A Web of Words: Lexical Meaning in Context

Nicholas Asher
CNRS, Laboratoire IRIT, UMR 5505
Université Paul Sabatier
Toulouse, France
and
Department of Philosophy
University of Texas at Austin

August 20, 2007

Contents

0.1 Preface

Just over fifty years ago, the publication of ‘Two Dogmas of Empiricism’ by W.V.O. Quine launched a persuasive and devastating attack on the common sense notion of word meaning and synonymy, according to which two terms were synonymous just in case they had the same meaning. Quine’s legacy continues to hold sway among much of the philosophical community today. The theory of word meaning is often thought either not to have a subject matter or to be trivial—‘dog’ means dog. What else is there to say? Well it turns out, quite a lot. Linguists like Maurice Gross (1975), Chuck Fillmore, Igor Mel’cuk, Len Talmy, Ray Jackendoff, Beth Levin, and James Pustejovsky, as well as researchers in AI who have built various on line lexical resources like WORDNET, FRAMENET, and

DOLCE, have worked to give us rich and suggestive descriptions of what words mean. Against this rich descriptive background, however, puzzles have emerged that make it not obvious how to proceed with a formal theory of lexical meaning. In particular, something that is commonly acknowledged but rarely understood is that when word meanings are combined, the meaning of the result can differ from what standard compositional semantics has led us to expect: in applying, for instance, a property term P to an object term a, the content of the result sometimes involves a different but related property P' applied to an object b that is related to but distinct from the original denotation of a. And it is not only the choice of terms that affects the content of the predication; the discourse context in which the predication occurs also affects it. The trick is to untangle from this flux a theory of the interactions of discourse, predication and lexical content. That is what this book is about.[1]

The idea that the meanings of words, or more properly the results of the semantic interactions between words, will shift depending on the other elements in a predication or in the larger discourse context is in some sense obvious if you look at dictionary entries or think about how words combine with other words in different contexts. But working out a precise theory, or even an imprecise one, of how this could go is quite difficult. I will begin slowly with some basic questions and observations.

What is it to give the meaning of a word? There are a number of answers in the literature on lexical semantics or theories of word meaning. Cognitive semanticists like Talmy and Givon, among others, think that meanings are to be represented as some sort of picture or graph; and so a lexical theory should be a theory of those pictures. Gardenfors supplies a more abstract version of such a theory in his recent book *Conceptual Spaces*. Others in a more logical and formal framework like Dowty (but also Jackendoff, Shank and other researchers in AI) take a specification of lexical meaning to be given in terms of a set of primitives whose meaning can be axiomatized. Still others take a denotational view; the function of a lexical semantics is to specify the denotation of the various terms,

[1] I owe many people thanks for helpful comments on this subject: Tim Fernando, Stuart Schieber, Mark Johnson, James Pustejovsky, Hans Kamp, Antje Rossdeutscher, Barbara Partee, Pierdaniele Giaretta, Denis Delfitto, Robin Cooper, Christian Retoret, Christian Bassac, Bruno Mery, Renaud Marlet, Sylvain Salvati, Sylvain Pogodalla, Alda Mari, David Nicolas, Friedericke Moltmann, Tony Veale, Laure Vieu, Claudio Masolo, Stephano Borgo, Julie Hunter, François Recanati, John Hawthorne, Ofra Magidor, Alexandra Aramis, George Bronnikov, Kiki Wang, Magdalena Schwager and participants of the seminars on lexical semantics at the University of Verona and the University of Texas at Austin where some of this material was presented.

typically to be modelled within some model-theoretic framework.

What all of these approaches agree on in some sense is that a specification of lexical meaning must at least consist in the specification of some element, whether representational or not, formal or not, that when combined in a certain way with elements associated other words in a well formed sentence yields a meaning for a sentence in a particular discourse context. This meaning could be understood as truth conditions, some sort of dynamic update condition of the sort familiar from dynamic semantics (Groenendijk and Stokhof (1991), Kamp and Reyle (1993), Asher (1993), Veltman (1996)), or whatever other theoretical reconstruction of meaning suits. For those who still believe that the direct interpretation of English is what a semantic theory should do, then a denotational approach to meaning suffices. For those who use some sort of intermediate representation between natural language and the world to specify meaning, then a lexical entry should specify something that when combined together with the contributions of other words in a well formed sentence will yield a representation that has an interpretation sufficient to yield truth conditions (or whatever sort of meaning you think is appropriate). I shall call such representations *logical forms*. In effect if logical forms can be in the limit identified with sentences of English themselves, then we can see the proponents of direct interpretation as a special case of the logical form view, although the evidence in this book will show that a much more robust notion of logical from is in order.

Thus, at a minimum, lexical semantics is concerned with the lexical resources used to construct logical forms. But what are those resources? One that most lexical semanticists agree on is argument structure. Lexical meaning must include a specification of how syntactically related items combine together; this typically means that a lexical entry of a word *w* must specify whether syntactically related constituents function as arguments of *w* and whether *w* is itself an argument of some other element in the clause. Argument structure is thus fundamentally linked to predication; when one bit of logical form functions as an argument to another, we have a predication relation between a property denoting term and its argument. So in some sense predication is what a theory of lexical information ought to enable us to give an account of, since it is predication that is essential in constructing a logical form and for giving truth conditions, whether those truth conditions be understood in the classical, static way or in the ”update” fashion now familiar from various theories of dynamic semantics. Two principal tasks then of a lexical theory are to: (1) determine lexical meanings and (2) furnish a theory of predication so that lexical meanings can combine together via predication to produce a logical form for a clause and ultimately a discourse.

Chapter 1

Predication and Lexical Meaning

The choice of a model of predication will affect the choice of how to represent lexical meanings. So at least in this sense, a model of predication is prior to an analysis of lexical meaning. One way to analyze predication is to use the lambda calculus. There are other formalisms that can be used; the formalism of attribute value matrices or typed feature structures with unification is one example. But the lambda calculus is the oldest, the most flexible and perhaps the best understood, and its links to various syntactic formalisms are also very well understood (de Groote (2001, 2002), Frank and van Genabith (2001)). Finally, its expressive power will more than suffice for our needs.[1] The pure lambda calculus contains variables (and constants if one wishes) as primitive terms together with the identity predicate and an abstraction operator λ. The set of terms is closed under the following rules: if t is a term and v a variable, then λvt is also a term; and if t and t' are terms then *the application* of t to t' is also a term $t[t']$, and so is $t = t'$. We can apply this language to analyze simple predicate argument structures: a predicate is understood as a lambda term and its arguments are other terms that saturate the lambda bound variables to produce a sentential logical form under the operation of application.

[1] There are other formalisms for predication with structural underspecification—e.g., where the scopes of various functors are not determined, as in Minimal Recursion Semantics (Flickinger, Copestake et al.), which uses typed feature structures. Such structural underspecification arises because of insufficient information about the syntactic parse of the sentence or because the syntax doesn't determine the arguments of the functors in question. The formalism I shall use here extends naturally to a treatment of such underspecification (Pogodalla (2004)). But underspecification will take us afield of our main questions for lexical semantics: what is the semantic content of the expressions themselves and how do they combine with their arguments, whatever they may be? So I will at least for the moment not worry about structural underspecification.

- Application: $\lambda x\phi[\alpha] = \phi(\frac{\alpha}{x})$

Application is the formal counterpart in the λ calculus of the intuitive operation of predication.

Sometimes predications go wrong. This is also something that lexical semantics has to explain.

(1.1) a. ?That person contains an interesting idea about Freud.

b. That person has an interesting idea about Freud.

c. That book contains an interesting idea about Freud.

d. That person is eating its breakfast.

e. That book is red.

f. #That rumor is red.

g. # The number two is red.

h. # The number two is soft.

i. The number two is prime.

A predication like (1.1f) or (1.1g) is malformed—they contain what Gilbert Ryle would have called a category mistake. Numbers as abstract objects can't have colors, so it's nonsensical to say something like the number two is red, unless the context enables us to understand this in some metaphorical or indirect way. A theory of predication and a theory of lexical meaning should reflect these facts.

There are also degrees of category mistakes. For instance, one has to exercise some care in understanding why a predication like (1.1a) sounds so much odder than (1.1bcd). In some sense people can contain information: spies have information that they give to their governments and that counter spies want to elicit; teachers have information that they impart to their students. But one can't use the form of words in (1.1a) to straightforwardly convey these ideas. The predication is odd; it involves a misuse of the word *contain*. If it succeeds at all in making sense to the listener, it must be subject to some reinterpretation, often as metaphor.

The reason why (1.1a) is just not the right way to say what one wants to say (as opposed to (1.1b)) is because, as with (1.1f), there is a conflict between the demands of the predicate for a certain type of argument and the type of its actual argument. People aren't the right type of things to be containers of information, whereas tapes, books, CDs, and so on are. This humdrum observation leads to a substantial modification of the lambda calculus, a lambda calculus with types.

The reason in such a theory why some predications involve misuses of words, don't work or require reinterpretation, is because the types of the arguments don't match the types required by the predicates for its argument places. (1.1a) involves a misuse of the language, because *contain* requires for its subject argument either a physical container or a container of information, and persons are not of the type information containers—that's just to say that they don't contain information the way books, journal articles, pamphlets and the like do. On the other hand, there is no such problem with (1.1c). Notice that (1.1c) and (1.1b) though both perfectly fine make the relation between information and container quite different. The difference has to do with the difference between the type of things that *contain* information, like books, cd's, tapes and so on, and people.

Such observations suggest a constraint on the fundamental operation of Application. Assume that every term and variable in the lambda calculus is assigned a type by a function TYPE.

- Type Restrictied Application: $\lambda x \phi[\alpha] = \phi(\frac{\alpha}{x})$, provided TYPE$(x)$ = TYPE(α). $\lambda x \phi[\alpha]$ is undefined, otherwise.

In what follows, I'll abbreviate TYPE to an assignment of the following form; to say that term α has type a, I'll write $\alpha\colon a$.

I will start off by modelling predication as type restricted application. This will require that each term gets a type in a given predicational context. Type concordance between predicate and argument is a matter of presupposition for the clause as a whole. If an argument in a predication cannot satisfy the type requirements of the predicate, then the predication cannot be literally interpreted and fails to result in a well formed proposition capable of having a truth value. This implies that semantically anomalous sentences like (1.1g, h) remain anomalous when embedded within interrogative mood or modal operators. This is indeed the case:

(1.2) a. # The number two might be red.

b. # Is the number two soft?

c. # The number too isn't soft.[2]

A feature of presuppositions that will be very important for the study of predication in this book is the ability of some presuppositions that are not satisfied to be

[2] A presuppositional view should allow that this sentence has a perfectly fine reading where the negation holds over the type requirements as well. But typically such readings are induced by marked intonation. If this sentence is read with standard assertion prosody, then it is as anomalous as the rest.

"accommodated". In the literature on presupposition, it is standardly assumed that the adverb *too* generates a presupposition that must be satisfied in the discourse context in which it is uttered by some *linguistically expressed* content. Thus, in an out of the blue context, it makes no sense to say,

(1.3) Kate lives in New York too.

even though as a matter of world knowledge it is clear that the presupposition of *too* in this sentence is satisfied—namely, that there are other people besides Kate who live in New York. Even if the proposition that there are other people besides Kate who live in New York is manifestly true to the audience of (1.3), (1.3) is still awkward, unless the presupposed content has been made salient somehow in the context. The presupposed, typing requirements of the predicates in (1.1) and (1.2) resemble the behavior of the presupposition of *too*; they have to be satisfied in their "predicative" context in order for the sentences containing them to receive a truth value. Some of those sentences fail to express a coherent proposition capable of having a truth value, because the relevant presuppositions cannot be satisfied, given that the arguments and predicates therein mean what they standardly mean and have the types that they standardly do. On the other hand, we will see that sometimes when a predicative context fails to satisfy certain typing requirements, the context can be "rearranged" or modified in some way so as to satisfy the presupposition. In such cases the typing presuppositions resemble the presuppositions of possessive noun phrases. For instance, *Sylvain's son* presupposes that Sylvain has a son, but this information is readily accommodated into the discourse context when the context does not satisfy the presupposition.

(1.4) Sylvain's son is almost three years old.

Figuring out when presupposed typing requirements can be accommodated and when they cannot will be a central task of this book.

Type restricted application is the central rule of what is known as the *typed lambda calculus*. It has been extensively studied and has many pleasant semantic and computational properties, especially in comparison to the untyped lambda calculus. This has made it the staple of compositional semantics since Montague first pioneered the model theoretic notion of meaning in the sixties. The data we have just seen, however, requires hypothesizing a great many more types than Montague Grammar ever envisaged.

1.1 Distinguishing Between Types

An analysis of predication that uses a rich system of semantic types is only promising if there is good evidence that languages encode such a system. Luckily, there is evidence that information about types is so far conventionalized that it even affects the case system in some languages. There are many ontological categories that languages reflect in their grammar. For instance, the distinction between abstract object and concrete or physical objects permits us to distinguish between the malformed (1.1f,g) and well formed predications involving predicates that denote properties of physical objects like colours and terms that denote physical objects.

But many other distinctions between types of objects exist. Here is just a partial list. The distinction between eventualities (events, processes and states) and non-eventualities (other sorts of objects) allows us to predict general patterns of predication. Events occur at times but objects don't; adverbs of manner of motion go well with events, but their adjectival counterparts fail to predicate felicitously of many noneventualities, as the minimal pairs below in (1.5) and (1.6) demonstrate:

(1.5) a. John's birth occurred at 10 am this morning.

b. #John occurred at 10 am this morning.

(1.6) a. The tree grew slowly.

b. ?The tree was slow.

Languages like Japanese reflect the distinction between eventualities and objects in the grammar within the system of particles. Chinese reflects the distinction between eventualities and objects within the system of classifiers.

Besides this distinction, work on eventualities has shown that languages are sensitive to the differences among eventualities. Dowty (1979), Vendler (1957) among others notice that punctual events like achievements don't go well with adverbials that express the fact that the event they modify took place over an extended interval of time whereas activities do.[3]

(1.7) a. #John died for an hour.

b. John ran for an hour.

[3] See also Krifka (1987), Rothstein (2003) for more discussion of this data.

Thus, further distinctions between different sorts of eventualities help explain patterns of predication involving temporal adverbials.

The distinction between states and other eventualities shows up in other parts of the grammatical system. For instance, verbs that denote states in general do not accept the progressive form in English, while verbs that denote events do.

(1.8) a. #Samantha is knowing French.
b. Samantha is running.
c. Arnold is dying.

Japanese marks a distinction between animate and inanimate in its predicative verb structure, which is related to the difference between event types and non event types (but does not exactly correspond to it).

- iru(x) x: animate
- aru(x), x: inanimate (event)

Another example of a grammatically grounded distinction between types in natural language has to do with a distinction between locations and physical objects. There is considerable evidence that languages distinguish types for places (fixed elements in the terrestial reference frame) and types for objects (elements that have a complex internal structure and can move with respect to the terrestial reference frame). Some evidence for this distinction comes from Basque, where the grammar encodes differences between location and objects via two genitive cases -*ko* and -*ren*; locations in general easily take the genitive -*ko* but not -*ren*, while objects in general do the reverse (Aurnague 2001). Aurnague (2001) distinguishes the following sortals: *places* (e.g., valley, field, river, mountain, hill), *objects* (e.g., apple, glass, chair, car), and *mixed objects* (e.g., house, church, town hall). Of particular interest are the "mixed objects" and the behavior of their expressions in Basque. The terms for mixed objects readily accept both forms of the Basque genitive. So if we accept the encoding hypothesis for Basque, mixed objects like houses would appear to belong to two types, or two ontological categories, at the same time—LOCATION and PHYSICAL-OBJ—neither of which is a subtype of the other (it is neither the case that the properties associated with physical objects are inherited as properties of places nor that the properties associated with places are inherited as properties of physical objects).

(1.9) Maite dut etxeko atea haren paretak harriz eginak direlariak.
(Michel Aurnague p.c.)
I like the door of the house the walls of which are made of stone.

Prepositions in English serve to distinguish between places or locations and physical objects, though the distinctions are less clear cut than in Basque (Asher 2006).

The mass/ count distinction is another distinction marked in many languages. Certain determiners in English are designated as mass determiners—e.g. *much*, as in *much water*, *much meat*. They do not go with count nouns in general—e.g., *much person, much people* are malformed. Other determiners like *many, every* and so on go well with count nouns but require special interpretations when put with mass nouns. Thus *every water, many waters* must range over portions of water or perhaps kinds of water. Work in semantics on mass and count nouns has firmly established a difference between these two types of entities, as well as the differences between these and a third type: kinds. Kinds are often expressed in English with a bare plural noun phrase (*cats, numbers, people*) but can also be expressed with other constructions:

(1.10) The Mexican fruit bat is common in this area.

Linguists in general take the definite noun phrase in sentences like (1.10) to refer to a kind rather than to range over individual members of the kind. They argue, quite sensibly, that the predicate *is common in this area* cannot hold of an individual but only of kinds or species. Such predicates along with others like *is extinct, is widespread* encode a three way distinction between the types of masses, countable individuals, and kinds.

Another type distinction encoded in the system of prepositions in English involves containers and containables. Agents put things inside containers, and in general anything that describes a specific volume or enclosure can be a container. Thus, many physical objects can serve as containers. But some cannot.

(1.11) a. The water is inside the pitcher.

b. The keys are inside the car.

c. John put the keys inside his pocket/inside the drawer.

d. # John put the keys inside the air.

e. # John put the wine inside the water. versus: John put the wine in the water.

In general, I will try to establish syntactic or lexical alternations to distinguish between types. That is, positing a type distinction in a lexical theory will require linguistic evidence: there must be a linguistic construction that accepts expressions of one type but not the other. This is at least a minimal condition.

1.2 Different Sorts of Predication

Having introduced types as part of the apparatus of predication, let me come back to predication itself. I have spoken so far of predication as a single operation of applying a predicate to its arguments or a relation on the terms involved. But in fact there are quite a few different forms of predication in natural languages, some particular to particular languages, others more general.

- ordinary predication of various sorts:
    - predication of a verb phrase to a subject or a transitive verb to an object
    - adjectival modification with different types of adjectives (evaluative adjectives like *good rock*, *bad violinist*, material adjectives like *bronze statue*, *paper airplane*, manner adjectives *fast car, slow cigar*
    - adverbial modification
- metaphorical usage (extended predication)

  (1.12) John is a rock.
- restricted predication

  (1.13) John as a banker makes $50K a year but as a plumber he makes only $20K a year.
- copredication

  (1.14) The lunch was delicious but took forever.

  (1.15) The book has a purple cover and is the most intelligible introduction to category theory.

  (1.16) #The bank is rising and specializes in IPOs.
- loose predication:

  (1.17) That's a square (pointing to an unpracticed drawing in the sand).
- vague predication:

  (1.18) That's blue.
- resultative constructions:

(1.19) a. Kim hammered the metal flat.
b. * Kim hammered the metal gleaming.

(1.20) depictives
a. Pat swims naked.
b. *Pat cooks hot.

- plural predication

(1.21) a. The students surrounded the huge building/ gathered in the square.
b. ??Most students surrounded the huge building/gathered in the square. .
c. #John surrounded the huge building/gathered in the square.

- the genitive construction

(1.22) a. Kim's mother
b. Kim's fish

- noun noun compounding

(1.23) a. lunch counter
b. party favor

Each one of these forms of predication presents its own challenges for semantics and for lexical theory in particular, since lexical theory must assign to the constituent words in these constructions the right sort of meaning so as to get the right result together with the composition rules it postulates for modelling the predication itself.

Loose predication and vague predication, or the predication of vague properties to objects, are difficult and well known problems in philosophy (it is not entirely clear that these *are* problems of predication, though they are in ordinary parlance referred to as forms of predication). But other forms of predication mentioned above, which also provide challenges for lexical theory and have recently received discussion in linguistic circles, have not received so much scrutiny. Co-predication, for instance, turns out to be a major challenge for typed theories of predication. I mentioned earlier that languages distinguish between events and objects, as different sorts of entities; the predicates of the one type do not predicate of the other type literally. It turns out that some objects, however, are both events *and* physical objects in some sense. Consider, for instance, lunches. Lunches can be events but they are also meals and as such physical objects. We can make felicitous *copredications*, a grammatical construction in which two predicates jointly apply to the same argument. In particular, in (1.24), one predicate selects for the event sense of *lunch* while the other selects for the physical object or meal sense:

(1.24) Lunch was delicious but took forever.

I will call such predications *aspect selections*; I will analyze these predications as predications that apply to selected aspects of the object denoted by the surface argument.

In trying to account for instances of copredication that involve aspect selection like (1.24), standard, typed theories of predication and lexical semantics confront some difficult if not unanswerable questions. How can a term have two incompatible types, as is apparently the case here? How can one term denote an object or set of objects to which apply two properties demanding different, even incompatible types of their bearers? It would have to be the case then that such an object must have, or belong to, two incompatible types. But how is that possible? Proponents of standard type theory have only one clear recourse, and that is to claim that terms associated with two incompatible types are ambiguous. But that deepens the mystery about copredications involving aspect selections: if, say in (1.24) *lunch* is ambiguous between a physical object reading and an event reading, then we must disambiguate the term in one way to make sense of the first predication but disambiguate it in a second way to make sense of the second predication; and the problem is that, on the surface at least, we have only one term to disambiguate—we have to choose a disambiguation, but such a choice will inevitably cause one of the predications in (1.24) to fail. At this point we might try a strategy of desperation and postulate a hidden “copy” of the problematic term, rewriting (1.24) in effect as:

(1.24’) Lunch was delicious but lunch took forever.

This copying strategy now allows the proponent of standard type theory to proceed to disambiguate the two occurrences of *lunch* in different ways allowing the two predications to succeed. But the promise of the copying strategy is shortlived. Copying expressions will get us incorrect truth conditions in many cases. Consider (1.25), where *finish* applies to events like eating a lunch, making a lunch and so on, while *tasted* applies only to objects (you can’t taste events except metaphorically):

(1.25) Mary gingerly tasted and then finished a lunch.

The copying strategy forces us to interpret (1.25) as

(1.26) Mary gingerly tasted a lunch and then finished a lunch.

It's easy to see that (1.25) and (1.26) have different truth conditions; (1.25) is true only in those situations where Mary gingerly tasted and finished the same lunch, while (1.26) can be true in situations where Mary gingerly tastes one lunch but finishes another. Thus, it is not obvious how to deal with examples of copredication even from the standpoint of compositionality, if one's lexical theory produces a rich system of types. Montague himself noted that there were copredications that were puzzling even within his much more impoverished system of types. In (1.27) *temperature* seems to have two aspects, one of which is a number on a scale, while the other is a function from times to numerical values.

(1.27) The temperature is 90 and rising.

Restricted predications seem to predicate properties of certain parts or aspects of their arguments. But it is not clear in what sense we should understand the word "part". When we say *John as a banker makes $50K a year*, it's not as though we're predicating something of some physical part of John as in *John's right arm has a bruise*. Restricted predication thus introduces some metaphysical and linguistic puzzles of its own.

Resultative constructions add a causal link between the result state, where the metal is flat, and the activity of hammering the metal itself. This sentence involves two predications on the term *the metal*—one by the adjective *flat* and one by the verb.[4] But further, this construction introduces a third predication, which features the causal relation. Genitive constructions also introduce a predication in which some relation is predicated of the objects described by the two noun phrases or DPs that make up the genitive construction. Sometimes this relation is given by the head noun if it is relational as in (1.22a) but sometimes it is not as in (1.22b); here the predication again seems to add to or to change the content of the words in the construction. The compounding of nouns in English, a semi productive predication device, also often yields a meaning that seems to go beyond simply fitting the meaning of the parts together following standard semantics—sometimes in radical ways so as to produce idioms whose meaning is not derivable from the meanings of its constituent terms like *party favor*.

[4] Syntacticians take the structure of this sentence to be quite complex, involving what is called a "small clause", *the metal flat*, and a verb *hammer*.

Plural predication shows that some predicates require a certain type of predication, *collective* or *group* predication, which requires the argument to be described in a certain way. In general in the study of plurals, three types of predication are distinguished: distributive, collective and cumulative. Distributive predication occurs when a property or relation is predicated of each element of a set as in

(1.28) The boys each worked hard.

On the other hand, a different form of predication seems to be involved in

(1.29) a. The students surrounded the building

b. The students mowed the whole meadow.

(1.29a,b) exemplify collective predications where a property is predicated of the whole set of students but not of each student individually. Sometimes collective predication occurs with singular nouns (and so this semantic phenomenon must be distinguished from the syntactic phenomenon of number).

(1.30) The committee is meeting in the lounge (group predication)

A cumulative predication does not tell us how the predication itself, exemplified in (1.31)

(1.31) Three girls danced with four boys.

is distributed, only that there were three girls and four boys and dancing went on between them. There is a grammatically marked distinction between plurality and singularity. How does that play into lexical meaning or compositional meaning? Most lexical theories have nothing to say about this. Nevertheless, they should. For instance, an account of the lexical meaning of the word *disperse* must mark it as requiring an argument that must be interpreted collectively. It does not go well with inherently distributive quantifiers like *most students*, whereas the predicate applies perfectly to plural noun phrases that can be interpreted collectively

(1.32) a. ?Most students in the square dispersed.[5]

b. The students in the square dispersed.

[5] Note that the partitive DP *most of the students* does admit of a collective interpretation, unlike the straight quantified DP *most students*.

Other predicates must be interpreted distributively. Thus features of the semantics of plurals must find some reflection in a theory of lexical semantics.

Even "simple" predications are not so straightforward. In many predications, what appears to be on the surface a predication of a property to some sort of object turns out to be something more. This phenomenon has been called "coercion", because it appears that one word coerces another word (usually the second word is an argument of the first) to have a different meaning from its usual one.

(1.33) a. good lunch, good rock, good children

b. Mouse isn't very tasty except if you're a cat.

c. John started a cigarette, started a car, began the sonata, finished the novel.

d. John liked the dress with the flowers, liked the garden with the flowers.

In each of these examples, the predications must be understood in a particular way. (1.33a) shows how adjectival modification can affect the type and meaning of the resulting noun phrase. A good lunch is one that tasted good. Pustejovsky (1995) and others have developed a theory of lexical meaning according to which an adjective like *good* selects a component of the meaning of its argument—roughly, its purpose or *telic* role. Nevertheless, as many have noted, such adjectives also apply to arguments that don't have purposes. For instance, rocks in and of themselves don't have purposes, and neither do children (in a post Aristotelian universe); yet when *good* modifies *rock* or *children*, we understand different things: when someone says *this is a good rock*, we understand the rock in this case as having some purpose for which it is good. When someone says *those are good children*, we understand that the children are behaving in a particular way or that they have certain laudatory dispositions to behavior. There is a subtle, though undeniable, shift in meaning in these predications. A theory that simply says that 'good' denotes the property of being good, that 'children' denotes children and 'rock' denotes rocks and that says nothing about how these meanings combine in predication other than that the objects denoted by the one term have the property denoted by the other cannot begin to make any headway explaining these nuances in meaning. Unfortunately, many philosophers and some linguists still hold such a theory to be true (for instance see Fodor and Lepore (1998)).

The phenomenon observed with the adjectival predications in (1.33a) is a very general and diverse one. (1.33b) shows how the bare singular use of a count noun can in many circumstances change the type of the noun phrase from count to mass.

This transformation is known as *grinding* in the semantic literature. The examples in (1.33c) show how aspectual verbs coerce their arguments into denoting some sort of event. Aspectual verbs require or presuppose that their direct object is some event involving their subject; when their direct objects are not event-like, a felicitous coercion sometimes occurs, and we infer defeasibly that some sort of activity involving the subject of the aspectual verb serves as its internal argument. Thus, we understand *John started the car* as John's started the running of the car's engine. To start a cigarette is typically to start to smoke a cigarette. (1.33d) shows that coercions can happen with prepositional phrases—*the dress with the flowers* has at least one interpretation where a representation of flowers is stitched, printed or drawn on the fabric of the dress, while *the garden with the flowers* does not have that interpretation, at least not nearly so saliently.[6] As we shall see, there are subtle differences with respect to the presuppositions in the typing requirements of various aspectual verbs and other coercing predicates.

There arises the question of whether such coercions are really part of lexical *semantics*. Another way to ask the question is whether it's defeasible but *a priori* inference that If John started the car, John started the engine of the car or that if Julie enjoyed the book, then (defeasibly) she enjoyed reading it. That is, do such inferences follow solely from one's linguistic mastery of the language? Fodor and Lepore think that none of these inferences belong as part of lexical semantics. Those interested in "minimal semantics" like Cappellen and Lepore (2004) will also assign such inferences to world knowledge. On the one hand, however, most people can distinguish between largely automatic interpretations that these predications seem to entail and those that require more conscious effort. One might take that to be a mark of the information as being present even during predication rather than inferred afterwards using background, nonlinguistic beliefs.

