of T then $T = C_i.R$ for some i . Now, $C_i.R \subseteq C_i.S$. Hence for all $R' \sim R$ there is a $S' \sim S$ such that $R' \subseteq S'$. \square

Proposition 4.3 \leq is an ordering relation. That is to say for all c, d and e :

- ① $c \leq c$.
- ② If $c \leq d$ and $d \leq e$ then $c \leq e$.
- ③ If $c \leq d$ and $d \leq c$ then $c = d$.

Proof ① is clear. For ②, suppose $R \in c$. Then by assumption there is $S \in d$ such that $R \subseteq S$; again by assumption there is a $T \in e$ such that $S \subseteq T$. So, $R \subseteq T$ for some $T \in e$. It follows that $c \leq e$. For ③ let R be minimal in c . Assume first that there is a minimal $S \in d$ such that $R \subseteq S$. Then by assumption there is a $R' \in c$ such that $S \subseteq R'$. Since $R \subseteq R'$ and both are of the same length, R' is not only minimal (by Proposition 4.2), we also have $R = R'$, by Lemma 4.1. It follows that $R = S$ and $c = d$. Now suppose that there is no minimal S such that $R \subseteq S$. Then d has smaller length than c , for there is at least one S of length $\ell(c)$ in d . Hence $\ell(d) < \ell(c)$. Now pick a minimal $S \subseteq d$. There is no $R \in c$ for which $S \subseteq R$, contrary to assumption. \square

The concatenation of concepts plays the role of conjunction.

Definition 4.5 Suppose that $c = \llbracket R \rrbracket$ and $d = \llbracket S \rrbracket$. Then we define

$$c * d := \llbracket S \times R \rrbracket. \quad (4.32)$$

This definition does not depend on representatives. We omit the proof. Notice that even if R is minimal in c and S is minimal in d , $R \times S$ need not be minimal in $c * d$. This is easily seen if $c = d$.

Proposition 4.4 $*$ is a semilattice operation on $\text{Conc}(M)$. This means that for all c, d and e :

- ① $c * c = c$.
- ② $c * d = d * c$.
- ③ $c * (d * e) = (c * d) * e$.

Proof Let $\mathfrak{c} = \llbracket R \rrbracket$, $\mathfrak{d} = \llbracket S \rrbracket$ and $\mathfrak{e} = \llbracket T \rrbracket$. Then, as $R \times R \sim R$ (using a series of diagonal expansions), we have $\llbracket R \times R \rrbracket = \llbracket R \rrbracket = \mathfrak{c}$. Further, since $R \times S \sim S \times R$ (using a suitable permutation) we have $\mathfrak{c} * \mathfrak{d} = \mathfrak{d} * \mathfrak{c}$. Finally, $(\mathfrak{c} * \mathfrak{d}) * \mathfrak{e} = \llbracket (R \times S) \times T \rrbracket = \llbracket R \times (S \times T) \rrbracket = \mathfrak{c} * (\mathfrak{d} * \mathfrak{e})$. \square

The concatenation is a kind of conjunction. It represents the conjunction without any identification. In fact we can show that under the ordering \leq defined above, $*$ is exactly the greatest lower bound.

Proposition 4.5 *$*$ is the greatest lower bound in $\langle \text{Conc}(M), \leq \rangle$. This means that*

- $\mathfrak{c} * \mathfrak{d} \leq \mathfrak{c}$ and $\mathfrak{c} * \mathfrak{d} \leq \mathfrak{d}$;
- for every \mathfrak{e} such that $\mathfrak{e} \leq \mathfrak{c}$ and $\mathfrak{e} \leq \mathfrak{d}$ we also have $\mathfrak{e} \leq \mathfrak{c} * \mathfrak{d}$.

Proof The first assumption follows from the second. Assume therefore $\mathfrak{e} \leq \mathfrak{c}$ and $\mathfrak{e} \leq \mathfrak{d}$ for some \mathfrak{e} . Pick $R \in \mathfrak{e}$. There is then $S \in \mathfrak{c}$ and $T \in \mathfrak{d}$ such that $R \subseteq S$ and $R \subseteq T$. Let R be of length n . Define the set R^{\triangleright} as follows.

$$R^{\triangleright} := \{\langle a_0, \dots, a_{n-1}, a_0, \dots, a_{n-1} \rangle : \langle a_0, \dots, a_{n-1} \rangle \in R\} \quad (4.33)$$

$R^{\triangleright} \sim R$ (by repeated generalized diagonal expansion). Moreover, $R^{\triangleright} \subseteq S \times T$. By Lemma 4.2, $\mathfrak{e} \leq \mathfrak{c} * \mathfrak{d}$. \square

There is no natural definition of disjunction, since this needs identification of columns. We leave it to the next section to go deeper into the topic of identification of columns across concepts.

As we have explained in Section 4.1, we claim that natural language meanings are not sets of assignments but rather concepts. For a formula φ of predicate logic we put

$$\langle\langle \varphi \rangle\rangle_{\mathcal{M}} := \llbracket \langle\langle \varphi \rangle\rangle_{\mathcal{M}} \rrbracket_{\mathcal{M}}. \quad (4.34)$$

Recall that $\langle\langle \varphi \rangle\rangle_{\mathcal{M}}$ delivers a relation (a subset of $\prod_{i < \ell(\varphi)} M_{s_i}$) based on the set of free variables of φ . In the sequel, we shall drop multiple references to the model whenever possible. Thus $\llbracket \langle\langle \varphi \rangle\rangle_{\mathcal{M}} \rrbracket_{\mathcal{M}}$ will often be simplified to $\llbracket \langle\langle \varphi \rangle\rangle \rrbracket_{\mathcal{M}}$, dropping innermost occurrences.

We can give a somewhat more compact version of this set. Notice namely that $\langle\langle \varphi \rangle\rangle_{\mathcal{M}}$ was based on a set that may properly include the set $\text{fr}(\varphi)$. For if x_i is not free but there is $j > i$ such that x_j is free in φ , then φ does not depend on x_i but nevertheless the i th component of $\langle\langle \varphi \rangle\rangle_{\mathcal{M}}$ records the values of x_i . It is thus easily seen that there are sets $A \subseteq \prod_{j < i} M_{s_j}$ and $B \subseteq \prod_{i < j < \ell(\varphi)} M_{s_j}$ such that

$$\langle\langle \varphi \rangle\rangle_{\mathcal{M}} \subseteq A \times M_{s_i} \times B. \quad (4.35)$$

There is a set $C \subseteq A \times B$ such that

$$\langle\langle \varphi \rangle\rangle_{\mathcal{M}} = \{\vec{x} \cdot y \cdot \vec{z} : \vec{x} \cdot \vec{z} \in C, y \in M_{s_i}\}. \quad (4.36)$$

By the laws of concepts,

$$\llbracket \langle \varphi \rangle \rrbracket_{\mathcal{M}} = \llbracket [C \times M_{s_i}] \rrbracket_{\mathcal{M}} = \llbracket [C] \rrbracket_{\mathcal{M}}. \quad (4.37)$$

Thus, we can actually eliminate from $\langle \varphi \rangle_{\mathcal{M}}$ all columns referring to variables that are not free in φ .

However, one should not be misled to think that it is exactly the free variables whose values need to be recorded for the formation of the concept. For sometimes variables occur free but nevertheless make no significant contribution to the formula. For example, for the formula $\chi := \varphi(\vec{y}) \wedge x_k^s = x_k^s$ we get $\text{fr}(\chi) = \text{fr}(\varphi) \cup \{x_k^s\}$. If $k \geq \ell(\varphi)$ we have

$$\langle \varphi \rangle_{\mathcal{M}} \neq \langle \chi \rangle_{\mathcal{M}}. \quad (4.38)$$

On the other hand we have

$$[\varphi]_{\mathcal{M}} = [\chi]_{\mathcal{M}}. \quad (4.39)$$

since both formulae are satisfied by the same assignments. We have $\langle \chi \rangle_{\mathcal{M}} = \langle \varphi \rangle_{\mathcal{M}}$. Thus the addition of “trivial” variables has no effect on the concept.

Let us finally turn to elementarily definable concepts. Suppose that R has the form $\langle \varphi(x_0, \dots, x_{n-1}) \rangle_{\mathcal{M}}$ for some $\varphi(x_0, \dots, x_{n-1})$. In this case R is said to be **definable**. Then

- ① $\pi[R] = \langle \varphi(x_{\pi(0)}, \dots, x_{\pi(n-1)}) \rangle_{\mathcal{M}}$.
- ② $R \times M = \langle \varphi(x_0, \dots, x_{n-1}) \wedge x_n = x_n \rangle_{\mathcal{M}}$.
- ③ $E(R) = \langle \varphi(x_0, \dots, x_{n-1}) \wedge x_{n-1} = x_n \rangle_{\mathcal{M}}$.

Hence, if one minimal member of a concept is definable, all members of the concept are definable.

Proposition 4.6 *Let c be a concept and $R, S \in c$. Then R is definable if and only if S is.*

Proof It remains to be shown that if $E(R)$ or $R \times M$ is definable, so is R . To this end, let $\langle \varphi(x_0, \dots, x_n) \rangle_{\mathcal{M}} = E(R)$. Then $\langle \exists x_n. \varphi(x_0, \dots, x_n) \rangle_{\mathcal{M}} = R$. Similarly, if $\langle \varphi(x_0, \dots, x_n) \rangle_{\mathcal{M}} = R \times M$ then $\langle \exists x_n. \varphi(x_0, \dots, x_n) \rangle_{\mathcal{M}} = R$. \square

Thus the variants of a relation can be obtained by adding some equation or existentially quantifying a relation. But there is more. Notice, for example, that the concept does not depend on the way we number the y_i . The relation will be a permutation of the original relation, which by definition is a variant of it. Additionally, let $\chi(y_1, y_0) := \varphi(y_0, y_1)$. Then $\langle \chi \rangle_{\mathcal{M}} = \langle \varphi \rangle_{\mathcal{M}}$. It is therefore the case that

$$\langle x_0^e < x_1^e \rangle_{\mathcal{M}} = \langle x_0^e > x_1^e \rangle_{\mathcal{M}}. \quad (4.40)$$

In other words, for objects of sort e the concept of “being smaller than” is the same concept as “being bigger than”. This looks like a contradiction but it is not. The idea

is that although the concept contains both relations, in the formation of complex formulae just one of them is being used at a time. This is achieved by the so-called *linking aspect*, to which we now turn.