It is notoriously difficult to distinguish between what is properly a part of lexical meaning and what is world knowledge. Quine's attack on lexical meaning can be seen as starting from the point that one cannot make this distinction in a principled way. Part of the difficulty is that, to some extent, the division between word meaning and world knowledge is a theory internal distinction. For instance, if you're an externalist for whom the meaning of two singular or natural kind terms t and t' is determined by their reference, it may be a fact of meaning that $t = t'$ or not. That is, *water is* H_20 would be analytic on such a view! Nevertheless, there is one test one can use to see whether certain information is conventionally associated with a particular word meaning rather than just the general content. If

[6] This example is due to Marliese Kluck.

we can find an expression that expresses the same content as a particular word, but the defeasible inferences associated with the word disappear when we employ the other expression, then this is a strong indication that the inference is in some way conventionally associated with the word as part of the linguistic system. As an example let's consider replacing the word *cigarette* with the relevant part of its entry in *Webster's New World College Dictionary*, which should at least roughly have the same content as the word *cigarette*.

(1.34) a. Nicholas enjoyed a cigarette.

b. Nicholas enjoyed a small roll of finely cut tobacco wrapped in thin paper.

Speakers immediately get the defeasible interpretation of (1.34a) where Nicholas smokes a cigarette but not in the second case. Moreover, these alternations appear to be pretty systematic (with words like *book, novel, sonata*, and so on), indicating that indeed such defeasible interpretations are a part of lexical semantics.[7]

Another test is to see whether these sort of inferences are specific to a particular type of verb. Let's suppose that *cigarette* always has associated with it a possible event reading. It should then be possible to access that appropriate event reading with other predicates that take events.

(1.35) a. Nicholas's smoking of that cigarette will begin in 2 minutes.

b. Nicholas's cigarette break will begin in 2 minutes.

c. ??Nicholas's cigarette will begin in 2 minutes.

It's quite clear that (1.35c) is semantically strange. The event associated with *cigarette* in *enjoy the cigarette*, *begin the cigarette*, *finish the cigarette*, just isn't available with other event predicates. This would suggest that there is some particular conventional meaning that issues from the predication of the properties these verbs denote to the objects denoted by their arguments that isn't available in other predicational contexts. That is, the eventuality of smoking isn't just accessible

[7] Laura Whitten and Magda Schwager independently observed to me that using the dictionary definitions is marked, and Gricean maxims would predict that there are some special uses of that word. The standard meaning or association would be blocked. But if the inference here concerns nonlinguistic knowledge, we wouldn't expect the flouting of the Gricean maxim to block such an inference. A Gricean explanation of why the inference fails for (1.34b) occurs precisely because that inference is based on lexical content, not world knowledge. This nevertheless gives rise to some questions: Is there a general test here that we can use? Should world knowledge be closed under the substitution of intensionally or just extensionally equivalent terms?

with any predication involving *cigarette*; it is the result of combining *cigarette* as an object or internal argument of an aspectual verb or a verb like *enjoy*.

Coercions, as Aristotle said of all familiar things, are easy to see but hard to understand. Some linguists have argued that in fact coercions are what they appear to be. They indicate that the meaning of terms fluctuates from context to context, and some have taken the moral of these observations to be some sort of radical contextualism about meaning (for instance, Recanati (2004, 2002)). But these conclusions do not follow from the evidence. They are also vastly counterintuitive: when I say that I enjoyed the cigarette, does the word 'cigarette' now all of a sudden change its meaning to mean 'smoking a cigarette'? It does not seem so. Fodor and Lepore (1998) and more recently **?** correctly, in my view, crticize such an approach to coercion, an approach which they take Pustejovsky (1995) to espouse as well when he speaks of "generativity" in the lexicon. When it comes to technical developments, I shall show that in fact basic word meanings cannot change if we are to be able to derive any predictions at all about lexical meaning.

If that approach to coercion is wrong, however, what is the right approach? Coercion is a ubiquitous, attested phenomenon in natural language; one has to be able to give an analysis of it in any remotely viable theory of predication. In order to explain the data one has to do one of two things: either one has to develop a theory of lexical meaning where the lexical entries themselves change in context, or one has to complicate one's notion of predication and logical form. Despite many claims that the lexicon is in fact generative or context sensitive in some radical way (Pustejovsky (1995), Recanati (2002)), I do not know of any formally worked out proposal of this view.[8] As we shall see below, the generative lexicon of Pustejovsky (1995) has static lexical entries that do not change during coercion. On the other hand, one can have the intuition that coercion phenomena call for a reanalysis of predication. When I say that I enjoyed a cigarette, the word *cigarette* does not change its meaning but what I enjoyed is doing something with the cigarette. That is, coercions involve a more complex act of predication than one might have thought.[9] This book will pursue this line of attack on coer-

[8] Although one possibility for formalisation would be in a connectionist approach, where word meanings are thought of as vectors of strengths of associations with other words that get recalculated every time the word occurs. This is very far from either Pustejovsky's generative lexicon or from contextualist's work like that of Recanati. It's also philosophically and conceptually extremely unsatisfying, as such an approach doesn't begin to tell us anything about lexical meaning or lexical inference or about how meanings compose together.

[9] Classic GL says little about predication. Because it fails to carry through on either way of analyzing the data, we shall see that classic GL fails to account for the coercion data as well as for

cions as part of a general approach to predication and make it formally precise. Viewed from the perspective of this book, coercion is not really a problem about meaning change in the lexicon; it's a problem about compositionality—about how lexically given meanings combine together in the right sort of way. I argue for a similar conclusion for nouns like *lunch*, *book*, *temperature* and so on, which appear to have multiple aspects or senses that can be accessed simultaneously in copredications. To account for many features of predication, the logic of meaning composition has to be rethought and revamped considerably from the standard approach to predication that underlies Montague Grammar. This is what I devote myself to in the bulk of this book.

1.3 The Context Sensitivity of Types

We want a theory of lexical information that offers a framework within which empirical research will yield a correct account of lexical content. I have argued that a lexical theory has to do two things to reach this goal: give an account of the meanings of lexical items and an account of the operation of predication needed to derive truth conditions for clauses. But in order to capture the observations and intuitions of most of those who have worked in the field of lexical semantics, we need to do this in a particular way: we need to construct a discourse sensitive theory of lexical meaning and predication, and that means a context sensitive theory of typing.

Observations that confirm this last claim have been around for years. Nevertheless, there have been few attempts in the literature to account for these observations. Lascarides and Copestake, for instance, in Lascarides and Copestake (1995) noticed that the event readings of the object of a verb like *enjoy* can depend on discourse factors. Normally *enjoy* coerces its object or theme argument that has the type BOOK into an expression that involves an event of reading the book as in (1.36a). But this reading depends on the assumption that Julie is a person, someone capable of reading the book, because this reading is not available with (1.36b). It becomes available if (1.36b) occurs in the context, say, of a fairy tale in which goats are capable of reading.

(1.36) a. Julie enjoyed the book.

b. The goat enjoyed the book.

the copredication data.

Let's assume, as do most stories about coercion (Pustejovsky (1995), Nunberg (1979), Egg (2003), Asher and Pustejovsky (2004)), that the object argument of *enjoy* is some sort of eventuality, which is the result of a typing adjustment due to a clash between the type of argument the verb demands and the type of argument that is in fact its direct object; the verb *enjoy* requires an event as object argument and so coerces the direct object into giving an argument that is of some event type. Regardless of the details of how this coercion process actually works, (1.36ab) shows two things. First, the inference from *enjoy the book* to *enjoy reading the book* must be defeasible. Second, the fact that we can get the reading that the goat enjoyed the book in (1.36b), given a discourse context in which goats talk, shows that the typing and typing adjustment rules must be sensitive to information in discourse.

Danlos (2005) has shown that aspectual verbs are also sensitive to discourse context. Aspectual verbs take some sort of eventuality as an object or theme argument. In studies within the GL framework, these eventualities have been assumed to be given by the lexical entries of the DPs in theme or in an adjoined PP—e.g., *start a cigarette* is to start smoking a cigarette. But the examples of Danlos show that these eventuality inferences must sometimes be a matter of discourse context.

(1.37) a. ??Yesterday, Sabrina began with the kitchen. She then proceeded to the living room and bedroom and finished up with the bathroom.

b. Yesterday Sheila cleaned her house. She began with the kitchen. She then proceeded to the living room and bedroom and finished up with the bathroom.

c. Last week Julie painted her house. She started with the kitchen. She then proceeded to the living room and bedroom and finished up with the bathroom.

The examples in (1.37) show that the eventuality is *not*, at least in all cases, given by the lexical entry of a noun in the theme argument of the verb or some adjoined PP. When a discourse is not clearly about any particular eventuality as in (1.37a), the coercions induced by the presence of aspectual verbs are rather bad; there is no way to recover an associated eventuality to serve as the theme argument of the aspectual verbs. On the other hand, when the discourse is clearly about some eventuality, then these aspectual verbs naturally get associated with that eventuality.

Skeptics at this point might say that this is not a matter of linguistics at all but cognitive psychology.[10] The judgements in (1.37) might be a matter of the interpreter's being primed with a recent mentioning of some event that can be used to fill in the needed event arguments to the aspectual verbs. However, if these judgements were merely a matter of priming, then the structure of the prior discourse shouldn't affect the judgements. Yet it does. Contrast (1.37c) with (1.37d):

(1.37d) ?Last week Julie painted her house. Then this week she started with the kitchen. She then proceeded to the living room and bedroom and finished up with the bathroom.

This discourse is once again much more problematic and resembles (1.37a) in that it's no longer clear what is the event argument of the aspectual verbs *start, proceed* and *finish up*. (1.37c) and (1.37d) form a minimal pair and provide strong evidence that furnishing an event argument to these aspectual verbs is a linguistic matter that involves an interaction between the discourse context and how it is structured, lexical meaning and the construction of the meaning of an individual clause. It is not a matter of language independent world knowledge or of psychological priming.

It's not just the aspectual verbs whose coercive force is sensitive to discourse context. Discourse context can push us to reinterpret many expressions:[11]

(1.38) a. I went to the gallery after the robbery. The elephant had been stolen.

b. I went to the zoo after the robbery. The elephant had been stolen.

Discourse context can also affect the interpretation of other forms of predication. The genitive is one such predicational construction where there is a relation inferred to hold between variables or constants introduced by the DPs in that construction. Sometimes this relation is specified by a relational noun in the genitive construction—e.g., *John's mother.* However, Asher and Denis (2004) note that discourse based information can override the relation provided by the lexical meaning of relational nouns:

(1.39) [Context: Picasso and Raphael both painted a mother with a child.]
Picasso's mother is bright and splashy —a typical cubist rendition. It makes Raphael's mother look so somber.

[10] Jerry Fodor put this point to me on a related matter after a lecture on discourse structure at Rutgers in 2005.

[11] These examples are inspired by examples from Marliese Kluck.

And where no relational noun is involved, discourse context can also affect how to interpret the underspecified relation introduced by the genitive.

(1.40) a. All the children were drawing fish.

b. Suzie's salmon was blue.

As Asher and Denis (2004) and a number of other researchers (Vikner and Jensen 2001, Partee and Borshev 1999) have argued, the genitive construction seems to be one that is partially determined by the lexical entries of the nouns involved—and Asher and Denis argue that in particular it is the types of the nouns involved that help determine the relation. But the examples above show once again that the meaning of this construction is discourse sensitive in a way that previous analyses of the interaction between discourse and the lexicon (cf. Lascarides and Copestake 1996) were unable to capture.

These examples of discourse sensitivity show something of much more general interest as well. The eventualities that are inferred to be the arguments of the aspectual verbs in, for example, (1.37b,c) are inferences that are not based on general conceptual or world knowledge but on information that is conveyed *linguistically* in prior discourse. This must be so, since if these inferences were driven by non linguistic knowledge then (1.37a) should be fine when it isn't. We haven't shown, of course, that these mechansims cannot be guided by non-linguistic information; but we have shown that they are not themselves simply a matter of world knowledge. The discourse sensitivity of the mechanism shows us something that the simple predicational examples cannot do straightforwardly (though see the last paragraph of the last section of this chapter); the coercion mechanisms are part of the linguistic system.

1.4 The Main Points of This Book

Examples of simple predication in which coercions occur like those in (1.33a,c) have received an extensive discussion in the lexical semantics literature, to be sure within the framework of the Generative Lexicon (Pustejovsky (1995)) as well as within other approaches.[12] But a satisfactory analysis of other types of coercion and phenomena like copredication or relative predication are lacking, and even the analyses of the standard examples of coercion within a typed framework need a good deal of sharpening and refinement. Finally, no one to my knowledge has

[12] See also Kleiber (1999), Egg (2003), Recanati 2005, for example for other discussions.

really investigated the interactions between discourse and lexical meaning of the sort introduced in the previous section.

Guided by this data, I will develop a theory of predication and lexical meaning in this book and an account of how it interacts with discourse structure. I will use the theory to investigate phenomena like copredication, relative predication, and coercion. The guiding idea, already implicit to some extent in Pustejovsky (1995), but made much more explicit in Asher and Pustejovsky (2004), is that almost all words will have single lexical meanings. Words like the nouns *cat*, *lunch*, *book* or the verbs *kill*, *read* and *master* contribute simple property or relation symbols to the logical forms representing the truth conditional content of the clauses in which they occur. I will represent these contributions via lambda terms like $\lambda x\, \mathrm{cat}(x)$. But these words all come with a rich amount of information in the types assigned to the lambda terms and the variables within them. When words are combined together to form clauses, sentences and discourses, the types associated with various terms will interact in complex ways; I will introduce new operations of type adjustment in response to type mismatches between predicate and argument. The effects of these type adjustments at logical form is that the logical form will contain elements that are not present in the lambda terms for the constituent words themselves. Predication is in general here much more than a matter of simply applying a function to an argument. Coercion and the sort of problematic copredications that I introduced earlier will invoke particular type adjustment operations with transfer effects onto logical form. As we will see, however, predication is not simply a matter of putting well formed lexical meanings together and adjusting them when they do not fit; type information will be to some extent dependent on the lexicon and to some extent dependent on the discourse context. The theory to be presented here will develop the connection between typing and contexts so as to account for the context sensitivity of type assignment.

While I will use the well-known framework of the first order typed lambda calculus to formalize the theory, I will extend the typed lambda calculus beyond the usual set of simple types and functional types to include several complex types. These complex types furnish the basis for my analysis of certain kinds of copredication and coercion. One complex type, the $\bullet$ type, will be used to analyze certain terms, those for which we can predicate properties of two different aspects of the same thing. I will argue that this kind of predication requires a special metaphysical conception of the objects whose aspects are the bearers of the properties predicated. The type adjustment with dual aspect nouns is, in some sense, just a shifting of emphasis or a reconceptualization of the very same object. Other coercions like those involving a shift from objects to eventualities, the sort of coercions induced by verbs like *enjoy*, shift the predication entirely away from the original object to some other object of a different type—typically, an eventuality associated with the original object. Such coercions may support predication but they do not affect the way the objects denoted by the term that is subject to the coercion are counted or individuated. I will model this second class of coercions with another sort of complex type, something that I shall call a *dependent type*.

I will provide rules for introducing and exploiting these types in the analysis of predication, thus giving these types a proof theoretic meaning. Rules for exploiting and introducing these types will allow us to change or to select the appropriate type of the argument for the requirements of the predication at hand. In addition I will show how the type system is sensitive to the discourse context. By integrating a theory of discourse structure and discourse contexts from earlier work (Asher (1993), Asher and Lascarides (2003)) within the theory of predication, I will show how discourse can transform and constrain type assignments and type transformations. The type system is dynamic and evolves as discourse proceeds in a way similar to the way that linguists and philosophers have argued that the semantics of discourse must be dynamic.

Another crucial ingredient in my approach is to distinguish between the logical forms constructed during predication and the types that guide and constrain predication. When we shift the type of a term from, say, a physical object to an event, we must also register this difference at the level of logical form. As part of the analysis of predication then, I will introduce *transfer* rules that tell us what are the effects of a type shift on the logical forms of the terms in the predication. I will argue that the distinction between logical forms and types is crucial to semantics. We need types to construct logical form, but the semantics of the type system, given by a proof theoretic or internal semantics, is not sufficient to give the semantics of a natural language. I will argue for a “two stage” or two level semantics for lexical meaning: a level with the usual intensions for the expressions of logical form, and a level with a proof theoretic semantics for the types. Lexical semantics establishes a tight correspondence between logical forms and types (although it is possible on my account for the logical forms and types to come apart in certain situations). This correspondence will enable us to encode as relations between types analytical entailments between the logical forms to which they are linked, entailments that is, that are *a priori* and follow from the meanings of those expressions. The approach is thus Anti-Quinean (and also contra Fodor and Lepore (1998) and Lepore and Capellen (2004)). Nevertheless, I accommodate many of the ideas of Fodor and Lepore, insofar as the meanings of words contain the intensions traditionally associated with them. Traditional lexical semantics and philosophy however postulate too simple a view of predication and they ignore the important role of types and type-theoretic operations that determine how word meanings combine to produce logical forms.

Investigations into the nature of types and the argument for this two level theory will occupy much of the preliminary work on the theory of type driven predication in the next chapter. The development of the system of complex types and the two stage theory of semantics and its applications will occupy the bulk of the book. Besides the analyses of copredication and coercion, I will show how the system yields an analysis of relative predication and the genitive construction as illustrations of the scope of the theory. We'll see that types in the theory of predication are closely linked to metaphysical principles of individuation and counting. Thus, the types used to guide predication will be of a quite general nature. The system of types, however, involves more types than those just needed for checking predication. It provides a linguistic foundation for a theory of concepts and of internally available contents. I will argue that these are essential to analyse loose talk. Though topics like metaphor and poetic license outrun the scope of this book, I will tentatively offer an application of this extended system of types to these areas at the end of the book.

Chapter 2

Types and Lexical Meaning

The type restricted form of application and the typed lambda calculus are familiar to anyone who has worked in formal semantics. But many questions arise about the nature of types, their relations to formulas of logical form and the effect of rules of type shifting on logical form. In addition, the data I want to account for show us that predications are affected by information about elements other than the predicate and the argument involved in the predication. We need to look at these questions in detail.

2.1 Questions About Types

Let us first turn to examining the nature of these entities that are to guide predication, the types themselves. For one thing, we must decide whether our types are all atoms or whether they have structure in which “type constructors” together with other types are combined to yield complex types. This sort of question is familiar from logical languages and syntax where a certain set of atoms along with a set of constructors are assumed to generate complex formulas or syntactic constituents recursively. The traditional typed lambda calculus of the extensional fragment of Montague Grammar distinguishes only between the type of entities e and the type of truth values ᴛ as basic types and countenances only complex functional types that are recursively defined from these two primitive types. Thus, in that framework we have a ready made answer to our question: there is only one complex type constructor $\Rightarrow$; if α and β are types, then the type $\alpha \Rightarrow \beta$, the type which given an argument of type α produces an object of type β, is also a type.[1] However, we need to enrich the system of types considerably in order to deal with the data presented, and we will also need other type constructors. Thus, the following questions present themselves.

- What are the basic type constructors?
- What is a good theory of type constructors?
- Are the type constructors fully general or are there restrictions as to what combines with what?
- How do we understand types? What sort of entities are they?

Our strongly typed lexicon also countenances operations on types such as type shifting. Type shifting operations from objects into eventualities of the sort that underly the coercion phenomena noted with aspectual verbs are one example, but there are many others that we shall investigate through the course of this book. The list includes: nominalization (the operation from higher types into subtypes of e, e.g. *is square* ⟶ *the property of being square*), grinding, the operation from type e, or a subtype of e, into a quantifier type to handle coordinations like *John and three students went to the party*, the operation from types of transitive verbs as types of first order relations into higher types, the operation from type e or a subtype of e into a group type (as in *Bob, Ted, Carol and Alice lifted a piano*), and the transformation from mass/count noun complements into activity/telic eventuality types (*drink wine* vs. *drink three glasses of wine*). We need a framework within which to talk about such operations on types, which are in effect type shifting operations. We also need to have some analysis of these transformations, as well as a catalogue of which sorts of transformations there are.

2.2 Are Types Extensional or Intensional?

To begin our investigation of the basic nature of types, let's turn to the classical theory of functional types in the lambda calculus. This is the theory used in extensional fragments of Montague Grammar. We start with two basic types, the type of entities e and the type of truth values T and then close the collection of types under the recursive rule that if a and b are types, then so is $a \Rightarrow b$. In standard Montague

[1] The notation → for the functional type constructor is standard. But I want to distinguish this functional type constructor from its related cousin, the implication constructor for logical forms, →. So I use the following slightly nonstandard ⇒

Grammar, we convert these extensional types into intensional types as follows: if a is an extensional type, then $s \Rightarrow a$ is its intensional correlate, where s is the type of worlds or more generally indices of evaluation. This theory has an extensional set theoretic model, where the primitive types are identified by their *inhabitants*, the set of objects of that type relative to some domain of interpretation, while the set of functional types over types a and b is modelled as the function space or set of all functions, $a \rightarrow b$, from a into b. When Montague developed his theory, these types served a logical purpose. Turing (19??) had shown that the untyped lambda calculus had a model in the set of computable functions, but the application of such a theory in formal semantics was logically problematic: when terms of the untyped lambda calculus include the standard truth functional operators essential to semantics, it is easy to form terms like $\lambda x \neg x[x]$, which is the property of not applying to oneself—the Russell Property. Using the standard rule of application to apply the Russell property to itself produces the following result, which is uninterpretable according to the rules of the interpretation of negation in classical logic:

$$(2.1) \qquad \lambda x \neg x[x][\lambda x \neg x[x]] = \neg(\lambda x \neg x[x][\lambda x \neg x[x]])$$

The standard theory of types avoids this problem, since the Russell Property does not have a consistent type in the typed lambda calculus. It is, in other words, not a well formed term. Thus, the typing of expressions had a logical purpose; it ensured that the theory was consistent if set theory was. The typed lambda calculus avoids paradox in a simple and pretty much cost free way in comparison to the attempts to do natural language semantics in a type free framework.[2] Since the work of Scott and Plotkin in the early seventies, we have abstract models of the type free lambda calculus, though they are more complex and have certain drawbacks for the purpose of studying natural language semantics. The models used by Scott require that the values of lambda terms be continuous functions in the sense that roughly one can compute their value in the limit given some long enough run of values. But it is precisely the operators of classical logic like $\neg$, $\forall$, or $\exists$ that fail to be continuous in the requisite sense. Furthermore, though the untyped lambda calculus has some very beautiful properties and appears to have advantages with certain self-referential predications,[3] there are compelling linguistic reasons to adopt a typed lambda calculus in constructing logical forms.

[2] For work on that see Turner 1989.

[3] Strongly functionally typed languages appear to have a problem with terms that can take themselves as arguments. Consider

The syntactic categories together with the types assigned to the individual terms in the lexical entries determine all the functional types. Thus, the construction of logical forms from lexical meanings can exploit syntax in determining which words combine with which and in which order. Without this tight connection between types and syntax, the problem of deriving a logical form from the meanings of terms would be much more complex and ignore many linguistic constraints on meaning that come from syntax.

Nevertheless, there are reasons to suspect that the set theoretic model of the theory of types isn't quite right. We need to distinguish between a type and the type's extension or inhabitants. Consider for example the relation between the type of physical properties (PHYSICAL-OBJECT ⇒ T) and the type of all properties *e* ⇒ T. It is intuitively obvious that the type of physical properties is a subtype of the type of all properties. But if we think of types as sets and stipulate the relation of subtyping to the relation of subset, then this intuitively obvious connection is lost, because the functions from physical objects to truth values is distinct from the domain of entities and so no element of the set of inhabitants of PHYSICAL-OBJECT ⇒ T is an element of *e* ⇒ T.[4]

Another reason we need to move away from the extensional conception of types is because sets are not a faithful representation of the basic types that are subtypes of *e*. Consider for example the set of fictional entities. Let us take seriously the idea that fictional objects are indeed fictional and so they don't exist; they are not part of the domain of interpretation. Further, fictional objects not

(2.2) a. being nice is nice.

b. it's bad to be bad.

Such examples seem to be cases where a property applies to itself; and indeed it is unclear what (functional) type to assign to the property if this is right. On the other hand, one might suppose that what is really going on is that the property applies to a nominalization of the property and that the nominalization is type theoretically significant. It transforms the type of its argument from some sort of higher order functional type into an abstract entity. Now one might argue that this won't save a typed theory of predication since there exists no 1-1 map from higher types into abstract entities; if one assumes that there are at least as many higher order properties as there are subsets of the basic domain, postulating such a 1-1 map would appear to violate Cantor's theorem. Indeed there is a cardinality problem if *all* properties have nominalizations. But why suppose that this is so with respect to natural language? There are only at most countably many *expressible* properties and so countably many nominalizations of those properties as abstract objects. At this point the cardinality problem disappears. Examples like (2.2) don't argue against a type driven notion of predication. I explore this issue further in the last chapter.

[4] We could reinstate the set theoretic interpretation by taking PHYSICAL-OBJECT ⇒ T and *e* ⇒ T to be sets of partial functions, but such a model is not terribly illuminating.

only don’t exist in the actual world; they don’t exist in possible worlds either—i.e. other ways in which the world could be. That’s precisely the meaning of what it is to be fictional. Then the type corresponding to fictional objects and the absurd type are intuitively different, yet nevertheless have the same extension or the same set of inhabitants, namely the empty set.

Why do I think there might be types corresponding to fictional entitites? Well, given that I think that lexical semantics ought to be concerned with predication, whether a term describes a fictional character or not certainly appears to make a difference as to how predications are understood. Within fiction, there is no question of checking or wondering whether the predication actually results in a literal truth. It is even quite controversial among philosophers who have written on fiction whether terms that appear to refer to fictional entities refer in fact to anything at all. On the other hand, fictional talk differs from metaphorical or loose talk; fictional talk is literal—the trees in *The Lord of the Rings* literally speak (see 2.3) whereas in metaphorical talk the predications aren’t to be taken at face value.

(2.3) Look! the trees are speaking (*Lord of the Rings*)

To make sense of this difference in predicational behavior, there should be a distinguished type of fictional objects. And it should be distinct from the absurd type no matter what the circumstances of the actual world are. This leads me to adopt the thesis that types are neither to be identified with their actual inhabitants (extensions) nor even their possible inhabitants (standard semantic intensions).[5]

So it looks like types are intensional entities, though they are not intensions in the standard semantic way of understanding this term. Thus, there remains the question of where they fit into an ontology of abstract entitites. We need to think about the relations between the following sorts of abstract entities:

- types
- concepts
- properties

Properties have at least one relatively well understood analysis that is standard in formal semantics and pragmatics. They are the denotations of predicates and are typically modelled as, or identified with, intensions, where intensions are functions from indices—possible worlds or sequences consisting of a world, time,

[5] Reinhard Muskens in Muskens (2007) also argues for such intensional construals of types.

context and other appropriate elements—to an extension. (In the case of a 1 -place property, the extension is just a set of individuals.) But *modulo* a certain understanding of fictional objects, we have established that types aren't to be identified with extensions, intensions, or sets thereof. Furthermore, properties are typically understood to be mind independent entities. But types, given their role in guiding predication, are part of the conceptual apparatus necessary for linguistic understanding; they appear to be mind dependent entities. So this leads us to the hypothesis that types are concepts.

If properties are, as many linguists agree, the denotations of predicates, concepts have always mystified linguists. For philosophers concepts are more familiar, albeit just as mysterious. I can understand them at least along the following lines. Concepts come at different levels and granularities. There are concepts of what it is to be a property, what it is to be an individual. There are also much more specific concepts that people may have; one can have a concept of red, as well as a concept of Ségolène Royale or of Hillary Clinton. Concepts are the internal, mind dependent reflection of mind independent properties and individuals they are concepts of, and they have their own internal semantics which has to "track" in some way the content of the properties and individuals they are concepts of.[6] It is in virtue of such tracking that a concept is a concept of some object or property. And it is in virtue of the fact that your concept of red and my concept of red have the same internal semantics that we can say that we can be said to have the same concept of red.[7] As internal reflections of properties and individuals, concepts are

[6] Although making this notion of "tracking" precise would probably lead to another book on a very different topic, I can say a few, speculative words here. I understand this notion quite literally in terms of how the rules for the application of the concept in the conceptual system function. One can think of these along the lines of natural deduction rules; a concept has certain "introduction rules" and certain "elimination" or "exploitation" rules. There can also be more complex combination rules, saying how one concept interacts with others. I shall develop such an internal semantics for types through this book that has, I believe, applications to this more general philosophical theme. Such rules supply the internal semantics of the concept. Such rules track the content of the objects and properties the concepts are about in situations like the following: a concept RED of the color red is introduced as holding of something at a particular location *l* in the conceiver's visual field when the conceiver is perceptually aware of something at *l* that is in fact red in color. They fail to track the content of the objects and properties the concepts are about in situations like the following: a concept RED of the color red is introduced as holding of something at a particular location *l* in the conceiver's visual field when the conceiver is perceptually aware of something at *l* that is in fact green in color. The concept RED is exploited to form more complex concepts like RED PEN, RED APPLE and so forth.

[7] One could also assume concepts and types to be mind independent objects. Frege held, and George Bealer holds, such a view. I don't see why such mind independent entities would be

not identical with mind independent properties or individuals but they are associated with them via the tracking. In this types resemble concepts; they are tied via the expressions they type to properties and real world entities, but they are not identical to properties or real world entities, nor to sets thereof. They are part of our conceptual apparatus used to guide predication. Concepts also have a hierarchical structure, just as types do. These resemblances lead me to hypothesize that types are concepts; the set of types is a subset of the set of concepts.

Concepts at least as traditionally used in philosophy are understood to be the constituents of thoughts. Many philosophers (see Peacocke 1992) have used concepts to account for informativeness and Frege style puzzles about the substitution of coreferential terms, understanding concepts to be something like Fregean senses. On a standard Fregean view, senses are not only components of thoughts but also components of propositions or what are expressed by complete sentences; thoughts and propositions are one and the same sort of mind independent entity. I will take a quite different view. A proposition is what is expressed by a sentence or a discourse. A proposition is the result of the compositional interpretation of logical forms for the words that make up the sentence or discourse. But the compositional interpretation of a logical form results for me, as for most semanticists and linguists, in an intension—a function from indices to truth values. So given what I have said already about concepts, concepts cannot be the constituents of propositions; in fact, on the standard semantic conception of propositions, propositions don't have "constituents" except in a set theoretic sense—and these would be objects, sets of objects, worlds and other indices. Now one might take intensions to be simply formal stand-ins for what propositions "really are". But even then, if sentences are typically about mind independent objects and the properties and relations these objects stand in, then "real propositions" will still not contain concepts of the sort I have in mind, concepts with an internal semantics. For although these may track the mind independent objects and the properties the concepts are about, they cannot determine reference to these mind independent entities for well known externalist reasons.[8]

needed, over and above the necessary internal reflections of mind independent properties and objects. If the internal semantics of the concepts is robust enough, then we can make sense of multiple people grasping the same concept without appealing to mind independent concepts: when this happens these agents have the same internal semantics for their concepts and track the same objects and properties.

[8] These sorts of reasons have been forcefully driven home by philosophers like Kripke, Putnam and Burge, as well as by a host of their commentators.

If concepts, and *a fortiori* types, are not constituents of propositions in the way most semanticists think of propositions, they can nevertheless compose together. In fact they are designed to do so, because they are designed to check predications in which one term is applied to another. Making the (linguistic) types a subset of the set of concepts allows us to use the logical framework of types to explore concepts. Types associated with properties are functions from one type into another. Thus, we can compose types together to give us types associated with propositions or semantic intensions. The type corresponding to a proposition on our hypothesis is a thought. This hypothesis together with the argument in the preceding paragraph leads me to distinguish thoughts from propositions. Concepts, and *a fortiori* types, compose together to form thoughts, not propositions. Thus, although concepts and types are not components of propositions, they are the components of thoughts.[9]

Once we distinguish between subtypes of the type *e* of entities non-extensionally, as we must if we are to countenance a type like fictional-object as distinct from the absurd type $\bot$, we must think of the complex types in a more fine-grained way as well. Just as we have different subtypes of the type *e*, we might also have different subtypes of t, which I'll continue to interpret as the type of propositions. Note, however that t is distinct from the set of propositions that inhabit it; just as with other types, I do not identify the type with the set of its inhabitants. If there are as many types as there are words with distinct meanings in the language, then the hypothesis that types are concepts and compose together to yield thoughts gains in plausibility.

Here is a simple illustration of how to think of the finegrained subtypes of t. One constructs the proposition that Red(a), by applying λxRed(x) to some individual term a, which, say, denotes the apple in front of me. Red(a) picks out a proposition, which I'll take to be a set of worlds or an appropriate dynamic meaning. Now what about the concept or type associated with that proposition? red is a concept, and it is a subtype of a linguistic type, the functional type from physical objects to truth values. Let's assume that a has the concept apple, which is a subtype of physical-object. Since red is a subtype of a functional type, we can apply it to a type of physical object to get the type:

[9] The talk of constituents here must be taken with a little care. It is true that concepts compose together to form thoughts, but that does not necessarily mean that the finished product will actually contain those concepts. Nevertheless, because I shall identify thought contents at least in part with the derivation of the thought from its constituents through composition, that's pretty close to constituency.

- RED(APPLE)

This picks out a distinct subtype of *t*, one in which the type RED composes with something of type APPLE. Suppose I have an indvidual concept of the apple in front of me—call it APPLE$_1$. Then the type RED(APPLE$_1$) corresponds to the complex concept or thought in which I predicate the property of being red to the apple in front of me. Its inhabitants are propositions, and thus it is a type of propositions. This leaves us with a very finegrained conception of propositional types, and we can also think now of propositional types as having *parameters* that can be replaced by variables. I'll come back to this issue later.

Nevertheless, we must be careful in assuming that all of these finely individuated types play the same role in the linguistic system. There is good evidence that finely individuated types do not play a role in the type checking relevant to a theory of predication, which is the principal role of types in lexical semantics. Typically only types that involve different individuation or counting conditions are relevant to the type checking involved in a type driven theory of predication as these examples show:[10]

(2.4) a. Tigers are animals.

b. Tigers are robots.

c. #Tigers are financial institutions.

d. #Tigers are ZF sets.

One might hypothesize that types at the upper end of the type hierarchy are relevant to type checking. ANIMAL is a type that would seem to be at the upper level of the type hierarchy (Dölling 2000). ROBOT is a subtype of ARTIFACT, which is also at the upper level. And most type hierarchies would take ANIMAL and ARTIFACT to be incompatible types. This assumption would predict that (2.4b) (2.4c) and (2.4d) are all equally semantically anomalous. But they are not, even though they are all false, indeed necessarily false. One way to distinguish between necessary falsity and semantic anomaly besides intuitions about truth value failure or unaccommodatable presuppositions is to think about whether a competent speaker of English person could believe or entertain (2.4b,c,d) or try to find out whether they are true. Given the history of mathematics, it's pretty clear that many people have believed necessarily false things, and it's pretty clear that competent speakers of a language do not believe what we express with a term involving a semantically anomalous

[10] Thanks to Dan Korman for the first two examples.

predication. According to intuitions, a competent speaker can certainly entertain or even believe that tigers are robots; he or she can also go about trying to figure this out (e.g. by dissecting a tiger). It is much harder to accept the possibility, or even to make sense of, a competent speaker's believing or even entertaining that tigers are financial institutions let alone ZF style sets.

The clash between the type demands of the predicate and the type of its argument in these examples seems to have to do with rather deep metaphysical principles like individuation and counting conditions. Sets, especially of the ZF kind, have simple individuation criteria and are all built out of a single object, the null set, on the standard, cumulative conception. Tigers share nothing with ZF sets with respect to individuation conditions. Financial institutions are more abstract than standard physical objects— they are relational structures that depend on various physical individuals but not any particular ones (the employees of the institution can change, so can its physical locations while the financial insititution continues). Once again tigers are not counted or individuated in anything like the way that financial institutions are. It is these general individuation conditions that are for the most part relevant to the type checking in predication. The type checking in predication is sensitive only to those types that have different individuation and counting conditions, in the sense that a semantic anomaly will result only when the type requirements of the predicate and argument give rise to incompatible individuation conditions.

The types relevant to type checking in predication are somewhat diverse. They include very general concepts—like PHYSICAL OBJECT, INFORMATIONAL OBJECT or ABSTRACT OBJECT, MASS, COUNT, PLURALITY SINGULARITY, and EVENTUALITY. On the other hand, there are relatively finegrained distinctions that predications, in particular those involving adverbial modification, are sensitive to: STATE, ACTIVITY, ACCOMPLISHMENT, ACHIEVEMENT, and so on. Robots and animals are close enough in terms of identity and counting conditions that we can make sense of (2.4b). On the other hand, seemingly closely related types like ACTIVITY and ACCOMPLISHMENT lead to a type clash because adverbials like *in an hour* and *for an hour* are sensitive to the defining and differentiating features of these types of eventualities. It's hard to find a predicate similarly sensitive to the defining and differentiating features of tigers and robots. (2.4c,d) are much harder to make sense of without a big background story. And, one can argue, a big enough background story may lead to a type coercion of the sort we've seen already. As we shall see, individuation and counting conditions will play an important role when we come to the analysis of copredication and coercion. The types relevant to type checking in predication thus form a restricted subset of the general set of types, which themselves are a subset of the set of concepts.

Most predicates are insensitive to the finegrained types of their arguments. For instance let us suppose that ‘Hesperus’ and ‘Phosphorus’, two names for the planet Venus, give rise to two different concepts. For the purposes of type checking in predication, the distinctness of these concepts doesn’t matter. What matters is the subtype $\sqsubseteq$ relation on types that can be exploited for checking predication.

(2.5) HESPERUS, PHOSPHORUS $\sqsubseteq$ HEAVENLY-BODY $\sqsubseteq$ PHYSICAL-OBJECT $\sqsubseteq e$.

That is, if we have a predicate P that requires the relevant argument to be occupied by something of type PHYSICAL-OBJECT, then the predication Pt should succeed if t has the type HESPERUS; P should apply to any subtype of PHYSICAL-OBJECT; and since HESPERUS $\sqsubseteq$ PHYSICAL-OBJECT, we know the predication will succeed. In fact, it’s quite hard to see how type constraints on predicates would ever distinguish predicational contexts in which ‘Hesperus’ was felicitous and ‘Phosphorus’ was not. This is not to say that there are not other ways of distinguishing between the types HESPERUS and PHOSPHORUS, as we shall see in chapter 9.

While fine-grained differences in our rich system of types do not play much of a role in singling out the semantically anomalous predications, these differences come into play when we turn to the contextual effects in many predications. Fine grained concepts are linguistically relevant in ordinary predication for guiding how the predication is to be understood. For instance, as we are interested in coercion, we will want to distinguish:

(2.6) a. RED(APPLE)

b. RED(SHIRT)

c. RED(PEN)

d. RED(MEAT)

Each of the predications in (2.6) are understood to be distinct. When we predicate a color of an apple, we predicate it typically of the skin. We can call an apple truly red even if most of it, its flesh, is white. On the other hand, this won’t do for *red shirt*. A shirt is red only if the entire shirt, or very nearly so, is red. We may call a pen red if it writes in red ink; the color of the visible surface of the pen may not be relevant at all for that predication to succeed. Finally, *red meat* refers to a kind of meat e.g., beef, lamb and the like, which may not be at all red on the surface, say, when it shows up on the dinner plate. While the adjective *red* like other color words is typically analysed as an intersective adjective (according to which the red X is both an X and red), it appears to become subsective (according to which the red X is an X but not necessarily red) when combined with *meat* or *pen*. That is, the following sentences are not necessarily false:

(2.7) The red pen is not red.

(2.8) That piece of red meat is not red.

Even though there is no type incompatibility here between the type of the color adjective and the type of the common noun it modifies, there is a subtle shift in meaning and semantic behavior of the adjective noun combination. This shift arises from predication of one expression to another, each of a particular type, and it is such shifts that a type driven theory of predication is supposed to explain. These subtle shifts in the resulting predication due to type combination and coercion make it plausible that in principle each word root, even each word, could give rise to a distinct concept that can play a role in the type system.

Finegrained differences in types affect many predications.

(2.9) a. John treated Mary to dinner.

b. John treated Sam for cancer.

c. John treated the cancer.

(2.10) a. John swept (shoveled, wiped, . . .) the closet (room, walkway, kitchen, fireplace, floor, counter . . .)

b. John swept (shoveled, wiped, . . .) the dirt (debris, manure, sand, slush, litter, shavings, cinders, dust . . .)

(2.11) a. John weeded (mulched, hoed, . . .) the garden (lawn, area, tomatoes, peas, plants . . .)

b. John weeded (hoed) the weeds.

(2.12) a. John shouted (whispered, whistled, whined, . . .).

b. John shouted (whispered, whistled, whined, . . .) a warning.

c. John shouted (whispered, whistled, whined, . . .) at the animal.

Many of these predication alternations have been noticed in the literature (Levin (1993)) and a variety of analyses have been proposed. But even to state the generalizations as to how these predications behave, we must recognize distinct types accorded to arguments of these transitive verbs. Predicating the verb *treat* of a living entity such as a person, animal or even a plant means something quite different from when the verb is predicated of a direct object that denotes a disease. And the type of the object of the prepositional phrase modifying the verb also will make a difference to what the predication ends up expressing.[11] Similarly, predications involving *sweep* have a different content depending on whether the direct object is a location, a place or a surface (we can assimilate all of these here to one type LOCATION for the purpose of this alternation), or whether the direct object is a portion of matter. Another illustration of how finer grained types make a difference to predication is the alternation in (2.11). *Weed* and *hoe* apply felicitously to locations and to certain desirable plants (which also could perhaps be coerced into locations) whereas to weed or to hoe weeds is to get rid of them.

We can easily see that the predications result in different meanings—in fact so different that they don't license copredication or ellipsis:

(2.14) a. John swept the kitchen and Mary the entryway.

b. #John swept the kitchen and Mary the leaves.

c. # John swept the kitchen and the dust.

(2.15) John hoed the tomato patch and Mary the weeds.

Another indication that finegrained differences in types make a difference to logical form is that certain constructions are completely out when *sweep* takes a portion of matter as a direct object:

(2.16) a. John swept the floor clean.

b. #John swept the dust clean.

There are fine grained differences in type requirements of these verbs. You can wipe the water away and even shovel water or sweep water, but you can't rake

[11] A famous alternation showing a striking interaction between prepositions and verbs is the following (again for a lengthy bibliography and summary of the research on this alternation see Levin (1993).

(2.13) a. John loaded hay on the wagon.

b. John sprayed paint on the wall.

c. John loaded the wagon with hay.

d. John sprayed the wall with paint.

water (at least with a rake). The verb *rake* has rather special type requirements on its direct object. Similarly verbs like *weed, hoe, mulch* have special type requirements. You can't weed or hoe certain types of locations—bodies of water for instance. Locations that can be weeded have to have dirt or soil in them. This is not a matter of world knowledge but a matter of grammar, broadly construed: one can perfectly well imagine someone cleaning a lake of algae or of water plants, but we don't call that *weeding*.

Let me summarize the main points of the discussion so far. I have hypothesized that types are fine grained, intensional entities and I have tentatively identified them with concepts. I have also spoken of an internal semantics that types and concepts receive in terms of their role within the linguistic and ultimately extra-linguistic cognitive system. The linguistic types have an internal semantics that is given at least in part in terms of the rules by which they combine with other types.[12] For fine-grained differences between the types, we will look to associated traits with the types. We may even think of each type relative to an interpretation of it as a program or proof—giving conditions of its application. We will look later at these rules in detail. But in effect we are specifying the semantics of types and concepts via proof rules in the extended lambda calculus that I shall develop. This is in keeping with a key idea of the theory of types that I will talk about much more below: the Curry Howard correspondence between types and proofs. If there are concepts other than the linguistic types that I focus on here, I speculate that they have the same sort of internal content, specified by rules of combination, introduction and elimination. We will end up identifying the concepts or types corresponding to propositions with the "proofs" of their construction in our lambda calculus.

2.3 Types and Logical Forms

If we interpret types as concepts, then we need to reevaluate how types relate to traditional formulas of standard semantics. But first let me say a word about logical forms and how they will be understood in this book.

Truth conditional semantics and dynamic semantics standardly use a language of logical form to state the truth conditions or dynamic update conditions of natural language discourse. As research in the semantics and pragmatics of natural language has come to concentrate on the analysis of expressions whose content is

[12] For similar ideas see Fernando (2004)'s automata theoretic analysis of verb meanings, which amounts to a constructive theory for their types.

specified by the discourse context, the use of a level of logical form has become well neigh inescapable. For instance, suppose we have to give the truth conditions of a sentence with multiple quantifiers.

(2.17) Every nurse examined a patient.

This sentence in isolation has two readings: one in which the universal quantifier introduced by *every nurse* takes wide scope over the existential quantifier introduced by *a patient*, and one in which the existential quantifier has wide scope over the universal quantifier. However, discourse contexts can often determine which of these readings is relevant to the truth conditions of the discourse as a whole. If for instance, (2.17) is followed by a sentence like *He was very ill*, it will be the second of the two readings that is intended, since only that reading provides an antecedent for the pronoun (assuming of course that *he* does not have a deictic use). In addition, there are many anaphoric phenomena in natural language—for instance, pronouns, tenses, presuppositions—for which a determinate semantics cannot be supplied independently of the discourse context. To handle such constructions, researchers have resorted to underspecified logical forms, logical forms with “holes” in them (Reyle 1993, Bos 1999), to express the contribution of such elements. These holes may encode structural ambiguities like those given by the quantifiers in (2.17. But they can also represent lexical underspecification. Take for instance a pronoun like *he*. Its meaning is supplied by the antecedent it is linked with in a particular discourse context. One cannot specify a lambda term conveying the meaning of such an expression without resorting to underspecification. However, with underspecification, specifying a lambda term is relatively straightforward; in Asher (1993)’s language for underspecification, *he* has the following lambda term— $\lambda P \exists x(P(x) \wedge x = ?)$, where ? marks the ”hole” to be filled in by a term provided by the pronoun’s antecedent.[13] Underspecification at the level of logical form will also play an important role in this book: we’ve already seen that that some coercions of lexical meaning are similar to anaphoric phenomena in that they are sensitive to discourse context. I will resort to underspecified logical forms to analyze these as well.

One could perhaps do without logical forms and have a directly denotational, context sensitive semantics, but it would have to be very complex. Meanings would have to be functions from discourse contexts to intensions, and, as far as I know, nobody knows how to specify discourse structures of the requisite sort

[13] See Pogodalla (2004) for a completely typed version of this expression in which holes are introduced as a special type.

without resorting to representations. Thus, for the rest of the book, I will take logical forms to be a basic component of semantics. A theory of predication must yield such logical forms for clauses; and a theory of discourse structure must yield logical forms for discourses.

In Montague grammar the expressions of the language of logical form are typed. The types have a simple set theoretic model, while the logical forms have their own intensional semantics. The set-theoretic model of types is much more impoverished than the intended model for the logical forms themselves. The types of the extensional typed lambda calculus convey little content; they do not, for instance, distinguish between any entities or between the truth values. Even in an intensional setting they do not distinguish between propositions or between different properties. In all classical semantic systems, types need to be distinguished from formulas, since types do not convey enough information to provide truth conditions.

On the conception of types that I have proposed, the need for a level of semantics independent from the level of types is less obvious. The structure of types is now very rich. Once we have functions from types to types and follow the principle that each lexical root gives rise to its own associated type, it is as rich as the structure of the language of logical forms which supply the truth conditions for clauses. We can even distinguish between different conceptualizations of the same physical object or of the same property. The language of types becomes in principle at least as rich as that of the logical forms they are supposed to help construct. Why not eliminate such a redundancy? If types are concepts, can't we identify types with logical forms and identify the internal semantics of types with the semantics for natural language sentences and discourses? This is the line taken by Martin Löf's intuitionistic Type Theory (Martin-Löf (1980), Ranta (2004)). The idea is a large scale development of an idea proposed by Howard (1980), the so called 'Curry-Howard isomorphism'. The idea is that each lambda term corresponds to a type which encodes a proof. These proofs determine the meaning of the term and its natural language correlate. For example, the lambda term corresponding to the English expression *every man* looks something like this:

- $\lambda P \forall x(\mathrm{Man}(x) \rightarrow P(x))$

The type of this term is $(e \Rightarrow \textsc{prop}) \Rightarrow \textsc{prop}$, where PROP is the type of propositions and $\Rightarrow$ is the functional type constructor. The interpretation, intuitionistically speaking, of the term itself is one which is a function from proofs to proofs; given a property P or a proof from individuals to truth values, the lambda term denotes a proof that every man has that property P. To apply this term to the lambda term $\lambda x\,\mathrm{sleep}(x)$ successfully is to produce a proof procedure for proving that every man sleeps. To assert that every man sleeps is to assert that one has a proof that every man sleeps. The theory also gives a nice account for semantically anomalous sentences: for such sentences a proof procedure cannot be constructed, and the sentence literally has no meaning.

In Type Theory, types, proofs and propositions are identified. The meanings for the types coincide with the meanings of the logical forms themselves. In fact one can simply obliterate the distinction between formulas of logical form and formulas for types. This conception is opposed to the one provided by Montague and classical semantics. But the Type Theoretic view proposes some intriguing ideas; predication fails when we can't put together the proof for the term as a whole—failures of predication are just failures of meaning. It also provides what computer scientists call an "internal semantics" for the lambda calculus, as well as its extension to higher order logic. That is, the semantics for the system is given in terms of the proof theoretic apparatus of the logical system, not in terms of some externally given values provided by an independent model. The notion of proof can be suitably generalized to an abstract mathematical sort of construction of the sort that intuitionists have worked on for many years in the field known as Topos theory.

On the other hand, this conception of the meaning of formulas runs into severe difficulties when we think about basic referential expressions: indexicals, proper names, demonstratives. The content of these terms, intuitively, has to do with the individuals they denote, not some proof object. The simplest statements about the external world like *you're hungry* expose the soft underbelly of intuitionist or proof theoretic approaches to meaning.

One can sharpen this critique. My reasons for distinguishing two levels are philosophical, conceptual and methodological. If types are concepts, then these are to be distinguished from the logical forms themselves that describe features of the real world, properties and individuals. The role of type formalisms and logical forms are different. Types and their adjustment are the heart of a theory of predication, while the logical forms are the soul of a theory of word meaning. Recall that the construction of logical form depends on type checking, and that it is the clash of general, high level types that lies behind, in my view, failures of predication. Thus, types are responsible for our intuitions about semantic well-formedness and, I will argue below, for analytical entailments. But types are only a partial guide to the semantic value of a word and certainly don't suffice to exhaust their contents; as we know from the many externalist arguments in the philosophy of mind and the philosophy of language, concepts, our internal mental reflection of real world properties and things don't suffice to determine denotations or intensions. The arguments of Kripke, Putnam, Kaplan and Burge show that our concepts associated with names of individuals and natural kinds do not suffice to determine the extensions or intensions of these expressions. If one looks to the behavior of such terms in modal contexts, there is compelling evidence that their meanings are not in general determined by "what is in the head" of a competent speaker of the language. Types are, on my conception, "in the head" if anything is. They are concepts and the core of a speaker's semantic competence. It follows that types cannot by themselves play the role of logical forms whose semantics incorporates the externalist elements isolated by Kripke, Putnam and others; nor can their semantics substitute for the standard denotational semantics of logical forms that underlie the philosophical arguments for externalism.

To make this argument more concrete, let's consider a typical Twin Earth scenario, familiar from the externalist literature cited above. Oscar on Earth and his twin 'Twin Oscar' on Twin Earth speak syntatically identical languages and are type identical down to their molecular constitution. In keeping with general physicalist principles then, they have the same internal make up, the same thoughts, the same conceptual system. In particular their linguistic judgements about semantic well formedness will be the same. Thus, when Oscar and Twin Oscar each interpret the strings 'water is wet' and 'water is a tree', they assign the same syntactic form and the same semantic types to each expression; for the first string they will each construct a coherent logical form using the tools of the theory of predication, while for the second they will not. But whereas they marshal the same type system and conceptual resources when dealing with their languages, the languages of Oscar and Twin Oscar are different—they have a different semantics. On Earth *water* picks out the kind $H_2 0$, or real water, whereas on Twin Earth the string 'water' has a different semantics—it denotes a chemical compound which, so the story goes, is XYZ or "twin water". Suppose that the terms 'XYZ' and '$H_2 0$' are both expressions of English and Twin Earth English and that XYZ and $H_2 0$ are indistinguishable to both Oscar and Twin Oscar and even indistinguishable for their respective linguistic communities. Nevertheless, they are different substances and so the string 'water' in English and Twin English has a different meaning and makes dramatically different contributions to truth conditions in sentences like:

(2.18) a. Water is $H_2 0$.

b. Water is XYZ.

Such Twin Earth scenarios are well established in the philosophical literature, and people's intuitions about them are relatively robust. A theory that identified the meanings of terms with the internal semantics of types has no means for accounting for these intuitions.

Furthermore, it is very unclear how types alone can play the role of logical forms when it comes to operations like the treatment of presuppositions or the integration of new information into discourse. Semanticists like Heim (1983), Van der Sandt (1992), Beaver (1997), among others have developed a sophisticated treatment distinguishing presupposed content from "proferred" content (the content that ends up, for instance, in the assertion). Presupposed content has to be integrated into the discourse context in different ways from proferred content, and in ways that make essential use of logical forms as semantic representations, as we'll see in the last chapter. Additionally, theories of discourse structure (Asher (1993), Asher and Lascarides (2003)) make essential use of logical forms as representational structures to build interpretations of discourse. A monostratal theory without logical forms would have to complicate the type theory considerably to be able to handle the complex operations involved in the treatments of presupposition and discourse update. We would in effect be saddled with all the problems of a semantic theory that attempts to do away with logical forms but that also attempts to do justice to the context sensitivity of linguistic meaning. A theory that distinguishes logical forms and types allows us to keep the system of types simple and dedicated simply to matters of predication.

I have given conceptual and philosophical reasons in favor of distinguishing a level of logical forms from the level of types in a theory of predication and lexical meaning. But one might think about simplifying the picture in a different way—by getting rid of types altogether. A different monostratal proposal suggests itself besides the Type Theoretic one—one which I'll call the Sortal Theory. The Sortal Theory claims that type information is really just a part of the language of logical forms that is singled out without justification. Type information is really just information about sorts. Semantic ill formedness is the result of using the semantic and proof theoretic apparatus of the language of logical forms to derive a contradiction based on meaning postulates about sorts.

At first glance, the distinction between the Sortal Theory and a typed theory of logical forms looks to be one of notational variance. The typed lambda term of the bistratal theory I propose in (2.19a) has the very similar looking Sortal Theory analogue in (2.19b) (ϕ_α is the object level formula associated with the type α).

(2.19) a. $\lambda x \psi(x), \quad x \colon \alpha$

b. $\lambda x(\psi(x) \wedge \phi_\alpha(x))$

But here appearances deceive. The typed lambda term of the bistratal theory clearly separates the presuppositions of felicitous predication with the term from the proferred content of the term. The presuppositions are encoded in the type assignment, and it is the phenomenon of presupposition failure that accounts for the lack of semantic well-formedness. The Sortal Theory has no satisfactory theory of semantic well-formedness. There aren't any syntactically well-formed sentences that are semantically ill formed; those sentences that are predicted to be ill formed because of type clashes on the typed view are just false on the Sortal View. Matters are no better with coercion. Recall that for the typed view, it is the mechanisms of presupposition accommodation that are supposed to account for the possibility of coercion. Coercion on the Sortal Theory monostratal view is just a matter of adjusting what was said based on charity, and it's no longer clear what is to be adjusted. Finally, there are technical problems with the sortal theory, as now we are in an untyped lambda calculus.

Of course those who don't believe in a level of lexical meaning separate from general world knowledge, knowledge of extra linguistic context in the widest sense, will be attracted to the Sortal Theory. Quineans or those philosophers with very strong contextualist leanings will think that the Sortal Theory makes entirely the right identification between coercion and belief revision. But then in the absence of any determinate context, the prediction of the Sortal Theory is that any sort of revision to a semantically anomalous predication is possible. But that isn't the case, as we've already seen. This counts, to my mind, as strong evidence against the Sortal Theory.