Exercise 4.5 Show that $c \leq d$ does not hold if $\ell(c) < \ell(d)$. However, give examples where $\ell(d) > \ell(c)$ and still $c \leq d$.

Exercise 4.6 Show that $\langle\langle\varphi(x_0, x_1) \wedge x_0 = x_1\rangle\rangle \leq \langle\langle\varphi(x_0, x_1)\rangle\rangle$ need not hold.

Exercise 4.7 Show that if $R \subseteq S$ then $C_i.R \subseteq C_i.S$ and $E(R) \subseteq E(S)$.

4.4 Linking Aspects and Constructional Meanings

The previous section has introduced the concatenation of concepts, which turned out to be the greatest lower bound in the space of concepts ordered by \leq . However, when we spell this out in terms of defining formulae we get something slightly different.

Proposition 4.7 *Let φ and χ be formulae. Let s be an injective substitution such that $\text{fr}(\varphi) \cap \text{fr}(s(\chi)) = \emptyset$. Then*

$$\langle\langle\varphi\rangle\rangle * \langle\langle\chi\rangle\rangle = \langle\langle\varphi \wedge s(\chi)\rangle\rangle. \quad (4.41)$$

The proof is easy and left as an exercise. We just point out an example to show why it is generally not the case that $\langle\langle\varphi\rangle\rangle * \langle\langle\psi\rangle\rangle = \langle\langle\varphi \wedge \psi\rangle\rangle$. Let $\varphi = x_0 < x_1$ and $\psi = x_1 < x_0$. Then $\varphi \wedge \psi$ is unsatisfiable, hence $\langle\langle\varphi \wedge \psi\rangle\rangle$ is the null or falsum concept. On the other hand, the concatenation is not empty, so cannot be the null concept. According to the theorem above it is $\langle\langle x_0 < x_1 \wedge x_2 < x_3 \rangle\rangle$.

This is a welcome result. Vermeulen (1995) has made the point that the merge operation for merging DRSs should not be as proposed in Zeevat (1989), namely simply taking all variables at face value. Recall that the Zeevat-merge was defined like this, where $\langle V, \Gamma \rangle$ and $\langle W, \Delta \rangle$ are pairs of variable sets and sets of formulae:

$$\langle V, \Gamma \rangle \bullet \langle W, \Delta \rangle := \langle V \cup W, \Gamma \cup \Delta \rangle. \quad (4.42)$$

One of the problems that this faces is accidental capture.

$$\langle \{x\}, \emptyset \rangle \bullet \langle \emptyset, \{\varphi(x)\} \rangle = \langle \{x\}, \{\varphi(x)\} \rangle \quad (4.43)$$

The left-hand sides read “ $\exists x$ ” and “ $\varphi(x)$ ”, respectively and the right-hand side “ $\exists x.\varphi(x)$ ”. Such results can only be obtained by intelligent variable handling. On occasion, though, we really *do* want variables to be identified. This is the case with the phrase /a dog/, which is the concatenation of /a/ and /dog/, which translate as $\langle \{x\}, \emptyset \rangle$ and $\langle \emptyset, \{\text{dog}(x)\} \rangle$, respectively. The result we want is $\langle \{x\}, \{\text{dog}(x)\} \rangle$.

To get this effect, Vermeulen (1995) introduces *names*. Variables are optionally paired with a name, which can be anything, even an index and the variables that have the same name will be identical after merge.¹ Let $[x \mapsto 1]$ be the function mapping the variable x to 1. Then with these stipulations we get

$$\langle [x \mapsto 1], \langle \{x\}, \emptyset \rangle \rangle \bullet \langle [x \mapsto 1], \langle \emptyset, \{\mathbf{dog}(x)\} \rangle \rangle \quad (4.44)$$

$$= \langle [x \mapsto 1], \langle \{x\}, \{\mathbf{dog}(x)\} \rangle \rangle,$$

$$\langle [x \mapsto 1], \langle \{x\}, \emptyset \rangle \rangle \bullet \langle [x \mapsto 2], \langle \emptyset, \{\mathbf{dog}(x)\} \rangle \rangle \quad (4.45)$$

$$= \langle [x \mapsto 1; y \mapsto 2], \langle \{x\}, \{\mathbf{dog}(y)\} \rangle \rangle.$$

In this system the names of the variables are insignificant. Variables can be renamed inside a representation as long as distinct variables are mapped to distinct variables. Yet, the names of the variables are significant in the same way as the variable was in the Zeevat-merge. Thus we have not made much progress, because the names cannot be part of the meaning.

What we need to find is a definition of merge that does not assume that the functions are part of the representation. Instead, we must be able to define them on the basis of the concept itself. We show how to transform Vermeulen's approach. First, we simplify it by using numbers in place of names. It is clear that the names can be absolutely anything, since the only thing that matters for merge is whether names are equal or different. Now think of each number as naming a position in a tuple. Then instead of using names to associate with the variable, we associate positions in a tuple and the positions are simply numbers. Same number means then that the variable will be associated with the same position in a tuple. This leads directly to the idea of simply associating a relation with a concept. So the idea is basically this. Assume that f and g are functions from concepts to relations such that $f(c) \in c$ for every c . Then put

$$c \circlearrowleft^{f,g} d := \llbracket f(c) \cap g(d) \rrbracket. \quad (4.46)$$

This is well-defined just in case $f(c)$ and $g(d)$ are relations of the same type. Write $c \circlearrowleft^f d$ in place of $c \circlearrowleft^{f,f} d$.

Example 4.5 Transitive verbs can be coordinated to form transitive verbs. The meaning of /fry and eat/ is again a 2-concept as witnessed by /fry and eat a sausage/. Let $f = g$ both be such that they assign to the 2-concept $\langle \mathbf{fry}'(x_0, x_1) \rangle_{\mathcal{M}}$ the set $\langle \mathbf{fry}'(x_0, x_1) \rangle_{\mathcal{M}}$ and similarly to $\langle \mathbf{eat}'(x_0, x_1) \rangle_{\mathcal{M}}$ the set $\langle \mathbf{eat}'(x_0, x_1) \rangle_{\mathcal{M}}$. Then on the basis of this choice,

¹ The actual referent systems operated with a pair of such injections but we can safely ignore that complication.

$$\begin{aligned}
& \langle\langle \text{fry}'(x_0, x_1) \rangle\rangle_{\mathcal{M}} \otimes^{f,g} \langle\langle \text{eat}'(x_0, x_1) \rangle\rangle_{\mathcal{M}} \\
&= \llbracket \langle\langle \text{fry}'(x_0, x_1) \rangle\rangle_{\mathcal{M}} \cap \langle\langle \text{eat}'(x_0, x_1) \rangle\rangle_{\mathcal{M}} \rrbracket_{\mathcal{M}} \\
&= \llbracket \langle\langle \text{fry}'(x_0, x_1) \wedge \text{eat}'(x_0, x_1) \rangle\rangle_{\mathcal{M}} \rrbracket_{\mathcal{M}} \\
&= \langle\langle \text{fry}'(x_0, x_1) \wedge \text{eat}'(x_0, x_1) \rangle\rangle_{\mathcal{M}}
\end{aligned} \tag{4.47}$$

It is however also possible to coordinate concepts of different length, for example /hit and run/. Here, /hit/ denotes a 2-concept and /run/ a 1-concept. In this connection, /hit/ functions in the same way as /hit someone/. To make this work we need to select for $\langle\langle \text{run}'(x_0) \rangle\rangle_{\mathcal{M}}$ not the set $\langle\langle \text{run}'(x_0) \rangle\rangle_{\mathcal{M}}$ but the set $\langle\langle \text{run}'(x_0) \rangle\rangle_{\mathcal{M}} \times M$. Intersect this with the set $\langle\langle \text{hit}'(x_0, x_1) \rangle\rangle_{\mathcal{M}}$ and one gets the set $\langle\langle \text{hit}'(x_0, x_1) \rangle\rangle_{\mathcal{M}} \cap \langle\langle \text{run}'(x_0) \rangle\rangle_{\mathcal{M}}$ of pairs $\langle x, y \rangle$ such that x hits y and runs. This is as desired. \otimes

As concepts are defined (uniquely) by their minimal members, a special variant of this approach is to assume that f and g always pick out minimal members. Such functions are called *linking aspects*.

Definition 4.6 A **linking aspect** is a partial function Y defined on some set of concepts such that $Y(c)$ is a member of c . Y is **minimal** if $Y(c)$ is a minimal member of c for every c .

A particular way to define a linking aspect is by means of critical sets.

Definition 4.7 Let c be a concept, R a minimal member of c . A **critical set for R** is a set A such that for all minimal $Q \in c$: if $A \subseteq Q$ then $Q = R$.

Instead of mapping concepts to relations we can map them to critical sets. Let V be such a map. Then given c , $Y_V(c)$ is defined to be the unique minimal member of c containing $V(c)$.