Types can and should play a role in semantics. Types and the partial order of subtyping on them permit linguistic generalizations about word meaning. More importantly, types serve a computational and theoretical need in lexical semantics and in the theory of predication. They simplify considerably the task of checking whether a syntactic sentence is semantically well formed or not; the property of semantic well-formedness for a sentence corresponds to the property of producing a lambda free logical form for it, and in a typed system with linear types like the one I will propose this property is not only decidable but polynomial in complexity.[14] For untyped systems, it is much less clear how to characterize semantic well-formedness, and in general the property of producing a lambda free logical form or *normalization* is in general not decidable. In fact it's hard to make sense of predication in the traditional sense—of a property to an object—in the type free framework, since the basic distinction between objects and properties doesn't

[14] One wrinkle to this property concerns nominalisation. I will deal with this complication in chapter six.

exist there. Types represent enough content to guide predication and the construction of logical form in the linguistic system. They are integral to the construction of logical form, which is an *internal* matter, something that speakers do “in their heads.” We need *both* types and logical form, but they need to be distinguished. There is a place both for types and for logical forms in semantics, especially in the theory of predication and of lexical meaning.

A much larger issue is how this separation reflects a view of thought in general, not just linguistically expressed thought. Many people have followed the view in the philosophy of mind (see for instance Burge (1977) Tye ??, Dretske (1981) , Fodor ??) that the contents of thoughts like the contents of many expressions in natural language are often determined by external factors, but it is hard to deny that there is an internal component to thoughts as well. How to demarcate these two realms and how to determine the internal component of thought is a very tricky matter. Perhaps such a project follows the outlines of the two stage theory of composition that I will detail below.

Types offer an interesting approach to problems that motivate the Fregean picture of language by developing a two factor theory of content in a rigorous way. An internal semantics for the types will allow us to construct an ”internal semantics” for language, which could be the internal reflection of externalist, model theoretic conceptions of content. That is, we could use a proof theoretic semantics for concepts and thoughts which is linked through the logical forms to a more externalist semantics. We have seen that we might want the meaning of the type HESPERUS to come apart from the meaning of the name *Hesperus*. In this way we can in principle make sense of the informativeness of certain true identity statements as well as acknowledge their necessary truth. Indeed having two semantics associated with formulas (one albeit indirectly) gives us the tools to make sense of puzzles about belief as well as paradoxes of informativeness.[15] We don’t want to collapse logical forms and formulas for types together or to identify types with propositions expressed by natural language clauses. We want to have something like a Fregean picture where two distinct types or concepts may correspond to the same referential meaning or intension. Unlike Frege’s senses, however, types and concepts do not determine the referential or standard, intensional meaning of terms. At best there is a homomorphism from concepts to sets of intensions that preserve the structures relevant to predication—i.e. the relations between types.

In sum, I think there are strong philosophical and conceptual reasons for pursuing a two factor theory of lexical meaning—involving types with an internalist

[15] For the beginnings of such an exploration, see Asher (1986, 1987, 1989).

semantics and logical forms with an externalist semantics. There also methodological reasons for pursuing a two factor theory. A monostratal theory of composition of either the sortal or type theoretic varieties will be more complex than a two stage theory in that it will put together both the tasks of type checking and type shifting for phenomena like coercion with the complex context effects on logical form that we know from dynamic semantics. Type Theory does address some of these effects (for instance there is a very nice treatment of donkey sentences in Type Theory), but clearly many problems with the composition of logical form, notably the case of presuppositions, have not been addressed in that theory and appear difficult to integrate within that monostratal account. Issues like copredication and coercion for Type Theory also appear to be difficult to handle solely at the level of types. Type Theory must analyse all of these phenomena at the level of types, since types, proofs and logical forms are all conflated. This leads to the introduction of complex types whose analysis is very problematic unless one in fact uses a two level theory. With each one of these phenomena that are problematic for a theory of lexical meaning and predication—copredication, coercion, real ambiguity or homonomy, presupposition and proferred lexical content, loose and vague predication— a theory that distinguishes logical forms and types has several options for analysing such phenomena. None of this shows that types and logical forms *can't* be conflated. But these considerations do show that there are practical and theoretical costs to making the conflation. In what follows I will pursue a bistratal theory distinguishing between types and logical forms.

Much of this essay will be devoted to a close examination of the interactions between the level of types and the level of logical form. For, as I'll argue below when I look in detail at models of complex types, it is precisely in the rules that determine how type shifts from complex types to constituent types affect logical forms that various complex types will distinguish themselves. In other words, a division between the level of types and the level of logical forms allows us to get at the right semantics for complex types, since the semantics for these types is in part determined by what they do to logical form.

2.4 The Question of Lexically Driven Inference

The view of types as concepts that I've just sketched entails a very rich set of atomic types, one for each word root. These give us a tool for exploring lexical inference or analytical entailment. The issues of lexically based inference and analytical entailment are murky ones, and one can study the problems of constructing logical forms and the phenomena concerning predication that motivate this book without staking out a position on them. The data concerning lexically driven inferences are also controversial. Fodor and Lepore (1998), for instance, question the robustness or legitimacy of the data as really part of semantics. Among most working linguists and lexicographers, however, the intuitions are strong if not entirely systematic that there are many analytical entailments. The question I want to pursue briefly here is whether, and if so how, the type mechanisms we have devised for the construction of logical form give us purchase on the often debated but little resolved issues of lexical inference and analytical entailment.

Simply studying predication and arriving at a logical form tells us little about analytical entailments. Analytic entailments are entailments of the logic underlying logical forms; they are valid and so true in all models for the language. But unless we add certain information based on the links between non logical words, we reduce the class of analytic entailments to the set of logical consequences in the language of logical form. If the lexicon is to add anything to the class of analytic entailments, it must do so in virtue of one of the following possibilities. (1) Certain lexical entries for words are logically complex and thus give rise to entailments using the logic validated by the models for the logical forms. (2) The lexicon contains axioms relating the meanings of words and thus adds to the logical theory of compositional semantics. (3) the rules for the analysis of predication in the particular the rules for handling complex types and the relations between types add information to logical form that the underlying logic can exploit to provide analytic entailments beyond the entailments provided by the underlying logic alone.

Linguists and philosophers have for the most part pursued the first track under the rubric of providing a "decompositional" semantics (Jackendoff's work here comes prominently to mind). Even our quite minimal hypothesis that every word root gives rise to a distinct type leads to at least some decomposition: distinct words with the same root should have some common bits of logical form; that is, the logical form of some words of the same root must be constructed out of the logical forms of at least one of the other words of the same root. For instance, many linguists have argued, for example, that *sink*'s syntactic behavior as a transitive but also intransitive verb indicates that its lexical meaning consists of two facts, one describing an underspecified action on the part of an agent, and the second describing the resulting state of that action, namely that the theme of the sinking is sunk. This yields a host of analytic entailments, some of which are not completely trivial:

(2.20) a. The enemy sank the ship with a torpedo ⟶ the enemy used a torpedo to sink the ship ⟶ the ship sank ⟶ the ship was sunk

b. The enemy sank the ship with a torpedo ⟶ the enemy's torpedoing the ship was the cause of its sinking.

Asher and Lascarides (1995, 2003) explore such decompositional analyses for a variety of verbs, including causatives and psych verbs. Here is the decomposition of Asher and Lascarides (2003) for *sink*. It involves the property of being sunk, a causal relation between two facts and an underspecified action marked with a question mark.

(2.21) sink: $\lambda x \lambda e \lambda y(?(y, x, e) \text{ CAUSE } \exists s(\text{sunk}(x, s) \wedge e \prec s)$

To handle the alternation between causative and intransitive *sink*, Asher and Lascarides assume that unsaturated arguments of the causing fact can be existentially closed off at the final stages of constructing the logical form for a clause containing such a verb. This decompositional analysis, together with analyses for *use* and gerunds, validates all the entailments in (2.20).

Levin (1993) contains a comprehensive list of verbs which invite similar decompositional analyses. They include verbs like:[16]

> bake, build, braid, brew, burn, carve, cast, chirp, cook, crochet, dig, draw, drill, fold, hum, knit, mumble, murmur, paint, sing, shoot, sketch, weave, shout, whisper, whistle, write, sink, break, open, close...

The following examples indicate how these very different verbs obey the same syntactic alternations as *sink*, suggesting a similar analysis.

(2.22) a. a. The bottle broke.
(intransitive/causative transitive)

b. b. John broke the bottle.

(2.23) a. a. The window opened suddenly.
(intransitive/causative transitive)

b. b. Mary opened the window suddenly.

[16] For a treatment of some of these, see Angeliek van Hout (1996) or Bittner (1999).

Asher and Lascarides (2003) generalize their decomposition to the class of psych verbs. They also all have an underlying causal structure, where some unspecified action by the subject causes the patient (the direct object of the psych verb) to have the state denoted by the past participle form of the verb. Thus, all actions denoted by psych verbs decompose into an event with a particular result state involving an adjectival form of the root lexeme. Such a decomposition obviously does not give a complete analysis of a verb like *bother* (it couldn't since it relies on the meaning of *bothered* to specify the meaning of the verb), but it is a useful way of portraying the similarity in meaning of psych verbs. Decomposition in this limited sense can show how the same basic lexical meaning can lead to several syntactic realizations and thus serves a useful purpose. These limited decompositions also provide for limited analytical entailments.

Nevertheless, the decompositional approach has received substantial criticism (Fodor, 1971). Fodor assumed that if a decompositional approach to lexical semantics, for example one in which *kill* is analyzed as *cause to die,* is psychologically real then it should take longer to process words with more complex meaning representations than those with simpler representations. However, many experiments have shown that this is not the case for the pair *kill* and *die*. Note, however, that *kill* does not support syntactic alternations like the ones above.[17]

Strongly decompositional approaches to word meaning have in general failed dramatically to provide plausible results for lexical meanings.[18] One major problem is that it is not clear what the primitives out of which all lexical meanings are built should be. It is reasonably clear that there should be some primitive corresponding to a causal relation between events or between facts. However, aside from some sort of causal primitive, it's not at all clear what the primitives should be.

In a few areas of lexical research such as the research on spatial and temporal

[17] It is true that *kill* takes different adverbials, in particular adverbials involving process like *sadistically*, *with a knife* and so on. These, so the argument goes, modify the process of *killing* (Pustejovsky (1995)). Pustejovsky takes these adverbials to be evidence that *kill* involves a process and a result state. This is not really convincing, however. What these adverbial modifiers show is that there must be a type distinction between events of killing and events like dying, not that there must be a decomposition involved, though indeed it does seem to be an analytical entailment that if Sam killed Pat then Pat died.

[18] However, recent experiments by Gail McKoon on activity, stative, change of state, and indirect causation verbs does provide evidence for the hypothesis that there is increasing representational complexity for these types as we go from one category on the left to the next on the right, as hypothesized by Rappaport and Levin (1993). McKoon discussed these experiments during her seminar at the LSA summer school in Stanford, California.

expressions, decomposition has led to more promising results. Researchers have largely agreed on the need for certain topological primitives (contact and weak contact, or equivalently parthood and weak contact) and geometrical ones. For time, the primitives required to construct linear orders are well known and axiomatized. More recently, the topological and even geometrical primitives have been axiomatized within a project to analyze the meanings of spatial, more precisely topological, prepositions (Vieu 1991, Aurnague 1991, Muller and Sarda 2002). Such work is extremely important for lexical semantics, because it gives a concrete meaning to the primitives, determining what sorts of models for the logical form language are the admissible ones (Asher and Vieu 1995, Donnelly 2002). Such an axiomatization yields a wealth of analytic entailments that go well beyond those given by the axiomatization of the first order connectives and quantifiers. There have also been, following Jackendoff's seminal work, thorough investigations of motion verbs (Sablayrolles 1995, Sarda 1999, Levin 1996), though correspondence theorems continue to be lacking for the motion verb primitives of goal, source, and path.

In most other areas of lexical meaning, the sort of thorough investigation that lexical research on temporal and spatial expressions has delivered is lacking, as far as I know. Axiomatizations are at best partial, and more often than not characterizations of lexical meaning rely on notions that are part of the background language in which one characterizes the reference of these terms and hence the models; and this frequently amounts to little more than a restatement of intuitions. There is of course a reason why axiomatizations in these areas don't exist. In the case of space and time or space-time, the mathematical structures underlying the meanings of various words in natural language were clear thanks to thousands of years of reflection on these topics by philosophers, mathematicians and physicists. It was relatively easy, though still not straightforward, to build a qualitative theory that characterizes that structure as it's reflected in language. But there's no guarantee that other areas of lexical meaning are amenable to such treatment. There have been hasty general treatments of lexical primitives like Shank's (1974) conceptual dependency paradigm. Such a description might lead to an abstract underlying structure and then perhaps an axiomatization. But one would need a much more detailed descriptive analysis, and there are no guarantees that an abstract structure or that an axiomatization would emerge therefrom. I don't know what is the abstract structure of buying and selling; looking at formal treatments in ontology (Smith, etc.) of various concepts, one gets the feeling that either there are no interesting structures worth axiomatizing or that research has somehow gotten off on the wrong track. It may simply be utopian to think that every part of the lexicon will end up looking like the parts concerning spatial and temporal expressions. Perhaps the best we can hope for is a partially decompositional analysis with some primitives properly understood.

There is a way to encode analytic entailments without full axiomatization or decomposition. That is to exploit the subtyping relation $\sqsubseteq$ on types. If $\alpha \subseteq \beta$ then anything of type α is also of type β, and this is itself a conceptual truth. That is, $\alpha \subseteq \beta$ reflects a relation of lexical meaning. This provides a simple way of writing down meaning postulates or truths that follow from the meanings of the constituent terms. For instance, we can easily encode the entailment that a bachelor is unmarried and male. via the following type constraints:

- $\textsc{bachelor} \sqsubseteq \textsc{male}$
- $\textsc{bachelor} \sqsubseteq \textsc{unmarried}$

We can go further, if we introduce another simple operation on types $\sqcap$, which is a meet and is defined as follows:

Definition 1 *$\sigma \sqcap \tau$ = the greatest lower bound of σ and τ in the partial order $\sqsubseteq$ on types.*

Given that $\sqcap$ is defined as greatest lower bound relative to $\sqsubseteq$, it follows immediately from these two principles that

- $\textsc{bachelor} \sqsubseteq \textsc{male} \sqcap \textsc{unmarried}$

If types are concepts, then linking types via the $\sqsubseteq$ relation really amounts to conceptual analysis, a time honoured approach to philosophy. But notice that in using the type hierarchy, it is much more natural (and productive) to look at one way entailments than try to define a given type (the standard approach in conceptual analysis). Repeated attempts to give necessary and sufficient conditions for concepts like knowledge in philosophy have ended in failure. But in a type hierarchy entailments can be captured without requiring any definitional reduction of one type to some boolean combination of other types. Nor do we have to take on board the dubious entailments of a profligately decompositional approach.

Analytic entailments aren't always exotic. Here are some simple and humdrum analytic entailments

(2.24) If it's Tuesday, it can't be Wednesday.

(2.25) If Kim bought something from Sandy, then Sandy sold something to Kim.

That is, the expressions that pick out days of the week pick out different days. This is a matter of what these expressions mean. One might capture such entailments by using a decompositional approach–e.g. *tuesday* would mean something like 3d day of the week, while *wednesday* would be the 4th. But we can also just use the type hierarchy to encode these facts by using the meet operation $\sqcap$. I will assume that for any two types σ and τ, there is at least one type that is a subtype of both and that's the type $\perp$.

- $\textsc{tuesday} \sqcap \textsc{monday} = \perp$

To express the equivalence between buying and selling we need to resort to a specification of propositional types using variables over types as parameters. I'll call such parametrized types *dependent types*. Dependent types are functions from a sequence of types, known as parameters, to types, typically subtypes either of the type of eventualities, the type of propositions or the type of entities generally. Let's suppose that *buy* has three arguments, a buyer, a buyee, and the object bought, labeled a_1, a_2 and a_3 respectively. Then, we can stipulate:

- $\textsc{buy}(a_1, a_2, a_3) \sqsubseteq \textsc{sell}(a_2, a_1, a_3)$
- $\textsc{sell}(a_2, a_1, a_3) \sqsubseteq \textsc{buy}(a_1, a_2, a_3)$

The analytical entailment involving *kill* and *die* can also be encoded with dependent types:

- $\textsc{kill}(a_1, a_2) \sqsubseteq \textsc{die}(a_2)$

Dependent types will turn out to be a very powerful tool in the analysis of lexical meaning. They will prove essential in the analysis of coercion. They also allow us to individuate quite finely subtypes of the type of propositions. Finally, they serve to model lexical decomposition at an abstract level.

I will argue below that in cases of coercion or copredication, new entailments based on lexical meaning and predication can arise from the adjustments in types needed to make the predication work. By using type rules to add information to logical form, we end up modeling all sorts of entailments that don't follow simply from the meanings of the syncategorematic or logical words of the language. Some of them could even be defeasible entailments such as *start a cigarette* $\mid\!\sim$ *start to smoke a cigarette*. For these it is the mechanisms of predication that introduce the entailment into logical form.

Assuming that every lexical root has its own unique type as I do thus doesn't preclude having a non trivial theory of lexical inference encoded in the type hierarchy. Subtyping relations between types enable us to encode entailments based on lexical meaning. And having a type associated with each word root doesn't preclude a decompositional analysis of various words in terms of the root type—consider the case of psych verbs, whose meaning is analyzed in terms of their adjectival correlates. It's a big step further, and as far as I can see a logically unsupported step, to conclude as Fodor and Lepore (1998) do that lexical semantics must couple an anti-decompositionalist approach together with a refusal to adopt any form of meaning postulates or partial axiomatization of word meanings. This is clearly not enough to lay the issue of analytical entailment to rest, but it shows that the issue can be approached in a sensible fashion.

Chapter 3

Previous Theories of Predication

With this survey of some issues confronting a theory of typed predication, I now turn to examining some precursors of the theory to be proposed here. All of these models assume that the sort of data examined above is the province of lexical semantics, and not relegated to a nonlinguistic level of world knowledge (contra Fodor, Lepore and Capellen). Several informal models of predication have been advanced that attempt to explain phenomena pertaining to coercion. All of these approaches make some claims about words with multiple senses or polysemy. Like discourse semantics, lexical semantics also has to deal with how to represent ambiguity. As in discourse semantics, we will adopt an underspecification approach to ambiguity. In discourse semantics, in effect, this is now the standard approach, but puzzingly it's not the norm in lexical semantics. Some people distinguish between monomorphic and polymorphic languages, which corresponds to two different approaches to ambiguity. A monomorphic language (or lexicon for a language) is one in which each word (we think of ambiguous words as many words with the same orthography on this view) has a unique type and syntactic category. This is the view found in Montague Grammar or even in HPSG, and it corresponds to an approach to ambiguity according to which ambiguous expressions are represented by the set of their disambiguations. A polymorphic language (or lexicon for a language) is one in which each word may have multiple types or equivalently underspecified types that may be further specified during the composition process. This corresponds of course to the underspecification approach to ambiguity.

3.1 The Sense Enumeration Model

The most orthodox model is the sense enumeration model of the lexicon, according to which all the different possible meanings of a single lexical item are listed in the lexicon as part of the lexical entry for the item. Each sense in the lexical entry for a word is fully specified. On such a view, most words end up being ambiguous. This account is by far the simplest conceptually, and it is the standard way dictionaries are put together. In effect this view posits many types for each word, one for each sense. Predication may select for a subset of the available senses for a predicate or an argument because of type restrictions. While conceptually simple, this approach fails to explain how some senses are intuitively related related to each other and some are not. This is not only a powerful intuition shared many linguists and lexicographers. There are also linguistic tests to distinguish between word senses that are closely related to each other and those that are not. Words or, perhaps more accurately, word occurrences that have closely related senses are called logically polysemous, while those that do not receive the label 'accidentally polysemous' or simply 'homonymous'. Copredication is one test used to distinguish logical polysemy from accidental polysemy (Cruse 1986): if two different predicates, each requiring a different sense, predicate properties of different senses of a given word felicitously, then the word is logically polysemous with respect at least to those two senses. Another test is pronominalization or ellipsis: if you can pronominalize an occurrence of a perhaps ambiguous word felicitously in a context where the pronoun is an argument of a predicate requiring one sense while its antecedent is an argument of a predicate requiring a different sense, then the word is logically polysemous at least with respect to those senses.[1] Contrast (3.3a-b) and (3.3c-e):

[1] Of course there are other tests for ambiguity, such as the so called contradiction test, according to which a particular sentence with an ambiguous word ω may be consistently judged both true and not true in the same context since an interpretation may access different senses of ω. Read with contrastive stress, sentences like

(3.1) The bank specializes in IPOs

(3.2) The book is beautiful.

can be understood as true and false in the same context. Nevertheless, the contradiction test is not a sure fire indicator of real ambiguity. Books are both informational and physical in some sense even though we can select each one of these aspects within a predication. We can also access both meanings simultaneously.

(3.3) a. #The bank$_i$ specializes in IPOs. It$_i$ is steep and muddy and thus slippery.

b. #The bank specializes in IPOs and is steep and muddy and thus slippery.

c. Lunch was delicious but took forever.

d. He paid the bill and threw it away.

e. The city has 500 000 inhabitants and outlawed smoking in bars last year.

Bank is a classic example of an accidentally polysemous word. As (3.3a-b), show, both the pronominalization and copredication tests produce anomalous sentences, which confirm its status as accidentally polysemous. On the other hand, *lunch bill* and *city* should be classified as logically polysemous, as they all pass the tests of copredication and pronominalization.

The distinction between accidental and logical polysemy isn't absolute. There are degrees of relatedness that the felicity of copredications and pronominal tests reflect. Contrast (3.3d-e) with (3.4ab), for instance:

(3.4) a. ?My janitor uses a brush and so did Velazquez.

b. ?The city outlawed smoking in bars last year and has 500 000 inhabitants.

(3.4ab) aren't as good; they're rather zeugmatic. Their zeugmaticity results from different factors; (3.4b) as opposed to (3.3e) involves the same two lexical senses but in a different order. This suggests that copredications are subject to discourse effects. On the other hand, (3.4a) involves two relatively unrelated senses of *brush*—paint brushes and cleaning brushes have for the most part very different properties—different functions, sizes, and so on. These senses don't cohere together in the way that the distinct senses for *lunch*, *bill*, *book* and *city* do. Though a full explanation promises to be complex, sense enumeration models have no way of explaining the differing degrees of success that copredications appear to have. In addition they do not posit a structure to lexical content making it difficult to say much in the way of linguistic generalizations.

3.2 Nunberg and Sense Transfer

Not all lexical theories focus their attention on logical polysemy and the tests for it. Many theories focus instead on coercion, which brings to mind a quite different view of lexical meaning and transformation. Coercions are often analyzed as involving shifts of meaning of a term. Frege is one famous philosopher who has such a theory of meaning. He thinks that intensional operators like verbs of saying or of propositional attitudes induce a meaning shift of the terms within their scope. He stipulates that the meaning of terms when they occur within the scope of an intensional operator shifts from their reference to their sense. The intensional operator coerces a meaning shift, a shift from customary meaning to non customary meaning. Predication for Frege had to involve certain type checking in order to discard the application of, say, a belief relation to the simple extension of its clausal argument. Depending on whether one subscribes to the many levels of sense interpretation of Frege or not, one can say that such coercions are carried through the sense hierarchy as the depth of embeddings under intensional operators increases.

Generalizing Frege's strategy, Nunberg's "sense transfer" view is the first proposal that lexical meanings be subject to shifts in predication (Nunberg (1979, 1995)). A lexical entry specifies a denotation of a referring expression or of a predicate (type of denotation) according to Nunberg. While Nunberg isn't completely explicit about what the specifications of lexical meanings look like, his ideas fit within the standard framework in which the denotations of terms are given as lambda terms.[2] These terms will then have their standard interpretation in a model. For instance, the word *cat* will have as its lexical entry an expression $\lambda x cat(x)$ and its interpretation in the intended model will be that function that for any given world w and time t returns 1 if x is assigned as a value something that is a cat at w and t, and returns 0 otherwise.

Sometimes such lexical entries can be transformed via some general rule like grinding or by a more specialized rule transferring the normal interpretation of a term to some salient associated entity when predication demands this. Nunberg applies such a notion of transfer to definites and indexicals (thus it would appear that Nunberg adopts a version of a type driven theory of predication). The examples Nunberg and others have found to motivate this view are quite vivid:

(3.5) I'm parked out back.[3]

[2] For instance, see Sag (1981) for a formal development of his view.

[3] Note that these examples are felicitous with all sorts of subjects.

a. i. John is parked out back.
   ii. The students are parked out back.
   iii. Most students park out back.

(3.6) The ham sandwich is getting impatient.

The basic idea is quite intuitive. In these examples, applying a particular predicate whose argument is required to have type α to an argument whose type is β, where $\alpha \sqcap \beta = \bot$ forces either the argument term or the predicate term or both to change their meanings so that the predication can succeed. For instance, ham sandwiches can't literally get impatient, and if I'm standing in front of you I can't literally be out back. So what happens is that we shift the meaning of the terms so that the predications succeed: it's my car that is parked out back, and it's the guy who is eating the ham sandwich who is getting impatient.

The problems lie in the details. When exactly is the sense transfer function introduced? Let's suppose with Sag (1981) that Nunberg's sense transfer functions work on lexical entries for common noun phrases or N′. The result we want for (3.6) is clear. 'The ham sandwich ' should have the following logical form

(3.6') $\lambda P the(x)(f(ham\ sandwich)(x) \wedge P(x))$,

where f is the transfer function mapping ham sandwiches to people who are eating them or who have just eaten them. The problem is that we only become aware of the need for a type adjustment mechanism in building the logical form of the sentence when we try to put the subject noun phrase together with the verb phrase, and at that point it is no longer straightforward to add the transfer function to work over the contribution of the common noun. Indeed if we think of the process of predication here as building up a denotation for the sentence in compositional fashion, then it is *hopeless* to try to apply the transfer function in the place that Sag and Nunberg would like; the common noun denotation is simply no longer available at this stage of composition. If we think of the process of predication as building a logical form rather than a denotation, then the contribution of the common noun is in principle available, but specifying *what* its scope is is not straightforward. It would require some sort of rewriting rule, whose status as a rule in the lambda calculus is not clear.

Moreover, it appears that it cannot be always a common noun that should be shifted in coercion cases. We have to generalize Sag's assumption, to get an analysis of (3.5): either the full noun phrase or the verb will have to undergo sense transfer. But if it is the entire noun phrase or, in the language of syntax, the DP (determiner phrase) that is shifted we seem to get incorrect truth conditions. Consider (3.7):[4]

[4] Thanks to George Bronnikov for this example.

(3.7) George enjoyed many books last weekend

A straightforward application of the Nunberg strategy yields the following logical form:

(3.7') $f(\text{many books}(x))\exists e(\text{enjoy}(g, x, e) \wedge \text{last weekend}(e))$

Notice now, however, that the transfer function shifts the meaning of the entire quantifier, and so presumably the quantifier ranges over the elements in the image of the transfer function, namely some sort of eventualities—perhaps, for the sake of concreteness, events of reading books. What (3.7') says is that there were many book reading events that George enjoyed over the weekend, which is compatible with there being just one book that George read over and over again that weekend. Clearly, this is not a possible reading of (3.7), and so we cannot implement Sag's proposal for Nunberg's idea straightforwardly for all coercions. In fact, the observation about (3.7) constitutes a problem for many extant accounts of coercion, Pustejovsky's GL account included, as we'll see below.

Of course Nunberg himself did not investigate coercions involving *enjoy*. But the same observation holds for the "parking" examples. Consider the difference between the two minimal sentences in (3.8):

(3.8) a. Everyone is parked out back and is driving an old volvo.
b. # Everyone is parked out back and is an old volvo.

(3.9) a. ??John was hit in the fender by a truck and will cost a lot to repair.
b. John's car was hit in the fender by a truck and will cost a lot to repair.

(3.8a) is fine but relies on the fact that the subject DP does not shift. Were there shifting going on, then (3.8b) should be good, but it is plainly not. Similarly if the DP had actually shifted meaning, then (3.9a) should have a reading that is equivalent to (3.9b), but it does not. (3.9a) is simply semantically anomalous.

An alternative strategy is to shift the meaning of the verbs. This can in principle avoid the problem noted with (3.7). But this cannot be done at the level of denotations. Consider the following ellipsis facts.[5]

(3.10) a. I'm parked out back, and Mary's car is too.
b. ? I own a car that is parked out back and Mary's car does too.
c. Mary enjoys her garden and John his liquor.
d. ?Mary enjoys digging in her garden, and John his liquor.

[5] Thanks to Alexandra Aramis for suggesting these sorts of examples.

If we transfer the sense of *parked out back* to the property of owning a car that is parked out back to account for the parking examples, then the ellipsis in (3.10a) should be odd: Mary's car does not own a car that is parked out back as well. Cars aren't the sort of things that can own cars. But the ellipsis is fine. Similarly if we transfer the sense of *enjoy* to something like *enjoy digging* to make sense of the predication that Mary enjoys her garden, then we should predict that the ellipsis in (3.10c) should be bad when it is fine.

If we do not apply meaning transformations at the level of denotations, we must spell them out at the level of logical form. And we can't do this in the obvious way without predicting that (3.10a,b) are equivalent—similarly for (3.10c,d). That is we cannot simply replace the contribution of *parked out back*, which I'll assume is a 1-place predicate here, to one that looks like this: $\lambda x \exists z(\text{parked-out-back}(z) \wedge \text{owns}(x, z))$. Instead one needs to make use of underspecification and of ways of resolving underspecifications in ellipsis which are not part of Nunberg's proposal and which were not really understood at that time.

- $\lambda x \text{parked-out-back}(x) \longrightarrow \lambda x \exists z(\text{ parked-out-back}(z) \wedge ?(x, z))$

- $\lambda y \lambda x \text{enjoy}(x, y) \longrightarrow \lambda x \lambda y \exists z(enjoy(x, z) \wedge ?(z, y)$

The '?' stands for a relation that must be filled in from the discourse context. Crucially, as Asher (1993) argues, such underspecifications in ellipsis contexts can be resolved one way in the source of the ellipsis and another way in the target. This explains why (3.10a,c) are good and distinct from the degraded ellipses in (3.10b,d). This gets us something like the right logical form for sentences such as (3.7) and other examples of coercion.

Many questions about this proposal still remain, however. For instance, is the shift at the level of logical form really to be described as a change of meaning of the verb in these predicational contexts? Is it really conceptually plausible that these verbs shift their meaning but that other words do not? Exactly which sentential constituents can be arguments of the sense transfer function? Exactly what triggers the application of the function? A more serious problem with this account is that it doesn't answer the question, "Why this meaning transformation and not another?" There are no constraints on when a sense transfer function can be introduced at all. It's clear that one can't invoke an arbitrary function whenever predications break down, for then we could not predict that any sentences are semantically anomalous. The whole type system has become otiose in that case. But neither can we appeal to some function that's salient in the context. Here's an example from Kleiber (1999), where it's perfectly clear what the transfer function should be in the context that would make this predication acceptable. The function takes us from pianos to the noise they make. Nevertheless, we cannot get (3.11) to mean that we heard the sound of the piano which came to us floating over the waters of the lake. The sense transfer model has no explanation why.

(3.11) Nous entendimes le piano, qui nous parvenait flottant par-dessus du lac (We heard the piano which came to us floating over the waters of the lake).

More generally, there's no account of what would validate the application of transfer functions. Why should we be permitted to make such transfers in some cases but not in others?

The notion of a sense transfer function doesn't address the distinction between logical and accidental polysemy and the data about copredication and pronominalization. How does the sense transfer view interact with these tests? The sense transfer function is designed to make sense of examples like:

(3.12) Plato is on the top shelf on the right.

where the speaker is talking about books or a book by Plato. Nevertheless, Kleiber argues that sense transfer doesn't work when an anaphor and its antecedent are forced by the predication contexts to have two different senses linked by such a transfer function.

(3.13) a. George Sand est lue par beaucoup de monde, bien qu'elle soit disparue depuis longtemps (George Sand is still read by many people even though she died long ago).

b. ?George Sand est lue par beaucoup de monde, bien qu'ils ne soient plus édités (George Sand is still read by many people even though they (the books she wrote) are no longer in print).

Taking their cue from such examples, Kleiber and others reject Nunberg's analysis of logical polysemy using reference shifters. They conclude that examples like (3.5) and (3.6) point to a different sort of phenomenon from logical polysemy exemplified by the classic coercion cases involving aspectual verbs.

Kleiber's observations have some force. Nevertheless, once again judgments in acceptability reveal a not completely black and white distinction. The following examples (from Abusch 1989, WCCFL 8) show that Kleiber's examples don't give us the full story. In some cases we can get anaphoric links between two expressions whose meanings are related by a sense transfer function of the sort that Nunberg envisions.

(3.14) a. The mushroom omelet left without paying, because he found it inedible.

b. ? The mushroom omelet is eating it with gusto.

And it appears that sometimes copredication works as well in these cases. (3.15) is somewhat zeugmatic but passable, I think:

(3.15) Plato is one of my favorite authors and is on the top shelf to the right.

3.3 Kleiber and Metonymic Reference

Georges Kleiber proposed another model under the rubric of "metonymic reference" to account for coercion.[6] This model uses part whole relationships to understand coercion and many examples of predication. Properties of some parts of objects can be predicated of the whole. Kleiber says that we can include as parts of an objects things commonly associated with it according to this metonymic model. But clearly now the part whole relation is no longer the usual one and so the model becomes rather mysterious. We clearly don't want to consider the fender of John's car to be part of John in any standard mereological sense!

(3.16) a. Paul est bronzé. (Paul is tanned).

b. Les américains ont débarqué sur la lune en 1969. (The Americans landed on the moon in 1969).

c. Le pantalon est sale. (The trousers are dirty)

d. Le stylo est rouge. (The pen is red)

e. John was hit in the fender by a truck.

Modifications of some of these examples show that part whole relations sometimes figure in copredications:

(3.17) a. Paul est bronzé et très athlétique. (Paul is tanned and very athletic)

b. Les américains ont débarqué sur la lune en 1969 et ont mené une sale guerre en indochine. (The Americans landed on the moon in 1969 and waged a dirty war in Indochina.)

c. Le pantalon est sale et troué. (The trousers are dirty and torn)

d. Le stylo est rouge et très cher. (The pen is red and very expensive)

[6] Kleiber's model is similar to Langacker's notion of "active zones".

However, the part whole relation can't cover all the cases of logical polysemy. In particular, it's not clear that we want to say that part of the lunch was delicious and part of the lunch took forever in (3.3c), repeated below:

(3.3c) Lunch was delicious but took forever.

While there is no doubt that predication may take advantage of a part whole relation, one needs more details as to how the compositional construction of logical forms proceeds in Kleiber's case as well. When does the metonymic shift occur and why does it occur? Kleiber's view isn't sufficiently developed in order for us to be able to answer these questions.

The examples of metonymy are not examples of semantic, type coercion. It is in principle possible from the perspective of conceptual coherence that, for instance, the entire population of America landed on the moon in 1969. Of course that's false and unbelievable, but that's not a semantic matter. In the Nunberg examples or the examples of coercion involving aspectual verbs, something like sense transfer or type coercion has to take place for semantic reasons: the predicate and the argument can't as they stand combine without producing a semantic anomaly.

Kleiber nevertheless extends his model to deal with coercion cases involving aspectual verbs. Kleiber sensibly enough postulates a temporal argument for *begin* which is temporally extended and has a well defined initial part and then a continuation (typically expressed by an accomplishment verb with telicity and durativity). For any other argument there must be a homomorphism from the accomplishment structure to the object. This implies that *begin* must have something like an accomplishment as its object. But now how does this relate to the metonymic model? It is not at all obvious that the relationship between an eventuality (Laurence's smoking a cigarette) and one of its participants (the cigarette) is one of part and whole. Still less is it clear that such constraints are sufficient to capture the relevant cases. Consider for example:

(3.18) a. Paul a commencé de déconstruire son livre. Marie a commencé la sienne aussi. (Paul began to deconstruct his book. Mary began hers too.)

b. Paul a commencé de déconstruire son livre. Marie a commencé à déconstruire la sienne aussi. (Paul began to deconstruct his book. Mary began to deconstruct hers too.)

One can construct a homomorphism from the deconstruction of the book to a temporal structure that captures accomplishments. Then Kleiber's model would predict that (3.18a) has a reading that is a paraphrase of (3.18b). But it does not.

3.4 "Classic" GL

The last approach to polysemy that I shall treat here is by far the most developed of the ones I have surveyed. That is the model given by the Generative Lexicon or "GL". The present approach draws its inspiration from many of the ideas in GL as developed by Pustejovsky (1995) and others. GL is motivated by the phenomena of coercion and logical polysemy. It aims to preserve compositionality as much as possible while giving an account of these phenomena beginning with a single lexical entry for logically polysemous words. In this it follows the outlines of a theory of "le sens général" of a word. A "sens général" is a general, perhaps underspecified meaning for all the uses of the word that can be specialized via predication. Pustejovsky (1995) thinks that specific lexical meanings are generated in the meaning composition process from interactions of type constraints and something like a sens général. The hard work is to figure out an appropriate format for a "sens général" and the procedures that in appropriate contexts will specify this underspecified meaning.

GL postulates a lexicon with the following parts:

1. **Argument Structure**: Specification of number and type of logical arguments.

2. **Event Structure**: Definition of the event type of a lexical item. Sorts include STATE, PROCESS, and TRANSITION, nothing really special here.

3. **Qualia Structure**: A structural differentiation of the predicative force for a lexical item.

4. **Dot Objects**: in *The Generative Lexicon* they are part of the qualia structure but they involve something more.

5. **Lexical Inheritance Structure**: Identification of how a lexical structure is related to other structures in the type hierarchy.

Classic GL avails itself of the formalism of attribute value matrices, or AVMs, combined with types known as typed feature structures, a very good formal account of which can be found in Carpenter (1992). In effect, a TFS can be thought of as a type with a bit of information attached via “attributes” whose values are more types. Formally, in the language of type theory, this amounts to associating with the “head” type of the TFS a record or collection of types. In effect these are all dependent types of the head type; when they take the head type as an argument, they return another type. Since there is a natural translation from feature structures to first order formulas, the use of TFSs for lexical entries allows one to combine type information with information that naturally goes into the logical form of a lexical entry.

The two most innovative aspects of GL are qualia structures and dot objects. Following Moravcsik’s article (Moravcsik (1975)), GL postulates that many nouns contain information about other objects and eventualities that are associated with the denotation of the noun in virtue of something like the Aristotelian explanatory causes or $\alpha\iota\tau\iota\alpha$.

Qualia Structure

The Qualia Structure of a word specifies four aspects of its meaning:

- ○ CONSTITUTIVE: the relation between an object and its constituent parts;
- ○ FORMAL: that which distinguishes it within a larger domain;
- ○ TELIC: its purpose and function;
- ○ AGENTIVE: factors involved in its origin or in “bringing it about”.

Qualia Structure of DOOR

$$\left[\begin{array}{l} \textbf{door(x,y)} \\ \text{const} : \textbf{aperture(y)} \\ \text{formal} : \textbf{physobj(x)} \\ \text{telic} : \textbf{walk\_through(e',w,y)} \\ \text{agentive} : \textbf{make(e,z,x)} \end{array}\right]$$

Qualia Structure of CIGARETTE

$$\left[\begin{array}{l} \textbf{cigarette(x)} \\ \text{const} : \textbf{tobacco(x, y)} \\ \text{formal} : \textbf{physobj(x)} \\ \text{telic} : \textbf{smoke(e, x, y)} \\ \text{agentive} : \textbf{roll(e', x, y)} \end{array}\right]$$

Pustejovsky and his students use the qualia information to introduce an appropriate eventuality or eventuality type as an argument of, for example, *enjoy, begin* and *finish*.

(3.19) a. begin a cigarette (i.e., smoking)
b. enjoy the book (i.e., reading)

According to classic GL, these data concerning semantic composition indicate a rich type structure in which objects of a given type α would conventionally associate with objects of other types relevant to an object of type α’s production or function; GL theorists also posit that this type structure is accessible to and guides the composition process. For example, they hypothesize that in predications like (3.19, the verbs *begin* and *enjoy*, which require an eventuality for their internal arguments, select as their arguments the eventualities encoded in the qualia structures of nouns like *cigarette*.

Qualia-based typing information is also motivated by coercion phenomena in certain prenominal adjectival modification constructions:

(3.20) a. fast car
b. fast motorway
c. fast water

To explain these data, Pustejovsky and Boguraev (1993) argued that *fast* selects for the TELIC-role of the NP head, sometimes acting coercively. For most of the above Adj-N combinations, we arrive at interpretations such as “cars that are fast moving” and “motorways that permit fast traffic.” Bouillon (1997) discusses a wide range of phenomena which are handled by reference to qualia in such constructions, and Bouillon and Busa (1998) extend this approach to an even broader set of Adj-N constructions.

Another application of qualia-based typing has been to account for certain denominal verb formation cases, as discussed in Hale and Keyser (1993). Hale and Keyser’s canonical examples are illustrated in (3.21) below.

(3.21) a. John put the books on the shelf/ J. shelved the books.
b. John sent the letter by fax./ J. faxed the letter.
c. John put the wine in bottles./ J. bottled the wine.

Within GL, the phenomenon has been analyzed as follows: the noun-to-verb transformation is licensed in just those cases where the noun’s TELIC role involves an object that matches the verb’s direct object. For example, from the noun *shelf*, whose TELIC makes reference to the relation of holding books (and related objects), the verb *shelve* is licensed, because its direct object shares the argument referenced in the noun's TELIC value. Climent (2001) discusses this idea in in more detail.

Let's now turn to a critical examination of the framework. First of all, we can quickly dispense with the idea of meaning generation during the composition process. There is no generation of meaning in the process of meaning composition in classic GL; meaning composition simply selects components of lexically specified meanings in the analysis of the coercion or copredication phenomena. The exploitation of qualia in the purported explanation of the various phenomena just listed simply focusses on a particular component of the lexical meaning of a noun and unifies this with the typing requirement of a predicate of the noun phrase or higher projection thereof (e.g., DP). The generative lexicon should rather have been called "the selectional" or "specificational" lexicon. Nevertheless, a selectional process in meaning composition is an interesting idea. How far does it take us?

It turns out that the notion of a selectional meaning composition process is not well developed in classic GL and breaks down rather rapidly in the face of recalcitrant data and formal scrutiny. First, it is often difficult to understand exactly what the values of the qualia are supposed to be. Is the telic role of a shelf, its purpose, really just to hold books? Presumably not. Shelves can hold all kinds of things—wine glasses, ski boots, outboard motors, clothes. So we should predict that sentences like

(3.22) a. John shelved his sweater.

b. Mary shelved her glasses.

c. Samantha shelved her chain saw.

have a similar meaning to *John shelved the books*, but if (3.22a-c) make sense at all, they involve a shift to a metaphorical sense of shelve—something similar to *shelve an idea*. So it looks like the analysis for denominal verb formation over generalizes; that is, the criterion proposed by classic GL isn't sufficient. An additional problem is that the qualia values seem to shift depending on the examples. For instance, to get the right meaning for *fast car*, which is a modal meaning (has the ability to go fast), we have to say that the car's telic role is to be able to drive or to be able to transport people. But if we modalize the value of the telic quale, then we should expect *John enjoyed the car* to mean something like *John enjoyed the ability of the car to drive or to transport people*, and we wouldn't expect the much more natural reading—*John enjoyed driving/ looking at the car*. Similarly for the telic role of *motorway*. In addition, one can doubt that the claim put forward by classic GL concerning denominal verb formation—that having the right qualia structure is a necessary condition for this phenomenon—is true. For example, the classic examples of denominal verbs like *lunch*, *snack*, *dine* and so on have intransitive uses in which there are no direct objects that could be referenced in the telic roles associated with the nouns *lunch*, *snack* and so on.

In spite of these worries, the telic roles of many nouns are nevertheless quite clear. It is much less clear what the constitutive or formal qualia are for many objects. Consider for instance the constitutive quale for *door* vs. *tobacco*. For *tobacco* the constitutive is something like an Aristotelian material cause, the matter out of which the object is constructed, but that is hardly the interpretation of the constitutive quale for *door*, which classic GL takes to be an aperture. Apertures are not matter—they are precisely the lack of matter. These remarks might seem quibbling, but there is a general point behind them. If the qualia are not defined in a precise way, then the theory can postulate anything for the values of qualia and thus loses explanatory power.

The general unclarity concerning what the qualia structure of a given noun is extends from the particular values of qualia to their types. What exactly is the type of the qualia object? This is a legitimate question for classic GL since it makes heavy use of types. An aspectual verb like *begin* seems to take an event type as its object argument or at least some sort of intensional event description, as it is perfectly coherent to say:

(3.23) a. John began to read that book, but he never finished (reading) it.

b. John began that book, but he never finished it.

If the logical form of the first clause of either (3.23a) or (3.23b) were something like

(3.23’) $\text{begin}(j, e, e') \wedge \text{read that book}(e)$

then we would predict that (3.23a,b) imply that there was a reading of the book and that that event was incomplete or unfinished, which if not outright contradictory is pretty problematic. Assuming that the infinitival contributes a property of events or event types as the argument of *begin* avoids this problem and invites a straightforward intensional analysis of (3.23a,b) along the lines of well known modal analyses of the progressive (Dowty 1979, Asher 1992).[7]

[7] Assuming that an infinitival provides an event type or a property of events would also be in keeping with the compositional semantics of infinitivals. See Asher (1993) for a discussion.

On the other hand, we don't have imperfective paradox-like phenomena with the verbs *enjoy* or *finish*. If it's true that I enjoyed smoking my cigarette then I did smoke my cigarette. Notice also that *enjoy* doesn't, unlike the aspectual verbs, subcategorize for infinitival clauses, at least in English.

(3.24) a. Nicholas enjoyed smoking a cigarette.

b. Nicholas enjoyed a cigarette.

c. #Nicholas enjoyed to smoke a cigarette.

d. Nicholas finished a cigarette.

e. #Nicholas finished to smoke a cigarette.

However, in classic GL the distinction between event types and events isn't clear at all. Some interpreters of GL (Egg 2003) have taken the object of the coercion verb to be an event type; others (Asher and Pustejovsky 2001, or Copestake and Lascarides 1996) have take the arguments to be events. The truth seems to lie somewhere in the middle, and to depend on the verb.

Our discussion of event types versus events as qualia values leads to problems in the compositional semantics. In fact, the interaction between compositional semantics and lexical semantics is hardly ever discussed in classic GL, and this leads to problems—not surprisingly if you believe, as I do, that a theory of lexical meaning has to also specify a method of meaning composition or an account of predication. For instance, some authors like Kleiber have claimed that GL does not account for the data concerning anaphoric availability. That is, the sort of coercions induced by aspectual verbs are alleged to fail the pronominalization test. Suppose we introduce an element into discourse during coercion that has the type of an eventuality, as the telic roles for the AVMs for *book*, *cigarette* and other artifacts do in GL. Then we should predict these eventualities to be accessible for future anaphoric reference, if none of the well-known semantic or syntactic principles barring anaphoric links obtain. However, the marginal acceptability of (3.25a) translated from Kleiber (1999) and (3.25b) appear to put this hypothesis in jeopardy.

(3.25) a. Paul has started a new book. ??It (that) will last three days.

b. ?Paul enjoyed his new book, though it was quick.

It's clear what is intended in these examples: an anaphoric reference to the reading event. Of course if we distinguish between the types of objects for aspectual verbs like *start* and verbs like *enjoy* as I did above, then accounting for the difference in acceptability of (3.25a,b) is straightforward.

Other examples, however, show that an event is anaphorically available in coercion examples, when the discourse context is right.

(3.26) a. Paul has started a new book, but that won't last.[8]

b. Paul has started a new book. His reading will take him three days.

(3.25a,b) and (3.26a,b) all require an event denoting expression to be accessible to the pronoun. I take it that (3.26a,b) are acceptable. The contrastive construction in (3.26a) makes the event more salient. The use of a definite description in (3.26b) also makes it easier to pick up the event of Paul's doing something with the book. However, other, similar uses of aspectual verbs as in (3.27), repeated below, also support anaphoric reference to events without difficulty when the discourse context is of a special kind.

(3.27) Last week Julie painted her house. She began with the kitchen. That didn't take very long. She then proceeded to the bedroom and the living room. That took forever, because she painted friezes on the walls in those rooms.

In this use of aspectual verbs anaphoric reference to the events of painting the rooms is easy. Such examples also pass the copredication test when we use relative clauses:

(3.28) Yesterday Julie painted her house. She began with the kitchen which didn't take very long. She then proceeded to the bedroom and the living room, which took forever, because she painted friezes on the walls in those rooms.

These examples show that the pronominal test with coerced eventuality readings does succeed, though the reasons for this success have to do not with qualia but with complex interactions between discourse structure and clausal logical form. It's not clear what, if anything, classic GL predicts about the possibility of referring anaphorically to an event within the qualia structure. It had better not be the case that the eventualities mentioned in the qualia are all available as anaphoric antecedents. That would give plainly false predictions. A more sensible hypothesis, empirically speaking, is that the terms denoting eventualities in the qualia

[8] Thanks to Elizabeth Asher for this example.

$$\left[\begin{array}{l} \textbf{cigarette} \\ \text{sem arg1} : \textbf{[1]} \\ \text{Pred} : \textbf{cigarette} \\ \text{telic} : \left[\begin{array}{l} \textbf{smoke} \\ \text{Pred} : \textbf{smoke} \\ \text{sem arg1} : \textbf{[2]} \end{array}\right] \\ \end{array}\right]$$

Figure 3.1: Partially Rewritten Qualia Structure of CIGARETTE

structure only become available as anaphoric antecedents when they become the arguments of the coercion predicate, giving them as it were some sort of linguistic realization. But when that happens does that mean that the original object of the coercion predicate is no longer linguistically realized? Is it no longer available for anaphoric reference? That seems plainly wrong. Figuring out a principled view of what is going on, however, will require much more attention to the details of composition and interpretation than is available in classic GL.

Qualia structures are written out in the formalism of typed feature structures or TFSs. The language of TFSs is an elegant formalism for formulating a composition logic. A typed feature structure is a function from a type to a set of pairs of attributes and values. These values may either be constants (for instance predicates of logical form), variables with indices that are to be identified with values of some other attributes or typed feature structures themselves. Values are thus recursively defined. The standard language of feature structures has a modal semantics (Blackburn (1997), Lascarides et al. (1996)). The models for TFSs correspond to directed acyclic graphs (DAGS). One can define the semantics of TFSs by thinking of features as modal operators that label arcs between the nodes of a DAG, types as propositions holding at nodes, and constraints on types as conditionals (cf. Blackburn, 1992). Notice that the types themselves don't have any particular structured semantics in this analysis. All types, that is, are considered as atoms.

Despite the elegance of the framework, there is more than a bit of "grunt work" to get everything to work out as advertised concerning the analysis of the phenomena. Classic GL leaves much to be desired on that score, as many have noted. Pustejovsky (1995) is rather informal and uses first order formulas, as we saw above in the examples of qualia structures, instead of the proper formalism for TFSs. To properly represent the feature structure for *cigarette* we would need to rewrite the feature structure above as:

It's quite clear that we want the value of the telic to be defeasible as well, since it may be that when we say that Max enjoyed the cigarette, he enjoyed eating it (it might be a chocolate cigarette). To reflect this fact we would need to add the default type assignments of Lascarides, Briscoe, Asher and Copestake (1996).

Once we have the appropriate formalism for qualia in TFSs, we are still not quite done. The operation of unification over TFSs is a binary operation that returns a new TFS where the old values in the common arguments of two TFSs are replaced with their meets (greatest lower bounds in the type hierarchy) and then adds all the argument value pairs that are not shared. Typed unification is unification checked by typing; for instance, we can unify the feature structure for *cigarette* with the feature structure for *enjoy*, provided that the object argument's type in the TFS for *enjoy* (the object argument in the TFS for *enjoy* is just a variable that will be replaced roughly under unification with the feature structure for *cigarette*) has a meet with CIGARETTE (the type) that is not the absurd type $\perp$. By hypothesis, however, the object argument for *enjoy* is an event, while CIGARETTE is a subtype of physical object and has no non absurd meet with the event type. So the composition crashes here, unless we can coerce either the type of the appropriate argument slot of the verb or the type of *cigarette*. Classic GL's idea is that in this case, one of the qualia types of *cigarette* should be substituted for CIGARETTE. But how is this done? How do we in fact shift the predication to hold of the telic value when we have a phrase like *enjoy the cigarette*? How exactly is the predication in coercion cases involving aspectual verbs or verbs like *enjoy* supposed to work in terms of unification? Does one unify the variable in the argument place of the verb with something of the appropriate type in the AVM or feature structure associated with the element that is supposed to fill the argument position? Something like this must be going on, but such a rule is not part of unification standardly construed.

Neither a defeasible version of unification like that in Lascarides, Briscoe, Asher and Copestake (1996) nor classic GL provide rules suitable to the analysis of coercion. The unification formalism underlying TFSs is not by itself sufficient to get the appropriate bit *in the logical form* as the argument for a coercing predicate like *enjoy*. We need some way of manipulating the feature structure so that we get the event in the telic quale as the argument of *enjoy*. In feature structure terms, this means stipulating a special lexical rule that transforms the original feature structure into one of the appropriate sort. This now looks very much like the sense transfer function of Nunberg. Alternatively, we can see the feature structure with qualia to be a disjunction of a primary sense (the basic word meaning) and associated senses (the meanings associated with the qualia); predication selects among the associated senses when necessary.

Now that we have brought the structures of GL into familiar territory, we see that they have the familiar problems. The qualia constrain but also pack Nunberg's sense transfer function into the lexical entries for nouns. That strategy brings with it the problems that we've seen already with the sense transfer function. For example, this strategy threatens to get the wrong truth conditions for sentences with coercions and quantification like (3.7) repeated below; we want the coercion to affect the argument of *enjoy* but we don't want the quantifier *many books* to be coerced into the quantifier *many book readings*, even though the coercion based on a type clash can only be identified once we are at the stage of combining the whole DP with the verb.

(3.7) George enjoyed many books last weekend.

Putting the sense transfer function into the noun seems to be precisely the wrong strategy to deal with coercions, since that will shift the domain of quantification in (3.7) on a standard compositional analysis of the DP *many books*. We should rather think of the transfer as doing something to the verb as I argued earlier. Perhaps one could implement something like this structure with special lexical rules for TFSs (although all classic GL approaches are, as far as I know, mute on this subject), but it would look extremely *ad hoc* and be highly non compositional. Hopefully, there is a better alternative!

Pustejovsky and Moravscik exploit the Aristotelian explanatory causes of traditional metaphysics to handle coercion. Aristotle of course takes the $\alpha\iota\tau\iota\alpha$ to be universal features of being. However, it was quickly realized that qualia aren't a universal feature of all types of substances. Types associated with terms that denote natural kinds—e.g., WATER, H_2O, WOOD, GOLD, BIRCH, ELM, PENGUIN, etc., do not plausibly have any associated agentive or telic qualia.[9] Pustejovsky (2001) suggests that the type ARTIFACT can be defined as any type whose associated feature structure has agentive and telic qualia. For such substances their origin or the event of their construction as well as their purpose are pertinent to understanding what they are. This certainly seems plausible. So it should follow that only artifacts give rise to the sort of coercions with the aspectual verbs and verbs like *enjoy* that motivate classic GL. Nevertheless, even this restricted qualia hypothesis runs into trouble, and on two counts. First, it would seem that one can enjoy some

[9] At least not unless the terms "acquire" a use or purpose in a particular context. Pustejovsky (p.c.) noted occurrences of *spoiled water*, which would indicate that the water was spoiled for some purpose like a lab experiment.

natural substances even without turning them into artifacts; climbers can enjoy a cliff; many people can enjoy the mountains or the sea, the beach, the forest, the wilderness, the wide open spaces, and so on. Nevertheless, these would all appear to be objects that are not artifacts. On the other hand, for some artifacts like *door*, the associated agentive and telic qualia don't work with verbs like *enjoy* in the way that the associated qualia of *cigarette* do. What would the telic role of *door* be? One might think that it is to be walked through, or closed. But if I say

(3.29) John enjoyed the door.

this sounds very strange; and if it means anything, it certainly doesn't mean that I enjoyed closing the door or walking through it. In a somewhat similar vein, consider

(3.30) John enjoyed the vegetable garden.

Vegetable gardens are artifacts; they are created by human for the purpose of growing food, as their name implies. Nevertheless, it's hard to get a reading for this according to which John enjoyed growing vegetables in the vegetable garden or that he enjoyed creating the vegetable garden. At least equally available is the much more generic "enjoyed looking at" reading. Such counterexamples abound with artifacts. Kitchens, bedrooms, lamps, forks, knives are all artifacts can all be enjoyed, but the GL predicted reading of, say, *John enjoyed the lamp*—the reading that John enjoyed turning on the lamp or using it to illuminate something—is not there. The list of artifacts that don't behave in the way classic GL predicts is much longer than the list of those that do.[10]

It does not really help, as Pustejovsky and Bouillon (1996) or Godard and Jayez (1993) do, to appeal to aspectual clashes between the demands of the aspectual verbs or *enjoy* and the eventualities contained in the qualia to explain why these examples don't work as they should. The argument is that the qualia for *door*, *lamp*, or *kitchen* involve eventualities that do not meet the demands of the aspectual verbs for some sort of accomplishment as an object argument; therefore, it's no surprise that the qualia don't show up as the values of the coercion. But

[10] Here is another counterexample from Kleiber (1999) with respect to the coercion data. Classic GL should predict that (3.31a,b) have at least one common reading, since one thing one typically does with bulletin boards is read them. Nevertheless, (3.31a) can't be readily interpreted as (3.31b).