Example 4.6 Take the concept defined by $<$ on the natural numbers. It has two minimal members: $\{\langle i, j \rangle : i < j\}$ and $\{\langle i, j \rangle : i > j\}$. The pair $\langle 0, 1 \rangle$ is in the first and not the second. Therefore $\{\langle 0, 1 \rangle\}$ is a critical set. Similarly, suppose that John is taller than Phil. Then the concept denoted by “is taller than” has two minimal relations, only one of which contains $\langle \text{John}, \text{Phil} \rangle$. Therefore, $\{\langle \text{John}, \text{Phil} \rangle\}$ is a critical set. \otimes

For an n -ary relation S let $\Pi(S)$ be the following partition of n : $C \in \Pi(S)$ iff for all $\vec{x} \in S$ and all $i, j \in C$, $x_i = x_j$. It is not hard to see that A is critical for R iff $\Pi(A) = \Pi(R)$. Now, $\Pi(\emptyset) = \{n\}$. We now define a sequence $\vec{x}_i \in R$ as follows. Put $A_i := \{\vec{x}_j : j < i\}$. If $\Pi(A_i) \neq \Pi(R)$ then let $\vec{x}_i \in R$ be chosen such that one of the sets from $\Pi(\{\vec{x}_i\})$ is not a join of partition sets from $\Pi(A_i)$. Such an element must exist if $\Pi(A_i) \neq \Pi(R)$. In that case, $\Pi(A_{i+1}) \neq \Pi(A_i)$. Since the size of the partition sets must decrease with every step it is easy to see that we can take only $n - 1$ steps; that is, we need to choose at most $n - 1$ \vec{x}_i .

Proposition 4.8 Let c be of length n . Then for every minimal $R \in c$ there is a critical set of cardinality at most $n - 1$.

This dramatically improves the bound given by Dorr (2004) of $n! - 1$. This is the best possible result. (We leave a proof of this claim to the exercises.)

Example 4.7 To see that it is not at all a weird idea to consider conjunction to be ambiguous let us look at the notion of a syntactic pivot. In English the following sentence implies that John fell:

John kissed the woman and fell. (4.48)

We say that /John/ is the **pivot** in the coordination. This is ordinarily attributed to the fact that we have a VP coordination and /John/ is the subject of both. There are languages in which the same coordination will imply that the woman fell. Such languages are invariably ergative (see Dixon (1994)); however, it is not the case that ergative languages all function in this way. Thus we need to distinguish between ergativity in case marking and ergativity in pivot choice. Similarly, some languages indicate whether or not a clause uses the same subject. Thus it explicitly marks part of the linking aspect to be used. ⊗

Example 4.8 The linking aspect is responsible for dealing with pronouns.

John saw the thief in his office. (4.49)

The pronoun /his/ may denote either John or the thief or a third person. In the present case we can paraphrase its meaning by “belonging to someone”. Thus, the phrase /in his office/ has the meaning “in the office belonging to someone”. We can interpret this someone as John, the thief or leave it unidentified. Again, for this we need different linking aspects if we insist that the only operation we want to use is conjunction. ⊗

Linking aspects give great flexibility in handling coordination. Every concept can be treated independently from the other. This might not be so desirable and leads to results that may be surprising.

Example 4.9 It is possible to define reflexivization of 2-concepts through concept conjunction. Namely, put $f(1) = \{\langle x, x \rangle : x \in M\}$ (this is not a linking aspect, in fact choosing a minimal aspect here cannot work as the minimal member of the truth concept is of length 0). Then let c be a 2-concept with minimal member R .

$$c \circledast^f 1 = \llbracket R \cap \{\langle x, x \rangle : x \in M\} \rrbracket. \quad (4.50)$$

⊗

Example 4.10 Let $M = \{a, b, c, d\}$. There are c and f such that $c \circledast^f 1 \neq (c \circledast^f 1) \circledast^f 1$. Namely, let $R = \{\langle a, a, a \rangle, \langle a, a, b \rangle, \langle a, b, a \rangle, \langle a, b, b \rangle, \langle a, a, c \rangle\}$, $c = \llbracket R \rrbracket$. Further, let $f(1) = \{\langle x, x \rangle : x \in M\} \times M$ and $f(c) = R$. Then

$$c \circledast^f 1 = \llbracket \{\langle a, a, a \rangle, \langle a, a, b \rangle, \langle a, a, c \rangle\} \rrbracket = \llbracket \{\langle a, a \rangle, \langle a, b \rangle, \langle a, c \rangle\} \rrbracket. \quad (4.51)$$

Finally, put $f(\llbracket \langle a, a \rangle, \langle a, b \rangle, \langle a, c \rangle \rrbracket) := \{ \langle a, a \rangle, \langle a, b \rangle, \langle a, c \rangle \}$ and we get

$$\begin{aligned} (c \otimes^f 1) \otimes^f 1 &= (\llbracket \langle a, a \rangle, \langle a, b \rangle, \langle a, c \rangle \rrbracket) \otimes^f 1 \\ &= \llbracket \langle a \rangle \rrbracket. \end{aligned} \quad (4.52)$$

✪

$\otimes^{Y,Z}$ is unfortunately somewhat inflexible. When we merge c and d via Y and Z , this is defined only if $Y(c)$ and $Z(d)$ have same length. Thus if $Y(c)$ has different length as $Y(d)$, then only one of $c \otimes^{Y,Z} d$ and $c \otimes^{Y,Z} e$ is defined. A better version is as follows. Let U be a function from pairs of concepts to pairs of relations such that if $U(c, d) = (R, S)$ then $R \in c$ and $S \in d$. Then put

$$c \otimes^U d := \llbracket R \cap S \rrbracket, \quad \text{where } U(c, d) = (R, S). \quad (4.53)$$

This function offers more flexibility than might be needed in natural languages but that is another matter. We conclude with a useful characterization of the logical strength of these operations.

Proposition 4.9 *Let $c = \langle \varphi(\vec{x}) \rangle$ and $d := \langle \psi(\vec{y}) \rangle$ with \vec{x} and \vec{y} disjoint. Then there is a formula χ , which is a conjunction of equations of the form $x_i = y_j$ such that $(c \otimes^U d) = \langle \varphi(\vec{x}) \wedge \psi(\vec{y}) \wedge \chi \rangle$.*

I conclude this section with a characterization of the constructional meanings. By an constructional meaning I mean such a meaning that is not provided through a lexical entry but is rather defined by the grammar. In Montague Grammar there was no need to talk about admissible meanings. If a constituent is formed, its meaning is completely determined by the meaning of its two parts. The introduction of concepts, however, has not only made it possible to use different linking aspects (and so to get different resulting meanings). The introduction of linking aspects was actually also necessitated since linking of arguments places is not unique. Additionally, the introduction of new intermediate variables has the drawback of introducing discourse objects where sometimes none should exist. Thus, we also need a mechanism to remove them. Section 4.7 will introduce a way to do this without removing them. Here we shall revert to the standard way, namely quantification. Thus we generalize the operation (4.53) once more. Let H be a set of numbers. Define for a relation R the operation $C_H.R$ as follows.

$$\begin{aligned} C_{\emptyset}.R &:= R \\ C_H.R &:= C_{H-\{i\}}.C_i.R \end{aligned} \quad (4.54)$$

In the equations above we assume that i is actually in H . (This is not strictly required but makes the definition well-founded.) The general scheme of constructional meaning is now this.

$$c \otimes^{U,H} d := \llbracket C_H.(R \cap S) \rrbracket, \quad \text{where } U(c, d) = (R, S). \quad (4.55)$$

Exercise 4.8 Prove Proposition 4.7.

Exercise 4.9 Show that the bound of Proposition 4.8 cannot be improved.

Exercise 4.10 Show that $c \otimes^Y c = c$ and that $c \otimes^{Y,Z} c = c \otimes^{Z,Y} c$. Show that $(c \otimes^{Y,Z} c) = c$ does not generally hold.

Exercise 4.11 Show that $c \otimes^Y d = d \otimes^Y c$. Give an example to show that in general $c \otimes^{Y,Z} d = d \otimes^{Y,Z} c$ is false.

4.5 Concepts and Pictures

Up to now it looked as if concepts were a complicated sort of relations. However, the intention is that in reality things are the other way around: that relations are a complicated sort of concept. In this section I would like to sketch a very different approach to concepts using *pictures*; moreover, I shall show that concepts are not at all difficult to use. The approach is just one among many and only illustrates the way things might go. We shall assume throughout that basic relations are *symmetric* so that questions of ordering between the argument places are irrelevant.

We want to define all sentence meanings as certain sets of pictures; a picture in turn is an array of coloured dots. Hence we construe pictures as functions from arrays into the set of colours. A simple approach would be to say that an array is a certain subset of, say, \mathbb{N}^2 (if the picture is planar) or \mathbb{N}^3 (for spatial pictures). However, we prefer a slightly more abstract definition. We start with a set L , the set of **locations**. A **space** is a pair $\mathcal{S} = \langle L, A \rangle$ where $A \subseteq \binom{L}{2}$ is a relation, the **adjacency relation**. Here, $\binom{L}{2}$ is the set of 2-element subsets of L . In what is to follow, relations will be identified through the two-element subsets rather than pairs. We define L^+ to be the transitive closure of L . (It follows that L^+ is symmetric and reflexive (if $\text{card } L > 1$.) We assume that any two points are related via L^+ . This means that \mathcal{S} is **connected**.

Let us assume that L is a subset of \mathbb{N}^2 and that $\{(x_0, x_1), (y_0, y_1)\} \in A$ iff $|x_1 - x_0| + |y_0 - y_1| = 1$. This means that either (a) $x_1 = x_0$ and $y_1 = y_0 \pm 1$, or (b) $y_1 = y_0$ and $x_1 = x_0 \pm 1$. Say that ℓ' is a **neighbour** of ℓ if $\{\ell', \ell\} \in A$. It follows that any $\ell \in L$ has at most 4 neighbours. We shall assume that no points have exactly zero or one neighbour; this excludes some trivial sets. From this we can define three sets of points (see Fig. 4.1):

1. **corners**: have exactly two neighbours;
2. **sides**: have exactly three neighbours;
3. **interior points**: have exactly four neighbours.