(3.31) a. Paul a commencé le tableau d'affichage (Paul began the bulletin board.)

b. Paul a commencé à lire le tableau d'affichage.

notice that the aspectual verbs or *enjoy* are relatively catholic in their requirements on the type of eventuality they can take.

(3.32) a. John enjoyed/began/finished/started swimming/sleeping/drinking (activities)

b. John enjoyed/began/finished/started swimming three laps/crossing the street/drinking three beers (accomplishments)

What these verbs don't take is states or achievements as arguments. One could try to argue that *kitchen*'s, *door*'s and *lamp*'s telic or agentive qualia are all states or state descriptions. But it would appear that the telic quale of *lamp* is to illuminate an area, and this is by the usual aspectual tests an activity. It would appear that the agentive quale of all of these objects is some sort of a creation event, which is an accomplishment. Thus, either the eventualities associated with the telic and agentive qualia of these terms are not the intuitive ones (in which case what use are they?) or the theory fails to make the right predictions for a vast number of ordinary cases.

In any case, the inadequacy of qualia to make the right predictions is not just an observation that applies to the coercions involving aspectual verbs. Consider (3.20c). Water that is moving fast is the desired interpretation of this adjectival modification, but it is one which does not seem to involve reference to any of the inherent qualia of *water*. As Fodor and Lepore (1998) point out and as we'll see in detail later, there are many types besides those given in a classic qualia structure that are relevant to coercion.

More importantly for the purposes of this book is that classic GL is far too inflexible to handle certain examples that show that what eventualities are selected is sensitive to information already present in the context derived from preceding discourse.

(3.33) Paul a commencé de tapisser sa chambre. Marie a commencé la sienne aussi.
(Paul has started to wall paper his room. Mary has started hers too).

These examples pattern with examples like (3.27), in which the particular eventuality selected as the argument of the aspectual verbs is inferred from the discourse context.

There are other ways types get affected as well. In classic GL only the verb is involved in the coercion of its arguments; for instance the aspectual verbs and *enjoy* coerce their object argument to be of event type, and the event type is specified by one of the qualia. But this is empirically incorrect. For instance, the subject of an aspectual verb may determine what the type of its theme or event argument is.

(3.34) a. The janitor has begun (with) the kitchen.
b. The cleaners have started the suits.
c. The exterminator has begun (with) the bedroom.

In each of these there is a coercion to a particular type of event for the object argument of the aspectual verb. Yet it is obvious that the noun phrases in object position do not by themselves supply the requisite events, as the minimal pairs in (3.35a-c) lack the salient readings of (3.34a-c).[11]

(3.35) a. Jan has begun (with) the kitchen.
b. Kim and Pat have started the suits.
c. Sam has begun (with) the bedroom.

We need information about the subject or agent of the aspectual verb to get the preferred readings for (3.34a-c). In classic GL, it's totally unclear how the agent of the verb affects the selection of the event argument. GL cannot explain the effects we have noted. And for those who would argue that these are readings based on "pragmatic" inferences, they are at least as robust as the readings of the coercion cases that motivate classic GL. Secondly, they don't depend on anything in the "context" other than the arguments of the verb! To be sure these inferences are defeasible as well as the ones for the basic qualia. As far as I can tell, there's no argument to say that qualia are part of lexical semantics if these aspects of meaning are not.

There are many instances of coercion that don't fit in very well in the GL framework of qualia. The coercions in (3.36) involve the (defeasible) inference to a particular type of object of *drink*, namely, alcohol. This phenomenon is known as "lexical narrowing."

(3.36) a. Chris likes to drink.
b. Chris drinks all the time.
c. Chris is a heavy drinker.
d. Let's have a drink.

[11] Of course (3.35a-c) have readings on which the agents have begun constructing or fabricating the objects.

Qualia cannot model this sort of coercion, being as they are tied to nouns denoting substances—mainly certain kinds of artifacts. We need to generalize the type system considerably in order to have, for instance, dependent types where verbs that are syntactically intransitive are interpreted as having a particular type of direct object, in order to model such coercion phenomena.

Other coercion phenomena, like the metonymic examples of Kleiber, don't find a proper home in classic GL. In Chinese, there are clear examples of this in the classifier system, where we shift from kinds to members of the kind. ***Laurent example*** One might attempt to assimilate these metonymic coercions to the qualia via the constitutive role, but then the constitutive ceases to have a clear definition—is it truly the material cause or not? In addition it's far fetched to suppose that a qualia structure will list *all the parts* of an artifact as complex as a passenger jet, for example.

Besides the event coercions involving the aspectual verbs, there are coercion effects with other constructions that don't enter into the GL model very easily. Some have argued that perhaps qualia might be used to interpret the genitive (Vikner et al. 199?).The hypothesis is that the qualia tell us what relation there is between the noun phrase or DP with the genitive marking and the head noun. However, a quick survey shows that the number of relations involved between these two expressions is vast indeed.

(3.37) a. Bill's mother

b. Mary's ear

c. Mary's team

d. The girl's car

e. The car's design

f. Mary's cigarette (i.e., the cigarette *smoked by* Mary)

g. Bill's cake (the cake *baked by* Bill or the cake *eaten by* Bill)

h. The wine's bottle

i. A mother's boy

j. The rapist's victims

k. Japan's economy

l. The economy's sluggishness

m. The economy's performance

n. Sunday's meeting

These examples illustrate that the relation between the DP in the genitive and the head noun is often determined by the meaning of the nouns in the construction—sometimes by the argument structure of a relational noun or a deverbal noun.

Nevertheless, the determination of this relation by the syntax/semantics interface is *defeasible*. Information from the discourse context can overrule, for example, the determination of the relation by a relational noun like *mother*:

(3.38) [Context: Picasso and Raphael both painted a mother with a child.] Picasso's mother is bright and splashy —a typical cubist rendition. It makes Raphael's mother look so somber.

With respect to GL, qualia appear to affect the genitive if the head noun is not intrinsically relational—e.g., 1.22 e (telic), 1.22 d (formal), 1.22 f, g (telic or agentive)—cf. Vikner and Jensen. On the other hand, the head noun and the noun in the DP with genitive case can affect the relation (not just the head N), and furthermore, many relations present in the genitive construction don't fall under any qualia — eg. 1.22k, n (spatial or temporal location), 1.22l ("predicational"), 1.22c (set membership), 1.22b (ownership or control). Qualia are at best a partial predictor of what's going on with the genitive construction.

Some might object that this example involves extrananeous elements. The terms 'mother' instead of 'Madonna' suggest a special meaning. But Recanati (2004) offers the following example to show that contextual effects override the semantic import of relational nouns for the treatment of the genitive:

(3.39) a. Ok. I just heard from John and Bill who are working at the old folk's home.

b. We need some supplies right away.

c. John's grandmother needs a wheelchair, and Bill's grandmother a bedpan.

Here we can perfectly well access the interpretation where John and Bill are in charge of certain aged persons. And it overrides the relational interpretation of *grandmother* and *grandfather*.

GL's introduction of qualia structures to analyze coercion looks very different from Nunberg's sense transfer model. But in the end it packs the sense transfer function into the lexical entry for nouns. It can also be thought of as specifying a restricted sense enumeration model, though in different clothing. GL does improve the sense transfer model by putting constraints on what the sense transfer function can be; however, the data on coercion show that the qualia model, although it attempts to put constraints on the mechanism of sense transfer proposed by Nunberg, is insufficiently general and too inflexible to succeed.

What about dot objects, another idea within GL? The underlying idea is again intuitive; copredications indicate that a word may have two independent senses in some sense at the same time. But this intuition is in need of development as well as defense. And once again, we don't have the right formal framework to investigate this intuition. Pustejovsky's introduction of dot objects to model copredication is an additional overload on the overtaxed AVMs with the typed feature structure and unification based framework. Dot objects actually introduce considerable complexities into the type formalism and involve a rather subtle metaphysical analysis. They don't fit in within the typed feature structure framework, as I'll show in detail in the next chapter.

In classic GL (Pustejovsky 1995), one could glean a proposal for exploiting terms of complex • type. The proposal was in effect to take an object of complex type and then "select" one of the constituent types at hand. It was in fact a simple coercion story. Suppose, as Pustejovsky does, that *book* is a complex object of type INFORMATIONAL OBJECT • PHYSICAL OBJECT. By using such projection operations, we can coerce book to the appropriate type for examples like:

(3.40) a. The book is interesting.

b. The book weighs 5 lbs.

On the other hand, it's not at all clear, since it appears that the projection operation is destructive, how one can either account for copredication as in (3.41a) or the sorts of anaphoric coreference in (3.41b), where the pronoun *it* refers back to an informational type object while the predication in the main clause forces *book* to be of type PHYSICAL OBJECT or P for short.

(3.41) a. The book is interesting but very heavy to lug around.

b. John's Mom burned the book on magic before he could master it.

In (3.41a), on the other hand, we need to use the informational type to handle the first predication and the physical type to make the second predication go through. But once we retype *the book* from PHYSICAL OBJECT• INFORMATIONAL OBJECT or P • I to, say, I, then how can we recover the physical type? This problem receives no solution in the classic GL model of dot objects.

However, matters were in fact worse. Suppose $\text{P}\sqcap\text{P}\bullet\text{I} = \bot$; that is, suppose that the meet of the complex type, PHYSICAL OBJECT • INFORMATION and the type PHYSICAL

OBJECT was inconsistent. Then, GL's theory of dot objects went inconsistent. After selection, what happened was that instead of

(3.42) $\text{book}(x), x : \textsc{p} \bullet \textsc{i}$

(where $x : \textsc{p} \bullet \textsc{i}$ is an assignment of the type $\textsc{p} \bullet \textsc{i}$ to x) you got, after one of the selection rules,

(3.43) $\text{book}(x), x : \textsc{p}$

But this was inconsistent with the basic typing rules for *book*. If one decides that the meet of the two types is not inconsistent, it would still produce a funny typing of *book*. It would imply, for instance, that the I part of this retyped $\textsc{p} \bullet \textsc{i}$ object was in fact not there or empty, and that would lead to very strange interpretations. Moreover, we would not be able to handle simple sentences like (6.56). So something has to be done to clear this up.

We have now surveyed both empirical and formal problems with classic GL. It should now be clear that despite its superficial appeal, classic GL is in big trouble. The core claim that meanings are generated during the composition process is at least misleading; worse, the posited complex lexical structures consisting of qualia and dot objects either don't make sense as they stand or fail to make the right predictions on a relatively massive scale. The formalism used is not worked out and seems to have major problems in accounting for the data it was designed to explain. We need to rethink in a major way what is going on with the phenomena classic GL concerned itself with. What about the TFS formalism itself? Should we use it to work out a theory or not? Despite its expressive power, I now feel that TFSs aren't the right formalism to implement ideas about coercion; there's too much going on that distracts us from the basic problems pertaining to predication.[12] Furthermore, much of what Pustejovsky writes about dot objects in his book really doesn't make sense in the typed feature structure formalism. Nothing in the Carpenter framework of typed unification has anything to say about what these objects are. The Carpenter system supplies neither rules nor a semantics for such type constructors as •. We need to go beyond the formalism of typed feature

[12] TFSs are a halfway step between a monostratal theory of composition where propositions are identified with types, as in Martin Löf's Type Theory, and a theory that clearly distinguishes types and traditional logical forms of, say, dynamic higher order logic as in Asher (1993). Standard logical forms are really virtual objects of TFSs; in them only the basic constructors of logical forms are present—the basic predicates which are values of the attribute PRED.

structures to do justice to the intuitions about copredication. We need a richer system of types and a richer model of information flow between the lexicon, syntax and discourse interpretation which includes both semantics and pragmatics.

The main problem with classic GL is that it postulates an elaborate theory of lexical meaning without providing a theory of predication over that implicit in the theory of unification. To get an adequate explanation of the data, we need to do both. We also need to think much harder about the conceptual underpinnings of dual aspect nouns and the types that help us understand them. Classic GL gives us a rich set of problems to think about, but we will have to build a replacement theory pretty much from the ground up.[13]

3.5 Recent Pragmatic Theories of Lexical Meaning

In contrast to classic GL's attempt to locate the mechanisms for coercion and aspect selection within the lexical semantic entry for particular words, there is a pragmatic approach to such phenomena, exemplified on the one hand by Stanley and Szabo's hidden variables approach and on the other by relevance theorists Sperber and Wilson (1986), Recanati (2002) or Carston (2002). Broadly speaking, such approaches attempt to analyze phenomena such as coercion or aspect selection as involving a species of pragmatic reasoning. For Stanley and Szabo the pragmatic reasoning involves finding contextually salient values for the hidden variables; for the relevance theorists, it involves a process of enrichment of logical form.

(3.44) It's raining.

(3.45) I've had breakfast.

By postulating a hidden location or temporal variable, Stanley and Szabo can account for the conveyed content that these sentences typically have in a particular discourse context. For instance, uttering *it's raining* conveys the content that it's

[13] A very useful notion linking classic GL to a type driven theory of predication is the notion of a record, which was developed in the Type Theory of Martin-Löf (Martin-Löf (1980)). See allso Cooper (2004). A record is a function from a type to a collection of attribute value pairs where the values are themselves types. In what follows I will recast the notion of record as a dependent type, that is a function from types to types. Coercion will then be a relatively simple matter of moving from one type to a dependent type, with attendant changes in the logical form to reflect this type shift. I detail this in the following chapters.

raining at some salient place of utterance. *I've had breakfast* conveys that I've had breakfast during some salient interval of time. Discourse context can of course affect salience:

(3.46) a. A: Are you hungry?

b. B: Thanks, I've had a big breakfast.

(3.47) a. Should we try to go to the cliff? It'll take us 4 hours by car.

b. No let's not bother. It's supposed to rain.

In (3.47), for instance, the salient location is that of the cliff. Stanley and Szabo provide, however, no mechanism for determining the salient locations with which to bind the variable they introduce in logical form.[14]

The Stanley Szabo proposal can be adapted to deal with coercions.[15] One would need to postulate for each predicate $\lambda x \phi$ two hidden variables, one higher order relational variable and another of the same general type as x (i.e., of type e or type $e \Rightarrow \mathrm{T}$, etc.). *Enjoy*, which is understood to take an event as a direct object would now look something like this:

(3.48) $\lambda y \lambda e \lambda z$ enjoy$(z, e) \wedge R(e, y)$.

When no coercion is present R could be resolved to identity, but when the direct object of the verb involves denotations to non event-like objects, then R would be resolved to some contextually salient property. So for instance *enjoying playing a soccer game* intuitively involves no coercion, and the variable introduced by the DP that ranges over playing soccer game events would be identified with the event variable that is the direct object argument of *enjoy*. However, *Laura enjoyed a cigarette* would involve a coercion; in this case the variable introduced by the DP ranging over cigarettes would be related via some contextually salient relation R like smoking to get the logical form:

(3.49) $\exists y \exists e($ enjoy$(l, e) \wedge$ smoke$(e, y) \wedge$ cigarette$(y))$

This isn't quite right, as one can see that one has lost sight that *enjoy* is a control verb and so the relation variable involved here must involve an extra argument coindexed somehow with the subject argument of *enjoy*. Additional stipulations,

[14] For examples like (3.47), one could appeal to the formulation of relevance in **?**to define the salient binder for the variable.

[15] In effect Egg (2003) develops such a proposal.

however, will fix the problem; we need some additional constraints on the hidden variable for control verbs.

The Stanley-Szabo approach has nothing to say about when coercion takes place. The typed approach I favor does—when there is a type mismatch, there will be an attempted type adjustment, which, if it succeeds, will in turn lead to an accommodation of the type presupposition. Without this sort of type driven restriction, the Stanley-Szabo approach predicts all sorts of interpretations for *enjoyed playing the soccer game* that do not fit the facts. It predicts the possibility of all sorts of strange relations between this event and the event that is actually enjoyed—like for, instance, that the event enjoyed was different from the one that involves the playing of the soccer game. But this variation is not attested at all, regardless of context. Thus, the account is too weak and predicts coercions where there are none, unless we add to it a type driven theory of predication.

On the other hand, it also predicts that certain coercions should succeed when they do not. Consider the following example due to Karen Vespoor.

(3.50) a. Last night, my goat went crazy and ate everything in the house.
b. At 10 pm, he started in on your book.
b”. At 10 pm, he began to eat your book.
b”. #At 10 pm he began your book.

(3.50b”) is plainly bad even though we are primed in the context to understand that the eventuality to be coerced to is an eating event. The salient value for the hidden variable is clearly salient. While *begin, start* and *finish* all coerce some arguments —e.g., *start a cigarette, begin a cigarette, finish a cigarette*, they are not all equally happy in their coercive capacity with other direct objects. This has to do obviously with the meanings of the individual words, not with a general pragmatic mechanism. The Stanley-Szabo approach attempts to explain coercion at the wrong linguistic level.

Further, the Stanley-Szabo account does not have an explanation of the difference between dual aspect nouns and standard coercions. It has no account of the complex cases of copredication that motivated classic GL's introduction of “dot objects” or Cruse's idea of perspectives. As we shall see, a type driven approach can develop these intuitions into a sophisticated theory.

The Stanley-Szabo view is not the only pragmatic approach to lexical meaning. Relevance theorists (Sperber and Wilson (1986), Recanati (2004, 1993)) attempt to capture these inferences by appealing to a process of free enrichment.

Interpreters will add to the content of (3.44) and (3.45) in order to make the information conveyed maximally relevant, according to their principle of relevance. They take free enrichment to be a local process in which litterally encoded meanings are replaced by pragmatically enriched ones to form the proposition that expresses the conveyed truth conditions of a sentence. So for example in Nunberg's *I'm parked out back*, the literal meaning of *I* is replaced by the enriched concept *the car I'm driving*.

Relevance theorists point to phenomena like lexical narrowing or loose talk as support for their theory, and indeed it's unclear how a Stanley and Szabo hidden variable approach would handle such examples. Similar examples resembling lexical narrowing involve the specialization of a broad general meaning to a particular argument.

(3.51) a. The ironing board is flat.
b. My neighborhood is flat.
c. My country is flat.

It is indeed true that each of these uses of flat specializes a general meaning, as we can see from our ellipsis test.

(3.52) This ironing board is flat and my country is too.

However, these cases can fit into the Stanley Szabo approach, if we assume that *flat* involves a hidden degree variable (**?**); the subject of the predication could in principle dictate a particular degree of flatness relevant to the subject's type. In fact, given Kennedy's work, it appears relatively straightforward to give a *semantic* account of the differences in (3.51). Nevertheless, Relevance theorists can also attack examples like (3.51) using the principle of relevance.

There are a number of problems with the proposal of free enrichment. First of all, free enrichment, as it's usually formulated, will predict the wrong truth conditions for many of the coercion cases, just as Sag (1981)'s precisification of the sense transfer view does. Furthermore, the definition of relevance is difficult to make predictions with.[16] The formulation of the principle of relevance requires an appeal to unknown mechanisms of inference and processing cost. Because of this, the relevance theoretic approach fails to explain why some attempted predications don't go through, and why predication is often constrained. For instance, we should expect on the enrichment view that we could always make sense of

[16] See Asher and Lascarides (2003) or **?**for a more detailed look at relevance theory.

predications in which there are, from my perspective, type clashes. There should be little difference between the classic coercion cases and cases where no coercion is possible such as some of the examples in (1.1), repeated below:

(1.1c) The number two is red.

The pragmatic approaches also fail to say anything relevant about the cases of coercion like *John enjoyed his glass of wine* or *Mary enjoyed her cigarette* in out of the blue contexts. On the free enrichment approach or the hidden variable approach, we should expect either no definite answer in out of the blue contexts or we should expect variation. However, speakers are remarkably constant in their interpretation of such sentences.

In principle, a pragmatic approach could provide an account of the phenomena that motivate a type driven theory of predication. However, such theories have difficulty in saying anything about the interaction between semantics, predication and the pragmatic principles they use to analyze coercion. Consider

(3.53) a. Isabelle has a temperature.

b. Check the temperature of the corpse in the morgue.

c. Every physical object has a temperature.

(3.53a) conveys that Isabelle has a fever, an abnormally high temperature, whereas of course (3.53b,c) do not. Why should pragmatic processes of enrichment be blocked in these cases? One might claim that only living animals can have fevers or abnormally high temperatures. I would agree but that is to say that information associated with the types of the terms in the predication affects the logical form.

Chapter 4

A Context Sensitive Model of Word Meaning

We have seen that previous accounts in general do not provide formally or materially adequate accounts of coercion and copredication. All of these accounts attempt to shift the meaning of various terms to account for the phenomena. The general story about coercion involves a predication in which the selectional restrictions of the predicate do not combine with the type of the arguments. The solution is to shift the meaning of the arguments or the predicate so that the predication succeeds. Nunberg left the shifting mechanism completely unconstrained, while classic GL makes the shifting mechanism part of the lexical entries of nouns. The former has little explanatory force, while the latter is too rigid to account for the data and appears to get the wrong truth conditions for coercions involving quantification. More generally none of these theories can account for how previous discourse context can affect predication. It appears uncontestable that coercions and meaning shifts are context sensitive. So we need a particular type of lexical semantics, which I'll call a *contextualist model of type driven lexical semantics*.

On the other hand, one might start with ordinary, simple meaning representations for each term, as in classical semantic theories like Montague Grammar and unlike classic GL, but suppose that when putting terms together in a predication, extra contributions to logical form can arise when there is a type clash between predicate and argument and a type presupposition must be accommodated. This means the predication must involve more than the simple application of a property to an object or a relation to its terms. It must involve accommodation mechanisms that will introduce material into logical form to make the predication succeed.

The theory of predication and lexical meaning that I will now introduce has this feature. In addition, I will make these mechanisms sensitive not only to the local predication—the application of the predicate to its arguments—but also to the discourse context in which this predication takes place. In other words, I propose a context-sensitive theory of predication.

My proposal brings the theory of predication into line with a general tendency in semantics and pragmatics that has been developed over the last quarter century and that finds expression in a dynamic framework. Dynamic semantics has proved extremely useful in analysing the semantics of anaphoric elements in language like pronouns and tenses and of expressions that generate presuppositions that must be integrated into the given discourse context (e.g., Kamp (1981), Heim 1982, Heim 1983, Asher 1986, 1987, van der Sandt (1992), Asher (1993), Kamp and Reyle (1993), Groenendijk and Stokhof (1991), Veltman (1996), Beaver 1998), and extensions of this idea to include richer ideas of discourse structure and pragmatics have also proven useful (Asher and Lascarides (2003)).

The general idea of dynamic semantics is to treat the meaning of a formula as a relation between information states, an input information state and the output information state. The input information state represents the content of the discourse context to date while the output information state represents the content of the previous discourse context integrated with the content of the formula. The interpretation of a discourse involves the relational composition of constituent sentences' relational meanings. In dynamic semantics for natural languages, as well as in the dynamic semantics for programming languages, the interpretation of a formula can either function as a test on the input context or can transform the context. So for example *John is sleeping* in dynamic semantics would yield a formula that functions as a test on the input context, which we can think of as a set of elements of evaluation. If an element of evaluation verifies the proposition that John is sleeping then it becomes part of the output context; if it does not verify the prosposition, it does not become part of the output context. Some sentences, however, in particular those containing indefinite noun phrases but also other elements like modals, output a context that is distinct from the input one. They transform elements of the input context; in particular they *reset* or *extend* elements of the context to reflect the information they convey. For instance, an indefinite has the action of resetting an assignment that is part of the point of evaluation for formulas, as in Tarskian semantics, or extending the assignment with a value to a ”fresh” variable But this new value variable pair becomes part of the output context (unlike Tarskian semantics).

A context sensitive theory of predication also involves these kinds of actions, but not on the standard information contexts of dynamic semantics but rather on a *type context*. The association of types with terms and variables given in the lexicon provide the basic typing context, in the present theory as well as in the standard theory of the typed lambda calculus. What is different here is that this typing context may evolve as discourse proceeds. To be more specific, lexical entries contain semantic information of two kinds, in keeping with the two level lexical theory I argued for earlier. The first is a λ-term with a model theoretic interpretation, while the second is a typing context. A *typing context* for a term t determines an assignment of types to all subterms of t. It is the types that are assigned to the various variables in the λ-term that will affect how it combines with other terms in the composition of a logical form. A *composition logic* determines how the types of lambda terms can combine.

Normal predication involves a test on the types of the expressions involved in the predication. If the type of the functor and its argument agree, then the predication is licensed and the construction of logical form can proceed via application. Application changes the typing context: once application has taken place some lambda expressions have been combined with their arguments and so are no longer around in the type context. On the other hand, the basic terms (variables) have their types unchanged in ordinary predication. Some predications, however, those involved in coercion and copredication, also introduce new terms with the appropriate types. These are in effect *update* operations, very similar to the ones that various forms of dynamic semantics posits for indefinites. It is important to note that the resetting operations operate on types of variables in logical forms, *not* on lexical entries. It will be crucial in what follows that the lexical entries remain constant. What is manipulated by the type composition logic is how these entries combine at the level of logical form. Thus, my theory combines certain elements advocated by Fodor and Lepore with a context sensitive theory of predication. I will show that this theory has advantages over other accounts.

4.1 Type Constructors

An important question is what sort of general type constructors we need for investigating the meaning shifts evident in coercion and copredication. The typed lambda calculus has built into it the distinction between functional types and simple types. But we need other complex types to model coercion and copredication. We need a complex type to deal with coercion that in fact maps us from a term of one type to a term of an associated type—this is known as a *dependent* type. Qualia are just a special kind of dependent type, a function from types to types of eventualities. The semantics of dependent types will constrain what sort of shifts are possible.

Besides coercions there are aspect selctions to consider. Do we need a separate type constructor to model the predication of properties to selected aspects, or more properly speaking to model the behavior dual aspect nouns? Recall that the copredication data suggest that objects like books or lunches have two distinct aspects. Books, for instance, appear to have both a ”physical” aspect and an informational ”aspect”; which is selected in a predication depends on the type restrictions imposed by the predicate on its arguments. Very roughly, a word like *book* may refer to a type with a “dual” nature—two conceptualizations, if you will, that are equally ”true of” or ”faithful to” the object. And often these two natures are in some sense also incompatible, as the copredications in which two predicates with incompatible typing requirements apply to the same term. There are many examples of this:

(4.1) a. Mary picked up and mastered three books on mathematics. [*physical object* and *informational content*]

b. L'homme n'est qu'un roseau mais un roseau pensant. (Pascal) [*physical object* and *thinking agent*]

c. That is a lump of bronze but also Bernini's most famous famous statue [*portion of matter* and *artifact*

d. Le prix Goncourt is 10000 euros and a great honor not accorded every year. [*amount of money* and *prize* and *winner–person*)

e. The lecture (interview, speech) lasted an hour and was very interesting. (*event* and *information*)

f. The exam was written in 10 minutes and was only 2 pages long but took 3 hours to complete [*informational object* and *physical object* and *event*]

g. The promise was made but impossible to keep. speech act (event), proposition

h. Linnaeus' classification of the species took 25 years and contains 12,100 species. (process and result)

i. ??John's belief is false but persists

j. The house contains some lovely furniture and is just around the corner. [*physical object* and *location*]

k. Most cities that vote democratic passed anti-smoking legislation last year. [*population* and *legislative entity*]

l. Lunch was delicious but took forever. [*food* and *event*]

m. The apple has a funny color but is delicious. [*food* and *surface* or *skin*]

These copredications can happen at the level of speech acts as well.

(4.2) Could you pass the salt please?

Asher and Lascarides (2001) argue that *please* types its sentential argument as a request, while the syntax of (4.2) types the clause as a question, and this forces the discourse constituent introduced by (4.2) to have a complex type QUESTION • REQUEST. Asher and Reese (2005) and Reese and Asher (2006) extend this notion to biased questions which are both assertions and questions.

(4.3) Does John do a damn thing around the house?

They adduce a wide variety of tests to show that such examples also function as complex speech acts.