If ℓ is interior, let n_0, n_1, n_2 and n_3 be its neighbours. We say that n_1 is **across** from n_0 if there is exactly one p such that (1) $\{n_0, p\}, \{p, n_1\} \in A$ and (2) p is not a corner. There is a exactly one point that is across from n_0 ; let n_1 be across from n_0 and n_3 across from n_2 . In the space \mathcal{S} , the relation of acrossness can be used to define lines: a **line** is a maximal connected subset $G \subseteq L$ such that for any three

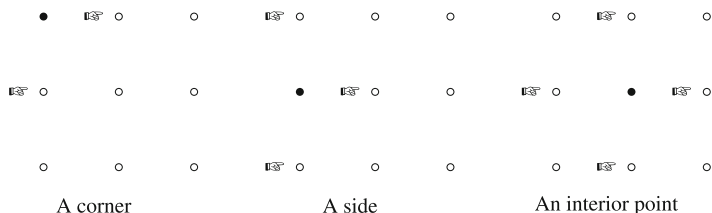


Fig. 4.1 Types of points

points p, q, r such that $\{p, q\}, \{q, r\} \in A$ p is across from r . It is easy to see that if p and q are neighbours, there is a unique line that contains them. In the plane $\mathbb{N} \times \mathbb{N}$, lines are subsets that are either horizontal or vertical. The vertical lines are parallel to each other, so are the horizontal lines. So we say that two lines G and G' are **parallel** if $G \cap G' = \emptyset$. If G and G' are not parallel, then we require that $\text{card}(G \cap G') = 1$. In the plane, if G and G' are parallel and H is not parallel to G , then it is also not parallel to G' . Now pick any line H and let $\mathcal{H} := \{H' : H \cap H' = \emptyset\}$. This defines the set of horizontal lines; pick another line V and put $\mathcal{V} := \{H : H \cap V = \emptyset\}$. This defines the set of vertical lines. Any line is either horizontal or vertical.

I stress that there is no way to say in advance which line is horizontal; the map $\bowtie: (x, y) \mapsto (y, x)$ maps L onto some different set L^{\bowtie} preserving the adjacency but interchanging horizontal and vertical lines. Furthermore, by symmetry of A , the directions “up” and “down” cannot be distinguished; likewise, the directions “left” and “right”. To fix them, we need to introduce extra structure. A **coordinate frame** is a triple $\mathcal{C} = \langle o, r, u \rangle$ in L such that $\{o, r\}, \{o, u\} \in A$ and u is not across from r . The line containing o and r defines \mathcal{H} and the line containing o and u defines the set \mathcal{V} . Now pick an interior point p . It has four neighbours, q_0 through q_3 . Which one of them is “up” from p ? First, there is exactly one line in \mathcal{V} through p and it contains, say, q_0 . It also contains one more neighbour of p , say p_1 . Then either q_0 is “up” from p or q_1 is. To settle the problem which is which we need to introduce more notions. First, the **distance** $d(x, y)$ between x and y is n if n is the smallest number such that there is a sequence $\langle x_i : i < n + 1 \rangle$ with $x_0 = x, x_n = y$ and for all $i < n$ $\{x_i, x_{i+1}\} \in A$. p is **between** q and r , in symbols $B(r, p, q)$ if p, q and r are on a line and $d(r, p), d(r, q) < d(p, q)$. Using betweenness it is finally possible to define what it is for two pairs (p, q) and (p', q') to be oriented the same way. This is left as an exercise. It follows that q_0 is up from p iff (p, q_0) is equioriented with (o, u) .

Pictures are pairs $\langle \mathcal{S}, f \rangle$, where $f : L \rightarrow C$ is a function assigning each location a colour. For simplicity we assume we have just two colours: black and white. Black represents the points of matter; white points represent nonmatter or “air”. In this case, in place of f we may just name the set of black points. This is a well known type of representations. For example, printers prints pictures by means of little dots of ink placed at certain points of a grid. Letters can be sufficiently clearly represented using a 5 by 7 grid (see Fig. 4.2). Thus we represent “matter” with a subset of the locations. A **picture** is a pair $\mathcal{P} = \langle \mathcal{S}, B \rangle$ where \mathcal{S} is a space and

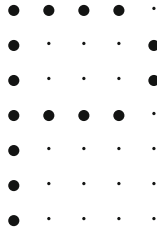


Fig. 4.2 Pictures by pixels

$B \subseteq L$. We shall continue to assume that \mathcal{S} is a rectangular subset of $\mathbb{N} \times \mathbb{N}$. An **object** in \mathcal{S} is a maximally connected subset of B . Here, $C \subseteq B$ is connected if for any two points $p, q \in C$ we have $\{p, q\} \in (A \cap \binom{B}{2})^+$. (In plain words: there is a sequence of pairwise adjacent points inside B .) $\mathcal{O}(\mathcal{S})$ is the set of objects of \mathcal{S} . Objects are therefore defined through their location. An **object schema** is a picture $\mathcal{P} = \langle\langle P, N \rangle, C \rangle$ containing a single object. We may for simplicity assume that the picture is a minimal rectangle around the object. Then we may say that \mathcal{S} **contains** an object of type \mathcal{P} if there is a function $f : P \rightarrow L$ such that (a) $\{x, y\} \in N$ iff $\{f(x), f(y)\} \in A$ and (b) $f[C]$ is an object of \mathcal{S} . The function f is also said to be a **realization** of \mathcal{P} in \mathcal{S} . The same object of \mathcal{S} can realize an object schema in different ways. This is exactly the case if it possesses internal symmetry.

Obviously, this is the most simple of all scenarios. We define an object schema as an arrangement of pixels and then declare any pixel schema that has identical arrangements (up to flipping it upside down or left-to-right) as an instantiation of that object schema. Evidently, however, we may easily complicate the matter by allowing more fancy embeddings: those that keep distance ratios intact (thus allowing to shrink or magnify the picture) or those that rotate the picture. This makes full sense only if pictures are defined over the real plane but nothing essential hinges on this, except that there is no more adjacency relation and we have to work with the topology and the metric. Let us remain with the scenario as developed so far. It is then quite easy to see how object schemata can be learnt. We need to be shown a single instance. Properties of objects (the denotations of common nouns) are inferred from their instances. It is not our concern to see how this can be done; this is the domain of cognitive science. Basically, it is done by inferring a set from some of its members (for example by constructing so-called Voronoi cells, see Gärdenfors (2004)).

The way such learning can take place in language is as follows. Let Paul be our language learner. Paul is shown a picture containing a single object, say, a football and is told that it is a ball. Thus, Paul will get the following data.

$$\langle / \text{This is a ball.} /, \boxed{\text{⊛}} \rangle \tag{4.56}$$

To the left we have an utterance, to the right a picture. That the utterance is paired with a specific picture is of some importance. Now, Paul will have to do some inference here to arrive at the fact that $/\text{ball}/$ denotes the object schema $\boxed{\text{⊛}}$ rather than the picture. Once this is achieved, however, he is able to identify the concept

denoted by /ball/. In a similar fashion he will learn other unary concepts such as “flag”, “hut”, “tent”, “telephone” and so on.

The next step from here is to learn the meaning of relational concepts. Let us take the concept “to the left of”. Unlike the denotation of common nouns, it is not identifiable by means of a single picture, since it is a relation between objects. How then can it be learned? The answer is that it is learned in basically the same way. Paul is presented with a picture and a sentence:

$$\left\langle \text{/The scissor is to the left of the ball./}, \boxed{\text{✂ } \text{⚽}} \right\rangle. \quad (4.57)$$

This picture allows to establish an association between the phrase /the scissor/ and the object to the left (since it is the only scissor) and between the phrase /the ball/ and the object to the right. This requires only knowledge of the meaning of the expressions. Similarly, Paul will encounter the following pair:

$$\left\langle \text{/The square is to the left of the heart./}, \boxed{\text{□ } \text{♥}} \right\rangle. \quad (4.58)$$

He may come to realize that the concept “left of” is independent of the shape and size of the objects involved and that it is about the location of the objects with respect to each other. In this case it can be represented just like an object schema, using a set of pictures. The burden is then entirely on the kinds of maps (“deformations”) that one is allowed to use to embed such pictures in others. It is not our main concern to do this; rather we wish to point out that the learning of the concept “left of” is no more complex using concepts than it is using relations.

How then is “right of” learnt? Basically the same way. It could be using the following data.

$$\left\langle \text{/The ball is to the right of the scissor./}, \boxed{\text{⚽ } \text{✂}} \right\rangle \quad (4.59)$$

Here we can appreciate for the first time that concepts really *are* simpler. The picture shown is namely absolutely the same. However, in conventional representations we would write (4.59) as

$$\text{right}'(\iota x.\text{ball}'(x), \iota x.\text{scissor}'(x)). \quad (4.60)$$

By contrast, the meaning of (4.57) would be rendered as

$$\text{left}'(\iota x.\text{scissor}'(x), \iota x.\text{ball}'(x)). \quad (4.61)$$

The two formulae are not the same. The positional regime in the formulae forbids us from treating them the same. To get an identical encoding we need to translate “right” systematically as “left” and invert the linking. This is what the concepts do anyway. Paul will learn that whatever is to the left of the occurrence of /right/ will be on the right in the picture of what is to the right of the occurrence of /right/.

I should point out that it is certainly not necessary that the meaning of (4.57) is exactly the same as (4.59). In this case /right/ denotes a different concept than /left/. We shall worry no further about that possibility. It should however be said that there can be concomitant differences in the choice between (4.57) and (4.59) stemming from different sources. I mention here that constructions of the form “X is in location Y” generally indicate that Y is more stable, less movable, or bigger than X (see Talmy (2000)).

The bicycle is to the left of the house. (4.62)

?The house is to the right of the bicycle. (4.63)

Again, this issue seems to be orthogonal to the one at hand. (Notice also that (4.63) is not *false*, just inappropriate.)

We shall now test Paul’s knowledge of English. We give him the picture (4.64)

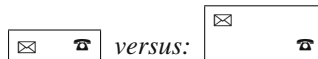


and ask him:

Is the letter to the left of the phone? (4.65)

Paul will perform the following steps:

- ① Compare the two arguments of /left/ in (4.65) in (4.57). The comparison on the basis of form alone yields that /the scissor/ must be associated with /the letter/ and /the ball/ with /the phone/.
- ② Take the picture of (4.57) and do the following replacement: replace the scissor by the letter and the ball by the phone.
- ③ Compare the resulting picture with the one given:



- ④ If there is an admissible deformation to take us from left to right for the concept “left” then the answer is “yes”.

Thus, the entire burden is still in learning the correct meaning of the geometry of “left”. Learning the associations with syntactic arguments is very easy by comparison. Moreover, a semantics based on relations offers no advantage.

We have deliberately picked the concept “left”. Unlike concepts denoted by verbs, geometric notions do not allow to pick out one of the arguments by means of intrinsic properties. For example, the sentence “John is pushing the cart.” is true because it is John who exerts force on the cart and not conversely. Likewise, it is known that directionals modify the mover in an event and no other constituent. Thus “John threw the banana out of the window.” means that John threw the banana and *it* went out of the window. If John decides to jump out of the window while

tossing the banana onto the kitchen table, that does not make the sentence true. The mechanism for learning such concepts is essentially the same. However, while the linking in relational nouns and adjectives has to be learned on a case by case basis, the linking on verbs sometimes allows to make big abstractions. This just means that the knowledge of how linking is to be effected becomes more abstract.

Let us finally turn to another complication, namely passive, or relation change in general.

John throws the banana out of the window. (4.66)

The banana is thrown out of the window. (4.67)

It is obvious that to learn English correctly consists in realizing that there are different verb forms, namely active and passive and that what they signal is that the linking has to be different in the two cases. From this point on there are two choices: Paul might start acquiring *two* different linkings for the verbs, one active and one passive; or Paul might develop a recipe of deriving the linking in the passive sentences from the linking in active sentences. How he goes about this is to a large degree a question of how the languages are structured (in other words: how systematic the active passive change really is).

I close this section with a few remarks about what we have done. We have described sentences as properties of pictures. There was therefore only one entity in semantics: that of a picture. To describe how it is that we arrive at the interpretation of a sentence, however, we complicated the ontology. If a sentence has subjects, something must correspond to them. Thus we introduced individuals, concepts and so on into the semantics. However, ontologically these were considered derived objects. We constructed a function that will derive from a picture \mathcal{P} the set of its objects $\mathcal{O}(\mathcal{P})$. The next object we introduced are the object schemes; an object scheme P is a picture \mathcal{Q} together with a family F of admissible embeddings. An object $o \in \mathcal{O}(\mathcal{P})$ has a property P if there is an admissible embedding $f : \mathcal{Q} \rightarrow \mathcal{P}$ such that the image of the black points is exactly o .

Exercise 4.12 Define the relation of “having same orientation” using betweenness in a plane. *Hint.* Start by defining it for pairs of points on the same line. Then show it can be projected to other, parallel lines.

4.6 Ambiguity and Identity

We have shown earlier that sentences are ambiguous and this is either because the words have several meanings or because a given exponent has several derivations. In view of ambiguity we must reassess our notion of what it is that makes a sentence true. Under the standard definitions in logic we declare a sentence true if it denotes the value 1 or the true concept, whichever. However, if a sentence is ambiguous this creates a difficulty. Consider the word /crane/. It has two meanings: it denotes a kind of bird and a kind of machine. This means that the lexicon contains two signs,

where crane_1 is the concept of bird cranes and crane_2 is the concept of machine cranes.

$$\text{BCR} := \langle \text{crane}, \text{crane}_1 \rangle \quad (4.68)$$

$$\text{MCR} := \langle \text{crane}, \text{crane}_2 \rangle \quad (4.69)$$

Consider now the following sentence.

$$\text{Cranes weigh several tons.} \quad (4.70)$$

This sentence has two derivations. Unless partiality strikes, in a structure term containing BCR we can replace BCR by MCR and the new term unfolds to a sign with the same exponent (but different meaning).

(4.70) is false if we interpret /cranes/ as talking about birds (that is, if we take the structure term to contain BCR rather than MCS) but true in the other understanding of the word. It is the other way around with

$$\text{Cranes can fly.} \quad (4.71)$$

This creates a tension between the notion of “true given an understanding” and “true simpliciter”. We shall propose (not uncontroversially) that a sentence is true simpliciter if it has a structure term under which it is true. This is a matter of convention but for the case at hand not far off the mark. It then is the case that both (4.70) and (4.71) are true.

Now what about negated sentences? Here we must distinguish between two kinds of negations. There is an inner negation and an outer negation. The inner negation produces a negated sentence, while the outer negation denies the truth of the sentence. Let us look at negation formed by /it is not the case that/.

$$\text{It is not the case that cranes weigh several tons.} \quad (4.72)$$

If taken as outer negation, this sentence is false (because (4.70) is true). If taken as inner negation, it is true. To see this, let us imagine that we do not have the word /cranes/ but in fact two: /cranes₁/, denoting the birds and /cranes₂/, denoting a kind of machines. Then (4.70) is true if either of the following sentences is true:

$$\text{Cranes}_1 \text{ weigh several tons.} \quad (4.73)$$

$$\text{Cranes}_2 \text{ weigh several tons.} \quad (4.74)$$

(4.70) is false if *both* (4.73) and (4.74) are false. It is possible though to negate both of them individually:

$$\text{It is not the case that cranes}_1 \text{ weigh several tons.} \quad (4.75)$$

$$\text{It is not the case that cranes}_2 \text{ weigh several tons.} \quad (4.76)$$

The first is true while the second is false. In English, where the two concepts are denoted by the same word, (4.75) and (4.76) are both expressed by (4.72). Since (4.76) is true, so is therefore (4.72).

I should say, however, that the notion of outer negation cannot be implemented in the present system without major changes. For if outer negation is a sign in its own right, its meaning is a quantifier over structure terms. However there is no way to get such a quantifier. It is not clear to me whether or not outer negation can be expressed in embedded sentences. If it cannot be expressed, the present theory can obviously be adapted rather straightforwardly; but if it can be expressed, then the adaptations are indeed major. They would require namely a grammar that uses the language transform L^{\S} of L rather than L itself (see page 95 for a discussion of L^{\S}).

The previous discussion can be used to shed light on identity statements as well. Consider the sentence

$$\text{The morning star is the evening star.} \quad (4.77)$$

This is true if and only if the star that is the morning star is the same star as the evening star. It happens to be the case that they actually *are* the same. If John however is unaware of this, then he believes that (4.77) is false and that (4.78) is true.

$$\text{The morning star is not the evening star.} \quad (4.78)$$

This problem has been extensively dealt with in philosophy. I shall not go into that discussion. Rather, I shall discuss how our definitions change the way in which this puzzle must be discussed.

Example 4.11 Let $M = \{x\}$. Furthermore, we shall assume that our language has the following basic signs.

$$\begin{aligned} \mathcal{I}(f_0) &:= \langle \text{the morning star}, \{x\} \rangle \\ \mathcal{I}(f_1) &:= \langle \text{the evening star}, \{x\} \rangle \end{aligned} \quad (4.79)$$

And let it have one mode:

$$\mathcal{I}(f_2)(\langle e_0, m_0 \rangle, \langle e_1, m_1 \rangle) := \langle e_0 \widehat{\perp} \text{is} \widehat{\perp} e_1, m_0 \otimes m_1 \rangle. \quad (4.80)$$

Here, \otimes is defined as intersection of two 1-concepts by intersecting their minimal members. Let L_1 be the language defined by all definite terms. It is

$$\begin{aligned} L_1 := & \{ \langle \text{the morning star}, (\{x\})_M \rangle, \langle \text{the evening star}, (\{x\})_M \rangle, \\ & \langle \text{the morning star is the morning star}, 1 \rangle, \\ & \langle \text{the morning star is the evening star}, 1 \rangle, \\ & \langle \text{the evening star is the morning star}, 1 \rangle, \\ & \langle \text{the evening star is the evening star}, 1 \rangle \} \end{aligned} \quad (4.81)$$

Now let $N = \{v, w\}$. We assume the same signature but instead the following interpretation:

$$\begin{aligned} \mathcal{K}(f_0) &:= \langle \text{the morning star}, \{v\} \rangle \\ \mathcal{K}(f_1) &:= \langle \text{the evening star}, \{w\} \rangle \\ \mathcal{I}(f_2)(\langle e_0, m_0 \rangle, \langle e_1, m_1 \rangle) &:= \langle e_0 \hat{\cup} \text{is} \hat{\cup} e_1, m_0 \odot m_1 \rangle \end{aligned} \quad (4.82)$$

Let L_2 be the language defined by this interpretation. Then

$$\begin{aligned} L_2 := & \{ \langle \text{the morning star}, (\{v\})_M \rangle, \langle \text{the evening star}, (\{w\})_M \rangle, \\ & \langle \text{the morning star is the morning star}, 1 \rangle, \\ & \langle \text{the morning star is the evening star}, 0 \rangle, \\ & \langle \text{the evening star is the morning star}, 0 \rangle, \\ & \langle \text{the evening star is the evening star}, 1 \rangle \} \end{aligned} \quad (4.83)$$

We have the following result: there are *two* languages, not one, whose corresponding string language is the same and we even have two string identical grammars. Nevertheless, qua interpreted languages, L_1 and L_2 are different. \otimes

The example has the following moral. Two languages cannot be the same if the models are not the same. Thus, to say that John and Paul speak the same language—in the sense of interpreted language, which we take to be the default—requires that their interpretations are the same. If Paul is convinced that the morning star is the evening star and John thinks they are different then Paul and John do not speak the same language. In order for them to speak the same language we require that not only the expressions are the same, we also require that the expressions have the same meaning. And “same” must be taken in a strict sense: both John and Paul would be required to take the expressions /the morning star/ to denote the same thing and likewise /the evening star/. But they do not. There are in fact two reasons why two people can fail to share the same language. One is as just described: they disagree on the truth value of some sentences. Another more subtle case is described in the next example.