We've seen that there is some evidence to take such dual aspect nouns as supporting ambiguity according to the contradiction test. However, dual aspect nouns and their aspects are certainly distinct from accidentally polysemous nouns like *bank* which don't support the copredication test. And there seems to be a strong intuition that dual aspect nouns should have a type in which both constituent types, the types of the aspects, are in some sense present. The copresence of these two types shows up in the use of different individuation criteria for the same objects, as we saw in the last chapter.

Nevertheless, postulating a new complex type for the dual aspect nouns is controversial. The traditional story about the data involving dual aspect nouns is that there is a sort of coercion going on. For (4.1a) the story postulates that books are physical objects but that they have associated with them an informational content. Sometimes predicates apply to books properly speaking while some predicates apply to the associated informational content. These cases, many argue, are examples of sense transfer. Why should we need anything else. One objection to this proposal, however, is that sense transfer functions don't tell us that there's anything special about lunches, books and other elements that seem to have two constitutive types at once. Real books have to exist in some physical form (even e books), but they also have to have some information content. The physical and informational aspects of books bear a very different relation to each other from the weak association that holds between me and a vehicle I am driving. If sense transfer functions capture the latter, they fail to explain what's special about books, lunches and other objects that I claim are of • type.

In general it's not easy to tell when a term is a dual aspect term. The linguistic test for such terms has two parts: first they must support copredications; secondly, predications that pick out a particular aspect of the objects associated with these terms must be able to affect the way such objects are counted or individuated. This second property of dual aspect terms is not shared by terms in standard coercive contexts. Consider the sentences below involving *book*, *question* and *newspaper*, all of which have been characterized as denoting objects with multiple aspects in the literature.

(4.4) a. The student *mastered* every math book in the library.

b. The student *carried* off every math book in the library.

(4.5) a. The teacher *answered* every student's question.

b. The teacher *repeated* every student's question.

(4.6) a. John bought two newspapers yesterday. (physical object reading)

b. Rupert Murdoch bought two newspapers yesterday. (institution reading)

The quantification over books in (4.4) is sensitive in one case to its informational aspect, and in the other to its physical aspect. In (4.4a), we simply quantify over all informationally distinct individuals without reference to the instantiations of these informational units; it is not necessary, for example, for the student to have read every distinct copy of every book in the library. In (4.4b), however, every physical individual must have been taken in order the sentence to be true. Similar remarks hold for the distinction in (4.5b): an answer to the same question posed on multiple occasions will count as an answer to each question; this is not the case with the act of repeating the question, however, since this refers to copying the speech act rather than providing the informational content of the answer.

Now contrast these examples to cases where coerced type shifts have been postulated, either of the Nunberg variety or the classic GL kind:

(4.7) a. Everyone is waiting to go home.

b. Everyone is parked out back.

(4.8) a. John enjoyed many cigarettes last night

b. John enjoyed no cigarettes last night

In neither (4.7a) nor (4.7b) do we quantify over cars. (4.7b) says something like *everyone came in a car that's parked out back*. And in (4.8a) while the object of *enjoy* is plausibly an event, it's the event of smoking many cigarettes. This is even clearer in (4.8b), where the only possibile reading is that there were no cigarettes such that John enjoyed some activity with them, presumably smoking them. In these coercions, the whole noun phrase or DP doesn't change its denotation, but rather the argument of the verb becomes some meaning related to the noun phrase. This would suggest that the logical form associated with the noun phrase in a typical coercion doesn't really shift even after the coercion takes place . In contrast, the logical form of the whole noun phrase involving dual aspect nouns does seem to shift when we predicate a property of an aspect of an object. So we must conclude that predications involving aspects and predications involving coercions are not the same thing.

One could attempt to use Nunberg's transfer functions on different terms to simulate this behavior. To handle coercion, transfer functions could apply to arguments of verbs; whereas to handle aspect selection, transfer functions would have to apply to an entire DP to get the counting arguments right. But at that point, we will not be able to account for anaphoric dependencies on *other* aspects of the object. This was a problem we saw with classic GL's account of dual aspect nouns. Consider for instance,

(4.9) John's mother burned the book on magic before he could master it.

If we apply transfer to *the book on magic* to make it an object of physical type, then we don't have an antecedent for the pronoun whose value has to be something informational.

If the use of transfer functions on terms does not look promising, then alternatively we can exploit standard approaches to ambiguity and postulate two senses of book: the basic physical book and ”informational books” which have the following logical from:

(4.10) i-books: $\lambda x \exists y(\text{book}(y) \wedge \text{info}(y) = x)$,
with the types $x : \textsc{i}$ and $y : \textsc{p}$

Although standard tests for ambiguity indicate sometimes that dual aspect nouns are ambiguous, it would be wrong to treat all ambiguous terms in the same way. We've already seen that accidental and logical polysemy differ in several linguistically discernible ways—for instance, the copredication and anaphora tests. The decision to have i-books and books is strange from a philosophical point of view. The point about objects like books or lunches is that a book is *both* a physical object and an informational object in some sense. *Pride and Prejudice* on my shelf to the right of my desk wouldn't be what it is without the information content of that book. But it is also a concrete physical object. This is why we can count or individuate books either according to their physical aspects or their informational aspects and why we can quantify over them ”information-wise” or ”physical-wise”.

The distinction between i-books and books corresponds perhaps to a distinction between types and tokens. Barbara Partee has suggested (p.c.) that one might handle the quantificational ambiguity seen above with *read* and *carry off* by treating the entire phenomenon as an instance of the type/token distinction. According to this suggestion, (4.4a) makes reference to the type while (4.4b) refers to the token.

There appear to be several problems with this solution. Simply reducing the above phenomenon to a an ambiguity between i-books and physical books or ’p-books’ or types and tokens does not solve the problem of how the copredication works; if the suggestion were right, we could envision using that distinction along with our $\bullet$ type apparatus in the analysis, but without the latter, it is not clear what the analysis would be. In copredications involving the informational and physical aspects of books, we would have to say that one simultaneously accesses a type or i-book as well as a token or p-book But we would still in that case have to have a relational analysis of the token to the type and we would need to complicate our analysis of copredication in the same ways as we are presently envisioning with relational tropes. Furthermore, there are cases where reference seems to be made to more objects than are available under a simple type/token analysis. For example, in (4.11b), quantification is over informational tokens which are distinct from the actual physical object tokens that would be available.

(4.11) a. John hid every Beethoven 5th Concerto score in the library.

b. John mastered every Beethoven 5th Concerto score in the library.

Hence, for an object of complex type, if there are type and token interpretations available for each component type of the dot, then the underlying typing is more complex than originally countenanced.

I can sharpen this observation. There are kinds of informational objects and kinds of physical objects, but we are not predicating or quantifying over kinds of objects in (4.4a); we’re quantifying over particular informational objects. Perhaps the examples in (4.12) will make this clearer.

(4.12) a. John has stolen every book there is.

b. Frances has grown every wildflower in Texas.

While there are (improbable) interpretations exploiting the token reading of the quantified expression in each example above, the type interpretation is more felicitous. However, the interpretation of the generalized quantifier in (4.12a) makes clear that the type reading of *every book* is distinct from the informational content interpretation of the dot object in sentence (4.4). That is, the verb *steal* selects for physical instantiations of kinds of books. This is the true “kind interpretation”, but it is distinct from that seen with the exploitation of part of a dot object from the verb *master* in (4.4).

Another traditional way to analyse dual aspect nouns is to claim that they are ambiguous. However, copredications like (4.1) cannot be handled by making the dual aspect nouns ambiguous with respect to their type. Which type for *book* do we choose in dealing (4.1)? Neither one on its own will do. So we must also postulate either sense transfers or ambiguities among the predicates of such objects. For instance we could make *master* ambiguous between its normal sense where it requires an object of informational type and one where it takes an object of physical type as in (4.13):

(4.13) p-master: $\lambda x \lambda y \exists z(\text{master}(y, z) \wedge \text{info}(x) = z$
with x : $\textsc{p}$ and z : $\textsc{i}$

Alternatively, we could have an informational *pick up*.[1] We now have multiplied ambiguities many fold in the theory. We have i-books and normal books (or vice versa books as basic informational objects and p-books) and we also have normal mastering and p-mastering, or normal pickings-up and i-picking-up. In fact we can't even choose between these alternatives; the dual nature of books says both are equally important. So we have in fact a ”second order” ambiguity in this analysis. A great many *ad hoc* decisions have to made now in lexical semantics. Books and lunches have only two constitutive aspects, but what about cities and so on? To be sure these are not ”expensive” ambiguities; it's easy to understand how one sense is related to the other, provided we spell out the function *info* appropriately. But the language doesn't seem to have these ambiguities itself.

Worse, the account predicts that intuitively false sentences are true. Consider,

(4.14) John burned a book and then read it.

Let's suppose that the pronoun picks up an i-book sense. Let us suppose that *master* takes as a direct object something of informational type as before. This then forces us to assume that *book* is to be understood in the ”i-book” sense. Suppose now that burn requires as a direct object something of physical type. So now we must postulate either a shifting in the type of *burn* or from an i-book back to the normal sense of book. The trouble is many physical books may be instantiations of the same i-book. So this theory predicts a reading of (4.14) where John burns a copy of some i-book, a copy of which he had antecedently mastered. Importantly the physical book that John burns need not be the same physical book introduced by the quantifer *a book*. Perhaps there are other ways of resolving the ambiguities to get the right truth conditions for (4.14) but there doesn't seem to be any principled way of ruling out incorrect readings like the one I've just described.

[1] This sense does indeed seem to exist, but it's not the one at issue in the example (4.1).

The traditional theory has to postulate that books are ambiguous between physical books and i-books, and it also has to postulate an ambiguity with respect to predicates of books. Thus, there is no principled way to rule out these readings.

This approach to copredication not only makes the wrong predictions but it also fails to explain certain fundamental differences between terms. It doesn't explain why copredications with certain ambiguous words works but not with others. What's different about the ambiguities? The association between cars and individuals who drive them is at least as clear as the informational relation between physical objects and informational objects but we can't say

(4.15) I"m parked out back and am an old Volvo.

anymore than we can say *the bank specializes in IPOs and has eroded because of the recent floods*. Researchers have often sought to label the distinction between type shifting in the semantics and type shifting in the pragmatics. The Nunberg examples are often taken to be examples of a pragmatic shift, whereas the sort of type shifting or ambiguities postulated for books and lunches are semantic. Nevertheless, this can't be the whole story, as there are semantically ambiguous words like *bank* that function equally badly with coopredication. Metaphysically, there is just a big difference between objects of • type and objects that are not. Books are equally physical and informational objects, but there is no sense in which I am equally a person and my car (unless you live in LA). Such big metaphysical differences should be and are reflected in our conceptual scheme, that is, in the system of types. Thus, I will model dual aspect objects with the complex type •.

Beyond • and dependent types, it is at least tempting to consider others. How should we think of accidental polysemy? Asher and Denis (2005) argue that accidentally polysemous terms can be typed in terms of a disjunctive type—each disjunct gives a possible sense of the word. Ambiguity resolution can then proceed via disjunction elimination during the composition process. Finally, I'll introduce a type for constructor to handle presuppositions and "postconditions". Rules for type shifting and for transferring the information from the level of types to the level of formulas will make clear that what drives the explanation of the phenomena under the rubric of logical polysemy as well as of accidental polysemy are type mismatch and then type exploitation or introduction to correct the mismatch. I will show how to account for the data that motivated GL's account of qualia and the data that show that it is limited. By integrating rules about how discourse context can guide the type shifting I will develop a context sensitive, type driven theory of predication.

4.2 An Abstract Model for Types

I've argued that the standard set theoretic model for types is not adequate. So what else can I propose? An abstract model of complex types comes from Category Theory, which has been long recognized as an important modelling source for type theory.[2] In particular the notion of a closed cartesian category has the resources within which to model such complex types Lambek and Scott (1986).

- A *graph* consists of a set of objects (points) and a set of arrows between points (edges). In addition, there are two maps from Arrows to Objects, one labeled *Domain* and the other *Co-Domain*.

- A *deductive system* $\mathcal{D}$ is a graph in which for each object A in $\mathcal{D}$, there an arrow 1_A, which is the identity map on A, and for each pair of arrows $f : A \to B$ and $g : B \to C$, $gf : A \to C$, the composition of g with f, is defined. Objects can be identified with formulas and arrows with proofs. The closure under composition gives us a rule of inference.

- A *category* is a deductive system where $\forall f : A \to B, g :\to C$ and $h : C \to D$:

  - $f 1_A = f = 1_B f$
  - $hg(f) = h(gf)$

Our system of types requires a closure under certain operations, product and function space. However, category theory allows such constructions over categories to form new categories.

- Let $\mathcal{A}$ and $\mathcal{B}$ be categories. Then $\mathcal{A} \times \mathcal{B}$ is also a category, where the objects are pairs (a, b) a an object of $\mathcal{A}$ and b an object of $\mathcal{B}$ and arrows are also pairs$(f, g) : (a, b) \to (c, d)$ where $f : a \to c$ is an arrow of $\mathcal{A}$ and $g : b \to d$ is an arrow of $\mathcal{B}$.

- Let $\mathcal{A}$ and $\mathcal{B}$ be categories. Then $\mathcal{B}^{\mathcal{A}}$ is also a category. The objects are *functors* $F : \mathcal{A} \to \mathcal{B}$ and the arrows are the *natural transformations* on the functors.

- A *functor* $F : \mathcal{A} \to \mathcal{B}$ is defined as a map from objects of $\mathcal{A}$ to objects of $\mathcal{B}$ and from arrows of $\mathcal{A}$ to arrows of $\mathcal{B}$ such that for objects a, a' of $\mathcal{A}$

[2] For a nice introduction to category theory and types, see Asperti and Longo (1991).

and an arrow $f : a \rightarrow a'$, $F(f) : F(a) \rightarrow F(a')$ and $F(1_a) = 1_{F(a)}$ and $F(gf) = F(g)F(f)$. Note that morphisms also compose (i.e. for $F : \mathcal{A} \rightarrow \mathcal{B}$ and $G : \mathcal{B} \rightarrow C$, $(GF)(a) = G(F(a)$ and $(GF)(f) = G(F(f))$,

- Given functors $F, G : \mathcal{A} \rightarrow \mathcal{B}$, a *natural transformation* $t : F \rightarrow G$ is a family of arrows $t(a) : F \rightarrow G$ in $\mathcal{B}$ one arrow for each object a in $\mathcal{A}$ such that the following square commutes for all arrows $f : a \rightarrow b$ in $\mathcal{A}$–i.e., $G(f)t(a) = t(b)F(f)$
  $t(a) : F(a) \rightarrow G(a)$
  $G(f) : G(a) \rightarrow G(b)$
  $F(f) : F(a) \rightarrow F(b)$
  $t(b) : F(b) \rightarrow G(b)$

Now if we allow arbitrary finite cross products and also function space operations, then we need to close some family of basic categories under these operatioins. This corresponds (roughly) to the notion of a *Cartesian Closed Category*.

What is the natural interpretation of the product and function space operations? Well, in terms of logic, we have conjunction and arrow corresponding to these type operations. Note that for products as well as functors that are constructed from more basic objects, certain projection operations are defined. More specifically,

Fact 1 *given objects $X, Y \in C$, a* product *$X \times Y$ is an object $W \in C$ along with two maps $\pi_1 : W \rightarrow X$ and $\pi_2 : W \rightarrow Y$ such that for every object $Z \in C$ and maps $f : Z \rightarrow X$ and $g : Z \rightarrow Y$, there exists a unique map $u : Z \rightarrow X \times Y$ such that $\pi_1 \circ u = f$ and $\pi_2 \circ u = g$.*

A similar fact holds for the functor construction. Given projection functions from products to their constituents, we can model both the introduction and elimination rules for $\wedge$:

- Let π_1 be a projection function from product or pair categories to first elements of the pair such that $\pi_1(a, b) = a$. We can think of π_1 as an inference rule on formulas that correspond to types: $A \wedge B \rightarrow^{\pi_1(a,b)} A$. Similarly for the second argument.

- Suppose $f : c \rightarrow a$ and $g : c \rightarrow b$, then $(f, g) : c \rightarrow (a, b)$; this corresponds to the rule on formulas $C \rightarrow A, C \rightarrow B \vdash C \rightarrow (A \wedge B)$.

- $\Rightarrow$, already familiar from the functionally typed λ calculus, provides corresponding rules of *modus ponens* and conditional proof.

In fact one can show:

Fact 2 *The positive implicational fragment of intuitionistic logic (same as classical logic) has a model in a cartesian closed category (CCC) (e.g. Lambek and Scott (1986))*

We can extend a CCC to close types under $\cup$. We thus consider the category of *algebras*. $\cup$ represents the operation of disjunction on types. Disjunctive elements of the form $a \vee b$ will be useful in representing true lexical ambiguities such as that of *bank*.

The model provided by algebras is, however, extremely abstract. One could interpret the product construction for complex types in several different ways, some of which are more sensible conceptually than others. The fact from Scott and Lambek interprets products of types as an intersective type. But this is not the only interpretation of products. A proper interpretation of complex types within a category theoretic setting must take into account not only the constituents of the complex type but also: (i) how the exploitation of complex types affects logical forms and (ii) how complex types and their constituents are related in the type hierarchy. I will argue below when I discuss the complex type $\bullet$ that these two factors preclude a simple intersective model, for instance, of $\bullet$. Because the product construction is so general, I will end up modelling several complex types as product types—for example lexical presuppositions and assertions will form another sort of product type besides $\bullet$. The differences will be in how the various complex types interact with logical forms.

Chapter 5

The Complex Type •

A theory that invokes complex types has to say how these types get used in the analysis of copredication and coercion. And to do that we need to do is to provide a way of manipulating complex types and deciding what information flows from them—i.e. their effects on logical form and truth conditions. Underlying this system of type manipulation must be a semantics or model of what such complex types are. Given the nature of the type system and the separation between logical forms and types, two questions need to be answered: (1) How does the semantic conception of such complex types interact with the type hiearchy? And (2) how does information about types interact with logical form? An answer to (1) will affect the model of complex types. For (2) the answer is also complicated but it stands to reason that there must be some interaction; if we shift types, we must also change the predicative information associated with that argument in logical form. Let's see how these questions play out with the complex type •, which has received the most scrutiny formally Asher and Pustejovsky (2004), Cooper (2005), and which has considerable metaphysical ramifications.

5.1 A Concrete Model of • Types: Intersective Types

The basic idea behind • types is this. A term of • type $\alpha \bullet \beta$ has at least two aspects, one of which is of type α and another of which is of type β. P Copredications where the predications have different typing restrictions may predicate of both aspects at the same time. The data on copredication strongly suggests that at least some terms have such a complex type as a matter of their lexical meaning. Of course this is not surprising when one type is a subtype of the other or even if they have a common meet in the type hierarchy. But the interesting cases of copredication are those where the types involved do not have a common meet. It is the latter case that • types are supposed to model. I will spend some time investigating what • types say about their inhabitants, as this will give us a much better understanding of the • operation.

There is one model for • types already available to Classic GL. Within the typed feature structure framework, types are modelled as propositions true at nodes; therefore, complex types have no natural interpretation, if the complex type forming constructor has no apparent propositional operator correlate. The only possible propositional operator correlate is conjunction or (if one thinks of propositions in terms of sets of indices) intersection. This would mean that • types would in effect be intersective or conjunctive types (Coppo and Dezani 1978), with the following axiom governing conjunctive types of the form $\alpha \sqcap \beta$:

- **Conjunctive Types Axiom**:
  $x : \alpha \sqcap \beta$ iff $x : \alpha \wedge x : \beta$

This model of • adopts the following conjunctive types hypothesis (CTH). Let's assume for simplicity, as most work in lexical semantics does, that the type hierarchy forms a complete lattice in which greatest lower bounds are assured to exist. Then:

- **Conjunctive Types Hypothesis**:
  $\alpha \bullet \beta := \alpha \sqcap \beta = glb\{\alpha,\beta\}$

To see how CTH fares, let's first see how the outlines of an account of copredication might go. Consider once again a classic copredication like (5.1):

(5.1) The book is interesting but very heavy to lug around.

Without going into too many details at this point, let's imagine that the coordinated adjectival phrases place a • type requirement on the subject of the verb phrase—in this case, the subject must be of type P • I. Now if we assume, again without going into details, that the type assigned to *book*, which is also by hypothesis P • I, percolates up to the complete noun phrase as a whole, then the types match and the copredication succeeds.

The problem CTH faces, however, is fundamental and stems from the basic axiom for conjunctive types. In particular for certain cases of copredication, it is just not the case that for $x : \alpha\bullet\beta$ that $x : \alpha$ and $x : \beta$, because α and β may be in fact inconsistent types. Two types α and β are inconsistent just in case assigning these types to any term *t* would make the denotation of *t* have incompatible properties. But hence according to CTH, $\alpha \bullet \beta = glb\{\alpha, \beta\} = \perp$, where $\perp$ is the absurd type. For the complex types of speech acts, for instance, we clearly cannot associate with the object a conjunctive or intersective type according to which it is *both*, say, a question and an assertion—that would amount to saying that its semantic content is both a set of propositions (at least two) and a single proposition. A conjunctive type $\sigma \sqcap \tau$ holds of a term *t* iff $t\colon\sigma$ and $t\colon\tau$; in effect a conjunctive type says that objects that have such a type lie at the intersection of the objects that have the first type and objects that have the second type. So in the case of a complex speech act of the type QUESTION • REQUEST would be empty.

Nor does CTH fare any better with lexical terms of • type. In the types proposed for the lexicon, this is much less clear. Let's consider the case of *lunch* again. *lunch* supports copredications and is of the complex type that consists of a meal, an object with parts (courses, dishes), and an event, the eating of the meal. Now it would appear that events and physical objects have different, even incompatible properties, though of course some philosophers, the so called four dimensionalists, don't shrink from identifying all objects with event-like four dimensional objects that course through time. But objects, from our commonsense point of view, perdure through time while events have a duration; objects are wholly present at each moment in time while events have temporal parts. At each temporal interval during which I'm writing this book it would be crazy to think that the entire event of writing this book were present—that would make writing this a good deal easier than it is! So it looks as though there is a strong reason to think that events and physical objects are disjoint types of entities.

I've already presented linguistic evidence that events and physical objects are distinct types. But some of that evidence points to a stronger conclusion—namely, that events and physical objects are incompatible types. To repeat some data from the first section, events take temporal and manner modification of a certain kind whereas objects cannot without producing a semantic anomaly.

(1.6) a. The tree grew slowly.

b. #The tree was slow.

c. The tree was slow in growing.

One could be so bold as to say that *nothing* is wholly an event and wholly an object. The *intersection* of the types EVENT and PHYSICAL OBJECT is empty. But if lunches, as I've argued, have the type PHYSICAL OBJECT • EVENT, then • is not to be understood as the intersection of two types or their meet.

Similarly, one can argue that the population of a city is not at all identical to the legislative body that runs the city; a city like Austin has a million inhabitants but its legislative body consists of the mayor and the city council which has eight people in it. More generally, legislative bodies are a strict subset of the citizenry, unless we have a direct democracy where all the inhabitants constitute the legislative body.

Pustejovsky uses BOOK as his favorite example of a • type. Books receive the type here of PHYSICAL-OBJECT • INFORMATIONAL-OBJECT. Is it the case that PHYSICAL-OBJECT $\sqcap$ INFORMATIONAL-OBJECT $= \perp$? The argument may be a bit harder to make here, but it would appear that informational objects and physical objects also have incompatible individuation properties. For instance, one important thing about informational objects, abstract objects that contain information can have multiple concrete instantiations, if indeed they have any physical instantiations at all. Individual physical objects cannot have multiple concrete instantiations. This leads to different counting principles for informational objects and informational objects, which, as we shall see later, are at the heart of what Pustejovsky and I called ”quantificational puzzles”. Moreover, these modal properties are incompatible and so it would appear that the intersection or meet of the two types is empty. Nevertheless, books are in some sense both informational and abstract objects

(5.2) a. Chris burned a controversial book.

b. Chris read every book he burned.

I think it’s pretty clear that for some • types, the glb of the constituent types is $\perp$. What follows from this, however? Well, if $glb\{\alpha,\beta\} = \perp$ in virtue of incompatible properties that objects of those types necessarily have, then no such complex type can have any inhabitants; that is, if $x\colon \perp$, then the interpretation of x relative to any points of evaluation is empty, because otherwise this semantic value or denotation of x would have to have incompatible properties, in the sense of two properties that cannot be co-instantiated. Since there are obviously books, then if we assume that books have a dual nature in some sense yet to be specified, CTH must be wrong.

Another argument about why an intersective type would be wrong is immediate in a case where the two types in a • type do have a glb that is not $\perp$, as in for example (5.3a,b).

(5.3) a. The apple is red.

b. The apple is juicy (is delicious).

The apple as a physical object or as a food has a certain part, its skin of which one can make certain predications, viz. about its color, that don't hold of the entire object. But if we think about types intersectively and as sets of objects, then the intersection of the type SKIN and FOOD just give us the skin of the food, and that's *not* what tastes delicious or is juicy. So once again a simple analysis of a • type as an intersective or conjunctive type is incorrect. It just seems wrong to model objects of • types as instersections of types. Of course this is an *extensional* construal of intersective types and I have assumed an intensional theory. But it stands to reason that such intensional intersective types should determine extensional meets in the type structure. I conclude that whatever • types are, they cannot be conjunctive types.

5.2 Another Model of • Types: Pair Types

In the Category Theoretic interpretation for types that provides us with an abstract model of complex types, we model • via a pair or types or a product. Pairing *can* be interpreted as the appropriate construction for conjunctive types in intuitionistic logic. But this interpretation is not available to us, because of the presence of the subtyping relation on types and the fact that we have, in the cases we are interested in for terms of type $\alpha \bullet \beta$ that $\alpha \sqcap \beta = \bot$. If the intersective model of • types is not appropriate, what alternative do we have? Perhaps we may interpret the category theoretic construction in a different way.

We could, for instance, consider the product construction to issue (as it does for the category of set) in a collection of pairs.

- **Pair Types Hypothesis**:
  $\alpha \sqcap \beta := (\alpha, \beta)$

If we think of objects of • type as inhabitants of such a collection of pairs, then each object of complex type would correspond to a pair, consisting of a component of each constituent type. This leads us to consider the Pair Types Hypothesis (PTH). Pustejovksy (1995) as we saw earlier seems to have had this model in mind.

As Tim Fernando (p.c.) has pointed out, pair types can account for our intuitions about the different individiuation conditions of informational and physical objects. Suppose that on a shelf

a there are exactly two copies of War and Peace, two copies of Ulysses, and six copies of the Bible.

b Pat has read War and Peace and Ulysses, and no other book

c Sandy has read the Bible, and no other book.

Now, consider

- (Q1) How many books are there on the shelf?
- (Q2) How many books has Pat read?
- (Q3) How many books has Sandy read?
- (Q4) Who has read more books, Pat or Sandy?

My guess is that most people would answer

- ten to (Q1)
- two to (Q2)
- one to (Q3)
- Pat to (Q4)

The pairing approach gives us for the situation 10 pair objects consisting of a physical object represented by a number and an information content represent by the title of the book or an abbreviation thereof:

(1, W&P), (5, Bible), (9, Bible) (2, W&P), (6, Bible), (10, Bible)
(3, Ul), (7, Bible) (4, Ul), (8, Bible)

Given that the second element of these pairs are informational objects, it should be noted that, e.g., *Bible* stands for the same information content or informational object in the last six pairs. To answer (Q2)-(Q4), what matters are the second elements, the information contents, to counting how many books a person has read. Counting the first components or whole pairs is relevant for answering (Q1), but not for (Q2)-(Q4).

To handle corpredication in the pair model, we can proceed as we did with conjunctive types, with the difference that instead of requiring a conjunctively typed argument, the conjoined predicates of different types α, β require a pair typed argument (α,β). Copredications are not the problem for this model. But this model does have a problem, a big problem, with simple predications that require only one of the constituent types as in (3.40) repeated below

(3.40b) The book weighs five pounds

The predicate *weighs five pounds* requires that its argument be of PHYSICAL OBJECT or of P type. Let's suppose that the types P and I (the type of informational objects) are objects of a category. Then we can form the product type P × I to model the complex type of *book*. Given categories P and I , we can define the morphisms for the product type

- π_1: $\text{P} \times \text{I} \longrightarrow \text{P}$
- π_2: $\text{P} \times \text{I} \longrightarrow \text{I}$

These morphisms are just what we need, it would seem, to adjust the type of a term like *book* so that its type matches up with the type of the predicate in (3.40). Classic GL Pustejovsky (1995) offers something like this proposal for exploiting complex types. This view, however, also has insuperable difficulties. Suppose we use the projection π_1 that isolates the P component of the book; thus, we retype *book* as of type P. With this manoeuvre, we've in effect coerced the term to pick out just one component of a complex object. This works for (3.40) but fails completely to give an adequate account of examples like those in (5.4) or (5.5). The problem is that when we use such a type projection, we end up losing the information that *book* is also of type I, information that we need to handle predications that occur in other clauses or with expressions that are anaphorically dependent upon the noun phrase containing *book*.