Example 4.12 L_3 is like L_1 except that y takes the place of x . Thus, for example,

$$\begin{aligned} L_3 := & \{ \langle \text{the morning star}, (\{y\})_M \rangle, \langle \text{the evening star}, (\{y\})_M \rangle, \\ & \langle \text{the morning star is the morning star}, 1 \rangle, \\ & \langle \text{the morning star is the evening star}, 1 \rangle, \\ & \langle \text{the evening star is the morning star}, 1 \rangle, \\ & \langle \text{the evening star is the evening star}, 1 \rangle \} \end{aligned} \quad (4.84)$$

Now let $P = \{y\}$. We assume the same signature but instead the following interpretation:

$$\begin{aligned}\mathcal{L}(f_0) &:= \langle \text{the morning star}, \{y\} \rangle \\ \mathcal{L}(f_1) &:= \langle \text{the evening star}, \{y\} \rangle \\ \mathcal{L}(f_2)((e_0, m_0), (e_1, m_1)) &:= \langle e_0 \hat{\sqcup} \text{is} \sqcup \hat{\sqcap} e_1, m_0 \otimes m_1 \rangle\end{aligned}\tag{4.85}$$

The grammars $\langle \Omega, \mathcal{I} \rangle$ and $\langle \Omega, \mathcal{L} \rangle$ are naturally equivalent. \odot

The languages L_1 and L_3 are different, yet in an abstract sense identical. Now picture the case where George speaks L_3 . We would like to say that George and Paul speak the same language but we cannot. In fact, this is as it should be. Notice that we must distinguish (for natural language) two notions of language. There is a private language, where expressions are interpreted as objects or constructs in a speaker; and a public language where expressions are interpreted with real objects (if applicable). We think for example that the public meaning of /the morning star/ is Venus, as is the public meaning of /the evening star/. The private language of an individual speaker needs to be “connected” to the public language in the correct way. This is similar in the distinction between phonemes and sounds. While two speakers can share the same phonemic system it may turn out that the two systems are differently realized in terms of sounds. And likewise it may happen that while Paul thinks that the morning star is the evening star and both happen to be Venus, it may also happen that George thinks that the morning star and the evening star are Mars. The private languages of Paul and George are different for the trivial reason that the internal objects of both Paul and George must be different; but we can easily establish a correspondence between them, an isomorphism, that makes them the same. And so the private languages of Paul and George are the same up to isomorphism, yet their public languages are different. The puzzle is thus resolved by appeal to different *de lingua* beliefs, to use a phrase of Fiengo and May (2006). The idea of Fiengo and May (2006) is roughly that what is behind many puzzles of identity is that speakers hold different beliefs concerning the referents of expressions. In the theory proposed here, this is cashed out as follows. The **abstract language** is a language where meanings are identifiable up to equivalence (as established in Section 3.7). Any two speakers can speak the same abstract language, so the abstract language is not the private language. Neither is it the public language. For that, we also need to **ground** a language by providing translations into real world objects. Abstract language behaviour can be established using logical connections between sentences, while concrete language behaviour can be established by asking people about meanings in terms of observable facts.² This is just a sketch of a solution but it serves as a formal explication of Fiengo and May (2006) who actually think that

² This is evidently a simplified scenario. The visible facts may not be the same across speakers, thus accounting for a different layer of confusion. But it is important to note that the distinction between what is abstract in a language and what is not is real. In a sense, the fact that Tully is Cicero is not part of the abstract language.

many sentences also express what they call a *de lingua belief*. A *de lingua belief* is a belief about what expressions denote. If the present theory is correct, it is a belief about the public language.

One puzzle that Fiengo and May discuss at length is the Paderewski puzzle by Kripke. It goes roughly as follows. Max goes to a concert by a musician named Paderewski and comes to believe that he is a great musician. Later he visits a political rally by a person named Paderewski. He comes to think that the latter person is actually a bad musician. So he holds two beliefs.

Paderewski is a great musician. (4.86)

Paderewski is a bad musician. (4.87)

It so turns out that the two people are one and the same. The philosophical problems arise from the fact that under certain views of reference Max holds inconsistent beliefs. Both Fine (2007) and Fiengo and May (2006) discuss this problem. Again we need not go into the philosophical detail here. What interests us is what may linguistically be said to be going on. The idea is that for Pavel, who knows (or believes) that both persons are the same, /Paderewski/ is unambiguous. For Max it is not. So, the language of Max has two signs, say, ⟨Paderewski, {x}⟩ and ⟨Paderewski, {y}⟩, while the language of Pavel only has one such sign, say ⟨Paderewski, {v}⟩. Thus, for Max the expression /Paderewski/ is ambiguous, for Pavel it is not. Given our notion of truth for ambiguous sentence, it is correct for Max to hold both (4.86) and (4.87) true. There is no logical problem, since the sentence is simply ambiguous. This contrasts with the idea of Fiengo and May (2006) who think that names are not expressions. They can only occur in the form [₁Paderewski], where the brackets are used to keep track of different objects. In the theory proposed here there is no sense in disambiguation on the syntactic level. This must be done in the semantics. Consequently, also the two occurrences of the name in the sentence

Paderewski is Paderewski. (4.88)

cannot simply be told apart by indexation so that one can distinguish between, for example,

Paderewski₁ is Paderewski₁. (4.89)

and

Paderewski₁ is Paderewski₂. (4.90)

The reason, again, is that there is no indication at the surface. Instead, in order to be clear, Max must use some expression that makes the referent unique. Notice that Max also agrees to the (inner!) negation of (4.88):

Paderewski is not Paderewski. (4.91)

The difference between this approach and Fiengo and May (2006) is brought out also by the way in which Pavel can make Max aware that he is wrong about Paderewski. For it is not enough for him to point out (4.91), for that is what is also true for Max. Rather he must use a sentence that would not be true for Max, for example

There is only one Paderewski. (4.92)

The problem is that Pavel cannot make himself understood to Max by using the name simpliciter. He must in order to discriminate his beliefs from Max's beliefs use sentences that come out differently. What Fiengo and May (2006) have in mind is that Pavel can also use a certain version of (4.88), for example

But Max, Paderewski *IS* Paderewski. (4.93)

But again, how is Max to interpret this if he cannot see which of the Paderewskis is pointed to on each of the occasions?

Exercise 4.13 In Example 4.11 the word /is/ is syncategorematic. Show that this syncategorematic use can be eliminated from the grammar.

4.7 Profiling

As I have indicated at many places there is a difference between what is commonly referred to as model theoretic semantics and the more popular representational semantics. It has not always been openly admitted by semanticists that the representations involved in many brands of formal semantics do not use meanings in the sense of truth conditions but rather are just pieces of notation. Such is the case with DRT, Minimal Recursion Semantics, semantics used in connection with TAGs, underspecification semantics, continuations and so on. If meanings only contain truth conditions, then all these semantics could not ever claim to implement a compositional approach to meaning. However, such argumentation misses a point. For one line of defence is still open and should be considered: that it is not the only objective of semantics to account for truth conditional meanings but *also* to account for internal meanings. Thus I believe that the justification for using such representations cannot be found in the truth conditions that they formulate. Rather, it must be in the fact that these objects are essentially what humans use. Whether that is so and which one it is that we use is an empirical question and will have to be left to empirical research. However, I shall just add a few remarks about the necessity of considering internal meanings. If we take, for example, the notion of a dog to be the set of all dogs, then that object is not the kind of object we can have in our head. We may say instead that the meaning is a particular algorithm (for recognising dogs); but even that has a similar consequence. The algorithm turns out to be abstract, too. The particular procedure that one person uses to differentiate dogs from other

animals might be different from that of some other person in certain insignificant ways. We will then still say that the two people have the same algorithm, though its implementations, that is, the concrete procedures, are different.

The crucial fact about the concreteness of meanings is that to understand whether or not two concrete meanings m and m' instantiate the same abstract meaning must be decided by explicit manipulation of the representations. This is the same in logic, where we distinguish between two formulae representing the same truth condition. Since truth conditions are too big to be stored directly we rely instead on a calculus that manipulates representations up to truth conditional equivalence. This picture undermines much of what I have said so far about semantics since it moves us away from a static notion of meaning and towards a dynamic semantics based on reasoning whose objects are symbolic in nature. I shall not continue this line since it is too early to tell how such an account may go.

It so turns out, however, that human languages are still different. There are certain things that have been argued to exist in internal representations for which there is no obvious external correlate. One such thing is profiling. Profiling is the way in which objects in an array are distinguished from each other, by making one more prominent than the others. We can explain the difference between “left” and “right”, for example, in terms of profiling. While they both denote the same concept, the profile of “left” is inverse of that of “right”. How can this be understood? In the pictures we can simply add a pointer to the profiled entity. If we denote concepts by formulae then we can use underlining to do the same: thus, $\langle \text{left}(x, y) \rangle$ and $\langle \text{left}(\underline{x}, y) \rangle$ are concepts in which different elements are profiled. If we use concepts, we reserve, say, the first column for the profiled element and restrict permutation in such a way that it does not permute the first column with any other. There is a temptation to think of profiling as just another instance of a sort. But we have to strictly distinguish the two. The two objects involved in the relation “left” (and “right”) are not sortally distinct. Moreover, one and the same object can at the same time be to the left of an object and to the right of another. This cannot happen if a different profile means a different sort. However, from the standpoint of combining meanings profiling has the same effect, namely to enhance the possibilities of combining two concepts.

In the first part of this section I shall outline a formalism for such meanings. In the second half I show how this gets used in practice.

Let S be a set of sorts. So far we have construed concepts as sets of relations. The minimal members of a relation had to be of similar type. Now we think of the relations of a concept to be divided into subparts, each corresponding to a particular profile. We allow individual sorts to be profiled independently.

Definition 4.8 Let P be a set of profiles and M a set. A P -profiled relation over M is a pair $\mathcal{R} = \langle \vec{p}, R \rangle$ where R is a relation and $\vec{p} \in P^*$ of length identical to the length of R .

The relation R contains vectors $\langle x_0, x_1, \dots, x_{n-1} \rangle$. When paired with the sequence $\langle p_0, p_1, \dots, p_{n-1} \rangle$ this means that x_i will have the profile p_i . Since the profile is paired with the entire relation, the profile p_i is also given to y_i in

$\langle y_0, y_1, \dots, y_{n-1} \rangle \in R$ in \mathcal{R} . One may or may not want to impose requirements on the profiling. For example, suppose there is a label saying that the element is in focus; this label we do not want to be distributed to more than one column. I will not pursue this here.

A profiled concept is a set of profiled relations. The details are similar to those of Section 4.3. Moreover, I add here that referent systems of Vermeulen (1995) can be seen as profiled concepts. The profiled concept generated by \mathcal{R} , also written $\llbracket \mathcal{R} \rrbracket_{\mathcal{M}}$, is the least set closed both ways under the following operations.

- ① $\pi[\langle \vec{p}, R \rangle] := \langle \pi(\vec{p}), \pi[R], \pi$ a permutation of the set $|\vec{p}| = \{0, 1, \dots, |\vec{p}|-1\}$;
- ② $E_{s,q}(\langle \vec{p}, R \rangle) := \langle \vec{p} \cdot q, R \times M_s \rangle$;
- ③ $D_i(\langle \vec{p}, R \rangle) := \langle \vec{p} \cdot p_i, \{\vec{x} \cdot x_i : \vec{x} \in R\} \rangle$.

Notice that when duplicating a column we must also duplicate the corresponding profile. It is therefore quite possible to have two identical columns, as long as they have different profiles. Notice that full columns are discarded regardless of their profile.

The **deprofiling** of $\langle \vec{p}, R \rangle, \delta(\langle \vec{p}, R \rangle)$, is simply R . Similarly, we define the deprofiling of a profiled concept.

$$\delta(\llbracket \mathcal{R} \rrbracket) := \{S : \text{there is } \vec{q} : \langle \vec{q}, R \rangle \in \llbracket \mathcal{R} \rrbracket\} \quad (4.94)$$

So, $\delta(\mathcal{C}) = \delta[\mathcal{C}]$. The following gives the justification for this definition. Its proof is left as an exercise.

Proposition 4.10 $\delta(\llbracket \mathcal{R} \rrbracket)$ is a concept.

There is a converse operation of introducing a profiling. While we could do that on a concept-by-concept basis, there are more interesting methods.

Definition 4.9 Let Y be a linking aspect and $f : \mathbb{N} \rightarrow P$ a function. Then define the profiled concept $f^Y(c)$ as follows.

$$f^Y(c) := \llbracket \langle f \upharpoonright \text{card}(Y(c)), Y(c) \rangle \rrbracket \quad (4.95)$$

In this definition, assume that $\text{card}(Y(c)) = n$. Then $f \upharpoonright \text{card}(Y(c))$ is the restriction of f to $n = \{0, \dots, n-1\}$. This is then viewed as the sequence $\langle f(0), f(1), \dots, f(n-1) \rangle$. The idea is that all we need to specify is the way in which the positions are profiled; the rest is done by the linking aspect, which lines up the columns of the relation in a particular way.

The crucial difference between profiled concepts and ordinary concepts is that we can use the profiles to define the linking; and that we can also change the profile if necessary (unlike the typing). In principle, since the profiling is arbitrary, we consider two profiled concepts as basically identical if they have the same profile.

Definition 4.10 Two profiled concepts \mathcal{C} and \mathcal{D} are said to be **homologous** if $\delta(\mathcal{C}) = \delta(\mathcal{D})$.

Any change from a profiled concept to a homologous profiled concept is thus considered legitimate. There are various methods to define such a change for the entire space of concepts. Here is one.

Definition 4.11 Let S be a set of sorts and P a set of profiles. A **reprofiling** is a family $\{\rho_s : s \in S\}$ of maps $\rho_s : P \rightarrow P$. The reprofiling of a profiled relation $\mathcal{R} = \langle \vec{p}, R \rangle$ of type \vec{s} is the relation $\rho(\mathcal{R}) := \langle \rho_R\{\vec{p}\}, R \rangle$ which is defined as follows.

$$\begin{aligned}\rho_R\{p_i\} &:= \rho_{s_i}(p_i) \\ \rho_R\{\vec{p}\} &:= \langle \rho_R\{p_i\} : i < |\vec{p}|\rangle\end{aligned}\tag{4.96}$$

Notice that the type of the relation is recoverable from the relation itself (in contrast to its profile). So the reprofiling assigns to elements of type s and profile p the new profile $\rho_s(p)$, whereas the type remains the same.

Proposition 4.11 Let C be a profiled concept and $\rho = \{\rho_s : s \in S\}$ a reprofiling. Then $\rho[C]$ is a profiled concept.

Again the proof is straightforward.

The simplification introduced by profiling is considerable. Suppose for example we want to conjoin two concepts. Then we can only do this if we have a linking aspect. However, linking aspects are in general not finitely specifiable. Thus, unlike syntactic rules, the semantic combination rules based on concepts are arbitrarily complex. In [Section 5.3](#) I shall give an example of a grammar for a fragment of English that essentially uses linking aspects only for the basic entries of the lexicon. If one wants to treat language in its full complexity one will be forced to do either of two things: make the linking aspect dynamic, that is, to be computed on the side; or introduce profiling. In this section I shall explore the second option.

Now that we have profiled concepts we may actually take advantage of the profiling in defining combinations of concepts. Our example here concerns the definition of linking aspects.

Example 4.13 Arbitrarily embedded relative clauses.

$$\begin{aligned}\text{a dog that saw a cat that chased a mouse that ate} & \quad (4.97) \\ \text{a cheese} & \end{aligned}$$

Let $D = \{d_i, c_i, m_i, h_i : i \in \mathbb{N}\}$ be the domain. There is only one sort. Let us define three binary relations:

$$\begin{aligned}E &:= \{(m_0, h_0)\} \cup \{(d_i, h_{i+1}) : i \in \mathbb{N}\} \\ C &:= \{(c_i, m_i) : i \in \mathbb{N}\} \cup \{(c_i, d_{i+1}) : i \in \mathbb{N}\} \\ S &:= \{(d_i, c_i) : i \in \mathbb{N}\} \cup \{(m_i, d_{2i}) : i \in \mathbb{N}\}\end{aligned}\tag{4.98}$$

$$\begin{aligned}
\mathcal{I}(g_0)() &:= \langle \mathbf{a}, \langle \top \rangle \rangle \\
\mathcal{I}(g_1)() &:= \langle \mathbf{that}, \langle \top \rangle \rangle \\
\mathcal{I}(f_0)() &:= \langle \mathbf{dog}, \llbracket \{d_i : i \in \mathbb{N}\} \rrbracket \rangle \\
\mathcal{I}(f_1)() &:= \langle \mathbf{cat}, \llbracket \{c_i : i \in \mathbb{N}\} \rrbracket \rangle \\
\mathcal{I}(f_2)() &:= \langle \mathbf{mouse}, \llbracket \{m_i : i \in \mathbb{N}\} \rrbracket \rangle \\
\mathcal{I}(f_3)() &:= \langle \mathbf{cheese}, \llbracket \{h_i : i \in \mathbb{N}\} \rrbracket \rangle \\
\mathcal{I}(f_4)() &:= \langle \mathbf{saw}, \llbracket S \rrbracket \rangle \\
\mathcal{I}(f_5)() &:= \langle \mathbf{chased}, \llbracket C \rrbracket \rangle \\
\mathcal{I}(f_6)() &:= \langle \mathbf{ate}, \llbracket E \rrbracket \rangle
\end{aligned} \tag{4.99}$$

There will be one mode of composition, which is binary. Let Y be the following linking aspect. For every unary concept it picks the unique minimal member and is defined on three binary concepts only, where $Y(c)$ is that relation which contains $V(c)$, where V assigns the following critical sets to the concepts:

$$\begin{aligned}
\llbracket E \rrbracket &\mapsto \{\langle m_0, h_0 \rangle\} \\
\llbracket C \rrbracket &\mapsto \{\langle c_0, m_0 \rangle\} \\
\llbracket S \rrbracket &\mapsto \{\langle d_0, c_0 \rangle\}
\end{aligned} \tag{4.100}$$

(Recall $V(c)$ is a set such that exactly one minimal member of c contains $V(c)$. $Y(c)$ is defined to be that set.)

Now, $\gamma(e, e')$ is defined if and only if either of the following holds:

- ① $e = /a/$ and e' begins with $/cheese/, /mouse/, /dog/,$ or $/cat/$.
- ② $e \in \{/ate/, /saw/, /chased/\}$ and e' starts with $/a_{\perp}/$.
- ③ $e = /that/$ and e' starts with $/chased_{\perp}/, /saw_{\perp}/$ or $/ate_{\perp}/$.
- ④ $e \in \{/cat/, /mouse/, /dog/, /cheese/\}$ and e' starts with $/that_{\perp}/$.

$$\mathcal{I}(m)(\langle e, m \rangle, \langle e', m' \rangle) := \begin{cases} \langle e \frown_{\perp} e', m \otimes^Y m' \rangle & \text{if } \gamma(e, e'), \\ \text{undefined} & \text{else.} \end{cases} \tag{4.101}$$

So, the syntax is right regular. Without specifying too much detail let me note the first steps in the derivation.

$$\begin{aligned}
&\langle \mathbf{cheese}, \llbracket \{h_i : i \in \mathbb{N}\} \rrbracket \rangle \\
&\langle \mathbf{a cheese}, \llbracket \{h_i : i \in \mathbb{N}\} \rrbracket \rangle \\
&\langle \mathbf{ate a cheese}, \llbracket \{\langle d_i, h_{i+1} \rangle : i \in \mathbb{N}\} \cup \{\langle m_0, h_0 \rangle\} \rrbracket \rangle \\
&\langle \mathbf{that ate a cheese}, \llbracket \{\langle d_i, h_{i+1} \rangle : i \in \mathbb{N}\} \cup \{\langle m_0, h_0 \rangle\} \rrbracket \rangle \\
&\langle \mathbf{mouse that ate a cheese}, \llbracket \{\langle m_0, h_0 \rangle\} \rrbracket \rangle \\
&\langle \mathbf{a mouse that ate a cheese}, \llbracket \{\langle m_0, h_0 \rangle\} \rrbracket \rangle
\end{aligned} \tag{4.102}$$

At this point we get stuck; for we must now be able to combine two binary concepts. If we combine them the wrong way, instead of interpreting /a cat that chased a mouse that ate a cheese/ we interpret /a cat that chased a cheese that ate a mouse/. As the embedding depth of relative clauses is unbounded there is no recipe for defining the linking aspect using critical sets as long as they do not exhaust the entire relation. So, we have to use a linking aspect instead. \otimes

Example 4.14 We come to the first repair strategy. Leave everything as is with one exception. In the interpretation of m , quantify away the lower elements, always retaining a 1-concept. M is the domain of the model.

$$\begin{aligned} \mathcal{I}(m)(\langle e, m \rangle, \langle e', m' \rangle) & \quad (4.103) \\ := \begin{cases} \langle e \hat{\sqsubset} \hat{\sqsubset} e', \llbracket \mathbf{C}_1.(Y(m) \cap (M \times Y(m'))) \rrbracket \rangle & \text{if } \gamma(e, e') \text{ and } m \text{ is binary,} \\ \langle e \hat{\sqsubset} \hat{\sqsubset} e', m \otimes^Y m' \rangle & \text{if } \gamma(e, e') \text{ and } m \text{ is unary,} \\ \text{undefined} & \text{else.} \end{cases} \end{aligned}$$

The derivation now goes as follows.

$$\begin{aligned} & \langle \text{cheese}, \llbracket \{h_i : i \in \mathbb{N}\} \rrbracket \rangle \\ & \langle \text{a cheese}, \llbracket \{h_i : i \in \mathbb{N}\} \rrbracket \rangle \\ & \langle \text{ate a cheese}, \llbracket \{d_i : i \in \mathbb{N}\} \cup \{m_0\} \rrbracket \rangle \\ & \langle \text{that ate a cheese}, \llbracket \{d_i : i \in \mathbb{N}\} \cup \{m_0\} \rrbracket \rangle \\ & \langle \text{mouse that ate a cheese}, \llbracket \{m_0\} \rrbracket \rangle \\ & \langle \text{a mouse that ate a cheese}, \llbracket \{m_0\} \rrbracket \rangle \end{aligned} \quad (4.104)$$

The step from the second to the third line is the crucial bit. We invoke the linking aspect on both concepts. The right-hand side is unary, so we get the unique minimal member. The left-hand side is the concept associated with one of the verbs and by using the critical sets we align them such that the first column is the subject and the second is the object. We identify the object with the unary relation and quantify it away.

Thus, when we have processed one embedding we are back to a unary concept and can continue:

$$\begin{aligned} & \langle \text{chased a mouse that ate a cheese}, \llbracket \{c_0\} \rrbracket \rangle \\ & \langle \text{that chased a mouse that ate a cheese}, \llbracket \{c_0\} \rrbracket \rangle \\ & \langle \text{cat that chased a mouse that ate a cheese}, \llbracket \{c_0\} \rrbracket \rangle \\ & \langle \text{a cat that chased a mouse that ate a cheese}, \llbracket \{c_0\} \rrbracket \rangle \end{aligned} \quad (4.105)$$

The problem with this approach is that the intermediate objects are gone and cannot be referred to any more (say, with /the mouse that ate a cheese/). \otimes

Example 4.15 The second strategy uses profiling. Let $P := \{t, b\}$. The rule of combination is this. We assume that the subjects of verbs are assigned the profile t ; all other arguments are assigned b . When a verb is combined with an object, the object position is identified with the object with profile t , upon which the profile of this element is set to b . On the assumption that only one column has label t , we define the following linking algorithm. Assume that $\langle t \cdot \vec{b}_1, R \rangle \in \mathcal{C}$ is of length m and $\langle x \cdot \vec{b}_2, S \rangle \in \mathcal{D}$ of length n . Then we put

$$R\overline{\otimes}S := \{x \cdot \vec{y} \cdot \vec{z} : x \cdot \vec{y} \in R \text{ and } x \cdot \vec{z} \in S\}. \quad (4.106)$$

This is almost like the Cartesian product, except that we take only the tuples that share the same first element and eliminate its second occurrence. With respect to the profile, we proceed slightly differently. On the assumption that $\langle t \cdot \vec{b}_1, R \rangle \in \mathcal{C}$ and $\langle t \cdot \vec{b}_2, \mathcal{D} \rangle \in \mathcal{D}$ we put

$$\mathcal{C} \otimes^t \mathcal{D} := \llbracket \langle t \cdot \vec{b}_1 \cdot \vec{b}_2, R\overline{\otimes}S \rangle \rrbracket_{\mathcal{M}}. \quad (4.107)$$

This is defined only if either (a) both the concepts are at least unary or (b) both profiles contain exactly one t . We extend this definition to the truth concept \mathcal{T} by putting

$$\mathcal{T} \otimes^t \mathcal{D} := \mathcal{D} \otimes^t \mathcal{T} := \mathcal{D}. \quad (4.108)$$

All nouns denote concepts where the one minimal relation has profile t . And so we put

$$\mathcal{I}(m)(\langle e, m \rangle, \langle e', m' \rangle) := \begin{cases} \langle e \hat{\wedge} \perp \hat{\wedge} e', m \otimes^t m' \rangle & \text{if } \gamma(e, e'), \\ \text{undefined} & \text{else.} \end{cases} \quad (4.109)$$

We denote the column with label t by underlining. The derivation begins as follows.


$$\begin{aligned} &\langle \text{cheese}, \llbracket \{ \underline{h}_i : i \in \mathbb{N} \} \rrbracket \rangle \\ &\langle \text{a cheese}, \llbracket \{ \underline{h}_i : i \in \mathbb{N} \} \rrbracket \rangle \\ &\langle \text{ate a cheese}, \llbracket \{ \underline{d}_i, h_{i+1} : i \in \mathbb{N} \} \cup \{ \langle \underline{m}_0, h_0 \rangle \} \rrbracket \rangle \\ &\langle \text{that ate a cheese}, \llbracket \{ \underline{d}_i, h_{i+1} : i \in \mathbb{N} \} \cup \{ \langle \underline{m}_0, h_0 \rangle \} \rrbracket \rangle \\ &\langle \text{mouse that ate a cheese}, \llbracket \{ \langle \underline{m}_0, h_0 \rangle \} \rrbracket \rangle \\ &\langle \text{a mouse that ate a cheese}, \llbracket \{ \langle \underline{m}_0, h_0 \rangle \} \rrbracket \rangle \end{aligned} \quad (4.110)$$

We have only one privileged member. We continue the derivation.

$$\begin{aligned}
 &\langle \text{chased a mouse that ate a cheese, } \llbracket \langle \underline{c_0}, m_0, h_0 \rangle \rrbracket \rangle \\
 &\langle \text{that chased a mouse that ate a cheese, } \llbracket \langle \underline{c_0}, m_0, h_0 \rangle \rrbracket \rangle \\
 &\langle \text{cat that chased a mouse that ate a cheese,} \\
 &\quad \llbracket \langle \underline{c_0}, m_0, h_0 \rangle \rrbracket \rangle \\
 &\langle \text{a cat that chased a mouse that ate a cheese,} \\
 &\quad \llbracket \langle \underline{c_0}, m_0, h_0 \rangle \rrbracket \rangle
 \end{aligned}
 \tag{4.111}$$

The next step is to merge with /saw/:

$$\begin{aligned}
 &\langle \text{saw a cat that chased a mouse that ate a cheese,} \\
 &\quad \llbracket \langle \underline{d_0}, c_0, m_0, h_0 \rangle \rrbracket \rangle
 \end{aligned}
 \tag{4.112}$$

And so on. Thus, the relations are growing in length but retain only one distinguished member. 

The idea of profiling is not new. In formal semantics, referent systems (see Vermeulen (1995)) formalize a variant of profiling. Also Centering Theory implements a notion of profiling (see for example Bittner (2006) and references therein).

Exercise 4.14 Prove Proposition 4.10.