(5.4) John's Mom burned the book on magic before he could master it.

(5.5) Mary bought a book that contradicts everything Gödel ever said.

It will not do to simply retype the term that had the • type with one of its constituent types. If we shift *the book on magic* to P so as to make the predication in the main clause of (5.4) succeed, then the anaphoric pronoun in the subordinate clause will not have an antecedent of an appropriate type. By being anaphorically linked to *the book on magic* which now picks out an object of type P, it will also have the type P and this will lead to a type clash in the predication in the subordinate clause of (5.4). Alternatively, if we try to coerce the object of *master* back to an informational object, we get a typing conflict with the typing requirements of *burn*.[1]

[1] One might think that an appeal to a sort of bridging inference could be made to rescue the view. That is, one might claim that the pronoun refers to the information content associated with

The crucial insight needed to solve this difficulty is that the projections from complex types to the constituent types go with different terms, *not* the original term. So let's in addition to our projections on pair types, let's assume to function symbols f_1 and f_2 that give us new terms associated with t. Modifying the PTH somewhat we could add the following axiom concerning these function symbols:

- **Separate Terms Axiom (STA)**:
  $t : \alpha \bullet \beta$ iff $f_1(t) : \alpha \wedge f_2(t) : \beta$

With the relevant details now omitted, the analysis of

(5.6) The book weighs five pounds and is an interesting story.

under PTH and STA would have the following logical form:

(5.6') $\exists! x(\text{book}(x) \wedge \text{ weighs five pounds}(f_1(x)) \wedge \wedge \text{ interesting story}(f_2(x)), \langle x : (\textsc{p}, \textsc{i}), f_1(x) : \pi_1((\textsc{p}, \textsc{i})), f_2(x) : \pi_2((\textsc{p}, \textsc{i}))\rangle$

What's different about this solution, for example, from the one in classic GL is that we have in effect different terms for the complex object, its physical aspect or part and its informational aspect or part. No term has two incompatible types nor does any term have the intersection of the two constituent types of a • type. Since the latter gave rise to incorrect results, this approach is certainly an improvement.

5.3 Relations Between Objects and Their Aspects

Nevertheless, a problem remains with this analysis. STA, as it stands, assumes a functional relationship between an object of complex type and its aspects. This implies that there are always at least as many objects of complex types as aspects. The problem is, how do we count objects of complex types? We have seen that books as objects of complex type can be counted relative to the individuation criterion provided by the informational component type. They can also be counted relative to the criterion provided by the physical component type. Take for instance the relation between a physical aspect a of a book b considered as a complex • type. It makes sense for us to say in the presence of a and a' that:

the object of type $\textsc{p}$. But I am skeptical. There simply is no such function for *arbitrary* sorts of physical objects, and the only thing we know right now is that we have an object of physical type. One would have to know that what one has is in effect a physical component of an object that has an informational component as well, but the formalism provided by PTH does not guarantee this. My proposal, discussed in the next section, will provide just this information.

(5.7) a and a' are copies of the same book b.

Indeed we can have many physical copies of the *same book*. So the individuation conditions for books in their informational aspect and in their physical aspect are different; we often count books with respect to their informational content, but we cannot do so for their physical manifestations—each different copy is a different physical object. Of course we can also count books with respect to their physical manifestations. Indeed such individuation conditions offer yet another sort of argument for why we cannot consider PHYSICAL OBJECT• INFORMATION as an intersective type, as objects of the second type have different individuation conditions (and hence a different essential property) from objects of physical type.

Can we count objects of complex type, where the constituent types provide two distinct criteria of indivudation, using *both* criteria of individuation. In fact this yields absurd results. Consider the following example. We have once again a shelf of books, where there are three copies of the Bible and one copy of Jane Austen's collected works, which contains *Pride and Prejudice, Emma, Mansfield Park, Sense and Sensibility, Persuasion, Northanger Abbey* and *Lady Susan*.[2] In answering the question, how many books are there on the shelf?, a person might say 'four' counting physical volumes, but she might also answer 'eight', using the informational type to individuate the domain. Which of these will depend on context, certainly. But taking the pair types hypothesis as an ontological thesis—namely each pair of a distinct physical book and a distinct informational content constitutes a countable object—would yield the crazy count of 10 books. From the current perspective, that's counting using two different principles of types, determined by two incompatible types. Thus, we can't count objects of type $\alpha \bullet \beta$ according to the combination of the principles of counting determined by the types α and β. We can only count according to one coherent principle of counting.

Since we can only count objects of $\bullet$ type relative to one coherent individuation criterion, we need to rexamine the functionality of aspects presupposed by STA. We need to determine whether functionality holds between books individuated informationally or physically and their physical and informational aspects. Suppose we count books relative to the informational criterion. Then, as we've seen the relationship between books and physical aspects cannot be functional; one informational book may have several physical copies or aspects. On the other hand if we consider the Jane Austen case again, then we see that books individuated phyiscally may have multiple informational aspects. Once again functionality

[2] Thanks to Julie Hunter and Laure Vieu for giving me this example.

fails to hold. STA as formulated presupposes something that cannot be satisfied in many cases.

Further evidence for a nonfunctional relation between aspects and objects of complex type from the minimal pairs, involving quantification over different aspects of the meaning of the nouns *book* and *question* that we saw above in the quantificational puzzle examples. I repeat these here.

(4.4) a. The student *mastered* every math book in the library.

b. The student *carried* off every math book in the library.

(4.5) a. The teacher *answered* every student's question.

b. The teacher *repeated* every student's question.

The quantification over books in (4.4) is sensitive in one case to its informational aspect, and in the other to its physical aspect. In (4.4a), we simply quantify over all informationally distinct individuals without reference to the instantiations of these informational units. There are, typically, many different copies of certain math books in a library. It is not necessary, for example, for the student to have mastered the books by reading every distinct copy of every math book in the library—or even any physical copy of a book in the library in order for (4.4a) to be true. When we individuate books relative to their information content, there will be no functional relation between the variable ranging over books and the term ranging over the physical aspects of books. These considerations show that the relationship between objects of $\mathrm{P} \bullet \mathrm{I}$ and the physical books of type *p* is not a functional one, nor is it a relation of types to tokens. Instead we have two different types involved with two different counting principles, two ways of talking about tokens. The notation in STA has to be modified and complicated.

In our model of • using product, we have focussed on the set theoretic implementation of this idea. Like many others who have thought about this subject, we have implicitly assumed a naive isomorphism from the model of types to the ontological structure of objects along the following lines. While nothing in what I have said so far speaks against the existence of a functional projection *on the level of types* from the complex *type* $\alpha \bullet \beta$ to its component types α and β, a projection basic to the product construction, I have shown that we cannot ”read off” a similar projection over terms, let alone their denotations. What I have shown is that the product model of types, especially in its realization in the category of sets, does not help us in understanding the nature of the objects that inhabit these types. However, one might want to explicate the notion of pair object, the example of

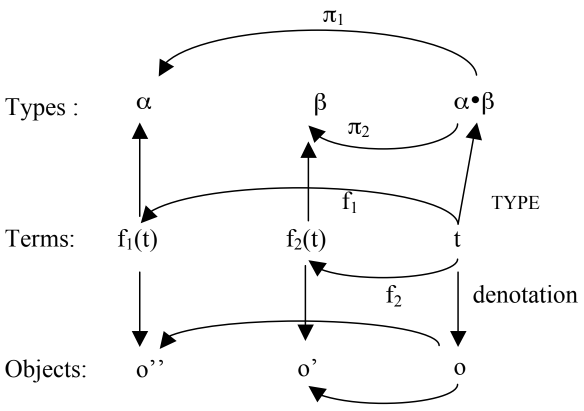


Figure 5.1: Naive isomorphism from types to objects

the shelf of books with Jane Austen's collected works shows that understanding objects of • type as pairs of objects of the constituent types in the set theoretic sense yields absurd results when we attempt to count such objects with the pair model. For according to the standard identity criteria for pairs, two books formalized in terms of their informational and physical aspects as $\langle a, b\rangle$ and $\langle a', b'\rangle$ will be identical just in case $a = a'$ and $b = b'$. In the Jane Austen example above, such individuation criteria would yield the unintuitive result that there are 10 books on the shelf.[3]

Even if we put aside the arguments against having a functional relation between objects of complex type and their aspects, there's still a question as to what is the interpretation of the pairing function at the level of truth conditional content and what is the analysis of the link between the semantic values of $f_i(t)$ and t. That is, how do we concretely spell out the analogues of the morphisms π_1 and π_2 at the level of the metaphysical constitutions of objects? We want $f_i(t)$ and t in the analysis of (5.1) to refer in some sense to the same book, but PTH + STA give us no clue by themselves. What is it to be an object of a pair type? If we knew the answer to this question, we could make progress on the relation between the values of $f_i(t)$ and t, where $f_i(t)$ is on of the terms postulated by STA. It's not that

[3] Of course there is a dependence of identity criteria here: if book a is identical with book b on the physical individuation criterion, then of course they will have the same informational content as well.

the object is itself a set or a pair of objects! When I copredicate two properties of a book in (5.1), I refer to just one object, not two. Furthermore, that object is not a set; if it were, it could not be a physical object, which it is! The problem is that it's in some sense once again both an informational and a physical object; we have not made much headway in understanding this puzzle.

One suggestion is that we understand the dependence of the object of simple type on the object of complex type in terms of the parthood relation. In terms of the now fatally flawed notation of STA, $f_i(t)$ would be construed as a part of the semantic value of f.[4] A lunch for example is part event and part food; it is the singular fusion of its parts. But normal parts of objects have names and can be referred to. This isn't true of the inhabitants of • types like lunches. This should lead us to be suspicious of this view.

Such suspicions guide us to a deeper problem. The presence of different individuation conditions for an object of • type like *book* militates effectively against a simple mereological conception of objects of complex type as being the mereological sum of objects of the constituent types, for on such an account such objects would have different identity conditions depending on which part one used to determine the criterion of individuation. We can formalize this situation using a relation of identity relative to some type providing a principle of individuation. Consider again two physical copies of the same book (individuated informationally). Letting $=_p$ stand for the identity relation relative to an individuation criterion appropriate for physical objects and letting $=_i$ stand for the identity relation relative to the criterion appropriate for informational objects, we have:

(5.8) a. $b_1 =_i b_2$

b. $b_1 \neq_p b_2$

In mereological terms, this implies that b_1 and b_2 have a common informational part but distinct physical parts. But then by the axioms of mereology, we have in terms of an absolute identity relation (in standard mereology two objects are equal just in case they have exactly the same parts):

- $b_1 \neq b_2$

But now this mereological predicts that in the example with the collected works of Jane Austen,we get the implausible count of 10 distinct objects! Thus, the

[4] Cooper, for instance, (p.c.) claims that lunches are composite objects with an event component and a food component.

mereological conception like the pair conception of objects of complex type seems fatally flawed.

Another proposed analysis[5] construes objects of complex type as collections like groups or orchestras. The latter suggestion certainly seems on the wrong track. Singular nouns that refer to groups can support plural anaphoric reference:

(5.9) a. The orchestra got ready. Then they started to play the Bach suite.

b. The battalion was in trouble. They were receiving heavy fire from the enemy. They called in for tactical air support.

But plural anaphoric reference with *lunch* is crazy:

(5.10) The lunch$_i$ for Chris Peacocke was very nice. They$_i$ pleased him too.

Arguably, 'nice' and 'please' can take both events and meals as arguments, so we should be able to predicate this jointly of both the event and the meal. But the plural anaphora is semantically uninterpretable when the coreference is stipulated as above. Since a singular use of *lunch* cannot support plural anaphora, this strongly suggests that it does not refer to a plurality, in the way *orchestra*, *battalion*, team arguably do. plural anaphora is semantically uninterpretable when the coreference is stipulated as above.

The alternative is to claim that an inhabitant of a • type is single but composite. Perhaps one can just bite the bullet about the counting argument given above. But mereological talk seems distinctly out of place when trying to make sense of objects like lunches. For ordinary objects like lunches, we have something like Benaceraff's problem: an analysis of such objects in terms of a theoretical apparatus like mereology just seems wrong.

(5.11) Part of the lunch is an event and part of the lunch is a meal.

Finally, we readily make sense of a parthood relation among objects of the same type. Physical objects have physical parts; an apple has various physical parts, the core, the skin, the flesh, and so on. At least some events also have readily identifiable event parts. Such homogenous objects easily have causal interactions as well. A much vaguer notion of parthood must be invoked to explain the inhabitants of • objects on the mereological view. Unrestricted mereological composition aside, we normally do not think of objects as having parts of different types. Substance dualist positions concerning persons are an example of such, but much of the

[5] Also due to Cooper, p.c.

unintuitiveness of this view comes precisely from our inability to have a clear conception of how the various parts interact causally. Further in order to accommodate a mereological conception of the inhabitants of • types, we would have to suppose substance *multi-ism*. Not only would we have to worry about minds and bodies as being parts of the same person; we would have to worry about events and foodstuffs being parts of the same object, information and paper as being part of the same object, and so on. Dualism would be simple in comparison to the metaphysical view being proposed. In each case we would have to elaborate some sort of special causal or other relation telling us how changes in one part might affect another. But this seems crazy for inhabitants of • types. When I tear pages out of a book, this alteration in the physical part doesn't cause a change in the informational part; in tearing out pages, I've changed the both the physical and the information content; it's not that there are two parts—there just one object the book, with two aspects. The part whole model, at least where we understand such talk in its physical, spatio-temporal sense, is not the appropriate one for lunches or other objects of • type.

What I've shown is that not all objects of complex type $\alpha \bullet \beta$ can be understood as mereological sums of objects of types α and β. But I have not shown that no complex types involve such mereological relations. As Kleiber notes, predications can be sensitive to meronymic information and part whole relations license copredication phenomena as well. (5.12) provides examples of copredication, where one predication applies to a part of the object while the other applies to the whole object or another part.

(5.12) a. The apple was red and juicy.

b. The car is shiny and powerful.

We can analyze such copredications in a similar manner to those we have already examined; the • type formalism is general enough to model a copredication to a term in which the two predicates pick out different parts of the term's denotation, as in (5.12). Metonymic predications sometimes lead to different counting principles.[6] Suppose that a company makes computers with dual processors, two CPUs for each machine. Then it appears that there are two ways of counting (CPUs or dual CPU machines) suggested for computers in the following sentence.

(5.13) The company has produced 500 computers.

Nevertheless, not every case of objects of • type need involve two incompatible counting and individuation criteria.

[6] I am endebted to Magda Schwager for this example.

5.4 The Relational Interpretation

We can accept PTH as a model of types if we wish but we have to modify STA. We need to take seriously the relation between the objects *a* whose type are constituents of some complex type an object of which *a* is intuitively related to. To avoid Benaceraff's problem, I will complicate the notion of predication rather than try to ride roughshod over our intuitions about ordinary and familiar objects. Predication typically involves the attribution of a property to an object considered under a certain conceptualization; this is what an aspect is. It is also, I maintain, what lies at the bottom of relative predication. To consider a book as a physical object is to think of it under a certain aspect; to consider the book as an informational object is to think of it under another aspect—similarly, for other object of • type. As we'll see, these aspects may also be quantificationally complex: the true informational aspect of a book is: an informational of a certain kind that has a particular sort of physical realization. This will allow several physically distinct books to have the same informational aspect. In addition, natural language does not give a preeminent status to physical objects as opposed to non physical ones. Thus, the ”informational aspect” of book is just as much an object of good standing as the physical aspect of book. And it is these thick individuals (Kratzer 1989) that we count and quantify over.[7] Thus, our primary ontological objects thick or clothed objects as opposed to bare particulars. Nevertheless, the way I think of aspects is that they depend on bare particulars. An aspect is, metaphysically speaking, a bare particular combined with some property or some property instance that it has (Asher 2006). We can speak of a mereological or constituent relation over these aspects, if we wish. But crucially this is not a parthood relation over the object itself, for we need not consider the object to be the sum of all its aspects. Given that we have defined aspects in a particular way, the sum of an object's aspects cannot be identical to the object itself (since each aspect contains the object together with some property that it has).[8] A lunch object is *wholly* an event (under one aspect) and wholly food (under another aspect). When we speak or think of lunches as food, there's no ”other part” of the lunch itself that's left out and that is an event.

The view that I propose then is not substance multi-ism but some sort of property multi-ism. These properties together with objects comprise aspects, and in ordinary language it is objects with respect to some property that we count and

[7] This is related to but also different from **?**'s proposal of conceptual covers for quantification. I thank Magda Schwager for pointing this out to me.

[8] Contra Asher 2006.

quantify over. Some objects, namely those of $\bullet$ type have two or more perhaps incompatible aspects and so give rise (according to context) to two or more principles of counting. But when counting or quantifying over objects of $\bullet$ type with two or more incompatible constituent types, we must choose one of these to guide counting and individuation. Thus predication is in some sense constitutive of the domain.

A point in favor of property multi-ism and the aspect theory, is that there are many aspects of lunches that we can make explicit using the device of relative predication. Indeed we can construct arbitrarily many using the relative predication construction:

(5.14) a. lunch as a gastronomic experience

b. lunch as a time to relax

c. lunch as an opportunity to smoke

Restricted predications or predications that take only a single aspect into account essentially involve predications on the trope defined by the property governing the aspect.

Even if we defined aspects or tropes differently, it seems unlikely that tropes are parts. If they are, they are for the most part inessential parts. John does not cease to be John if he ceases to be a banker. Further, if the *qua* construction allows properties like that of being a paddle to constitute a trope involving a book, then these parts cannot be essential parts as well. But now clearly, an object cannot be the sum of its parts in the sense that if the identity of the object changes if it loses one of its parts. To say that objects are the fusion of their tropes would make their tropes count essentially toward their individuation conditions. There are some tropes that do seem to be constitutive of objects such as books. As we've seen, aspects contribute to sometimes divergent individuation conditions for objects that are of complex type.

Another way in which parts and aspects differ is this. If an aspect of a thing exists, then the thing itself must exist as well, but this is not true of parts in general. This gives us an explanation of why something like STA is needed. Aspects are dependent upon the objects of complex type. I will codify the relation between aspects and the objects of which they are aspects with the relation O-elab, which stands for Object Elaboration. When I write O-elab(x, y), I mean x is an aspect of y, or x ”elaborates” on the sort of object y is.

Although tropes that define aspects of things are not constitutive of the objects themselves, we can still ask about an ordering on tropes. For example it seems that John's being a banker in Toulouse is a more specific trope than the trope of John's being a banker. We can think of this ordering now and wonder whether tropes or aspects are maximal are not. From the study of restricted predication, we see that in fact it is desirable to have these be maximal tropes. That is *John as a banker* picks out the maximal trope, John the banker. So if we want to predicate something of a particular trope of an object denoted by a term t, then we can think of the relation between an object of type τ and the trope as a function from τ to another type σ.

So far I've talked about why complex $\bullet$ types are needed to analyze copredication and how the different predicates in a coordinate construction select different tropes in which the object has a type that is a constituent of the complex type of the object of the copredication. However, in addition to copredication constructions, there are other grammatical and lexical devices that introduce or select $\bullet$ types. In Asher and Pustejovsky (2004), Pustejovsky and I argued that the verb *read* is a predicate that requires a dot object as its complement; it can even coerce its direct object into something of just this complex type, namely, an informational entity with physical manifestation.

(5.15) a. Mary <u>read</u> the book.

b. John <u>read</u> the rumor about his ex-wife.

c. Mary <u>read</u> the subway wall.

The coercion phenomenon in (5.15) involves a subtle shift in meaning. One can hear rumors and spread rumors, which one cannot do with books (even if you're listening to a book on tape); on the other hand, one can't see or look at rumors whereas one can see or look at a book. In contrast, one can see a subway wall or look at it, without getting any informational content. However, in (5.15b,c) the arguments of *read* change their meaning. For instance, (5.15c) implies that the subway wall is a conveyor of information, and the only way to understand (5.15b) is to assume that the rumor has been printed or exists in some physical medium. One explanation of this phenomenon is that *read* coerces its arguments into objects of the same type as *book*. For both (5.15b) and (5.15c) the predicate coerces its complement to the appropriate type, that of an informational object with physical manifestation. In each of these cases, there is a "missing element" to the complex type: for (5.15b) the coercion effects the introduction of the physical manifestation to the otherwise informational type; for (5.15c) the coercion results in the introduction of an informational component to an otherwise merely physical type. Other grammatical means for introducing such complex types involve relational predication, as we will see below in chapter 7.

Our discussion of aspects now enables us to amend our conception of the types themselves and the relation of the type structure to the metaphysical nature of aspects and the individuals they are aspects of. The main problem with our formal product model of the types was in the way we interpreted the structure on the types with respect to the inhabitants of that type. If • is a product of types, it is nevertheless not wise to read off the ontological structure of the inhabitants from the structure of the type. If there is a functional projection from the product type to its constituents *types*, this function does not necessarily exist at the level of the inhabitants of the types, because of the way inhabitants of • type may be individuated. It was an even worse idea to interpret the product construction within set theory as a set theoretic pair and assign that structure to the object. The lesson to be learned is perhaps to be wary of using the set theoretic model of a categorical construction as a guide to ontology. But we can capture much of the structure of the inhabitants of • types via a categorial construction related to product, something known as a *pull back* or *fibre product* which is defined as follows (see e.g. Crole (1993)).

Definition 2 *Let C be a category and $X, Y, Z \in C$, with morphisms $r : X \to Z$, $t : Y \to Z$. The fibre product of X and Y over Z , denoted $X \times_Z Y$, is an object $W \in C$ with two morphisms $\pi_1 : W \to X$ and $\pi_2 : W \to Y$ satisfying $t \circ \pi_2 = r \circ \pi_1$, such that for every $V \in C$ and morphisms $f : V \to X$ and $g : V \to Y$ satisfying $t \circ g = r \circ f$, there exists a unique morphism $u : V \to W$ such that $f = \pi_1 \circ u$ and $g = \pi_2 \circ u$.*

The interest of a fibre-product is that in the category of sets it collects those pairs that project down to the same element in Z. This allows us to model the behavior of • objects relative to distinct criteria of individuation supplied by the constituent types. That is, I take $\alpha \times_\gamma \beta$ to be the model of a • type $\alpha \bullet \beta$, where $\gamma = \alpha$ or $\gamma = \beta$.[9] The fibre product $\text{P} \times_{\text{I}} \text{I}$, where t is a 1-1 function that preserves all properties and r is the "contained in" function collects all those pairs that have in effect the same information content—thus corresponding to books informationally construed, while $\text{P} \times_{\text{P}} \text{I}$ collects all those pairs (in the category of set) that have perhaps differing information contents but occur in the same physical volume. So for instance, if we are individuating books relative to their information content, then many possible physical aspects of the book will be mapped under the relevant morphisms to the same informational aspect.

[9] Note that the product or the fibre product of arbitrary P and I inhabitants is not guaranteed to exist by this construction, which is well and good, because we do not want just any old physical object and any old informational object to form a dual aspect object.

$$\begin{tikzcd}
P \times_I I \arrow[r, "\pi_1"'] \arrow[d, "\pi_2"'] & P \arrow[d, "\text{contained in}"] \\
I \arrow[r, "ID"] & Z
\end{tikzcd}$$

Figure 5.2: The ”I” pullback over P • I types

When we individuate books relative to their physical aspect, we have a slightly different pullback in which the individuation conditions for physical aspects of books determine the number of inhabitants of the type. this situation is represented in Figure 5.3.

Fibre products or pullbacks give us a model of • types *and* a model of how the inhabitants of a • type are to be understood—how they are individuated and how they relate to their aspects. It is the structure of the whole pull back with its different maps that tells us about the objects in the category. For instance, in the picture above, it is the ”Z” objects that are counted; for it is they that reflect the effects of the criterion of individuation. They are also isomorphic to one of the aspects and to the projections onto either the P or I types. They are in 1-1 correspondence and preserve all of the same properties that the objects that are the values of those projections do. We can also read off from the entire structure of the pull back when objects of • type are *not* well defined. For instance, if we cannot specify the maps from the constituent types to Z in a metaphysically natural way, then the pullback may in fact not exist. For instance, there is no FRUIT • CAT type or inhabitants of such a type, because there is no natural way of specifying maps from fruit and cats to the complex object.

Pullbacks within a Cartesian Closed Category serve as a suitable interpretation of • types. And we will see below that all other complex types used in this book can also be modelled within Closed Cartesian Categories. One cautionary lesson we must draw, however, is that an object of complex type does not have the *internal structure* of a pull back in the category of sets; we don't want to say that an object of complex type is a restricted set of pairs any more than it is a set of

$$\begin{tikzcd}
P \times_I I \arrow[r, "\pi_1"'] \arrow[d, "\pi_2"'] & P \arrow[d, "ID"] \\
I \arrow[r, "\text{exemplified by}"] & Z
\end{tikzcd}$$

Figure 5.3: The ”P” pullback over P • I types

pairs or a pair. The virtue of category theory is that we are not forced to this set theoretic model; in effect, I am saying a book as an inhabitant of a complex type is a pullback with respect to its aspects. A pull back in the category of sets gives us a particular formal model of this structure, but not one from which we are forced to read off messages about ontological constitution.

5.4.1 Effects of • Types on the Models for Logical Forms

I haven't said much about the models of the logical forms themselves. Are there any modifications of standard models of dynamic semantics that the transfer rules call for? There is one modification that we must make to the standard models in order to accommodate the inhabitants of complex types. Because I have introduced terms that refer to both thick individuals (aspects) and thin individuals (individuals of complex type), we must have both in the domain. In the quantificational puzzles, we saw that counting and quantification may be directed over physical aspects or informational aspects of books, or over books as objects of complex type, depending on the predicational restriction. There is ample evidence to suggest that in fact we do count or quantify over aspects or thick individuals in the relevant circumstances as well over thin individuals.

To model this, I must postulate domains for each type. Domains D_α for simple types α have their own individuation and counting criteria provided by α. But what about the inhabitants of $\bullet$ types? Could they simply be identical to those that inhabit the simpler types?

The answer to this last question is not a straightforward yes. It seems obvious that the following axiom should be adopted, however, governing types between identicals:

- $x = y \rightarrow (x : \alpha \leftrightarrow y : \alpha)$

or, given that typing contexts are functions from terms to types:

- (ITID) $t_1 = t_2 \rightarrow \textsc{type}_C(t_1) = \textsc{type}_C(t_2)$, where C is the typing context for the formula $t_1 = t_2$.

In fact ITID follows from TCL as a theorem, if we assume the following lexical entry for ’is identical to’”

- *is identical to*: $\lambda x \lambda y x = y \langle \textsc{type}(x) = \textsc{type}(y) \rangle$

But it also seems reasonable to say that $\alpha \bullet \beta \neq \alpha$. They really are distinct types, because objects of $\bullet$ type have different properties from those of type α. Together with the principle of identical types for identicals, we now see if $x : \alpha \bullet \beta$ and $y : \alpha$ in any typing context C, then we can immediately infer $x \neq y$. Now suppose that $D_{\alpha\bullet\beta}$ and D_α have a common inhabitant. This should make (5.16) true in our composition logic:

(5.16) $\exists x \exists y\ x = y,\ \langle x\colon \alpha, y\colon \alpha \bullet \beta \rangle$

But now we can derive an immediate contradiction from ITID. So the inhabitants of $\alpha \bullet \beta$ must be disjoint from the inhabitants of the constituent types, unless we give up the plausible entry for ’is identical to’.

How do individuals and their aspects relate in the ontology of the model? The categorical notion of a pullback or fibre product gives us an abstract model. Can we say more in a different medium? Over our domain, we impose a partial ordering to model the ordering of the thick individual with the bare or thin individuals at the top. The partial ordering is defined using the subtyping relation *sqsubseteq* on types. Note that the typesα and β have corresponding formulas ϕ_α and ϕ_β in the object language. I will assume that $\alpha \sqsubseteq \beta$ if it’s a theorem in the logic of the object language that $\phi_\alpha \rightarrow \phi_\beta$—i.e. $\vdash \phi_\alpha \rightarrow \phi_\beta$

- Partial ordering on aspects:
  $x \geq y$ iff $\alpha \sqsubseteq \beta \wedge x\colon \alpha y\colon \beta \wedge$ O-Elab$(x, z) \wedge$ O-Elab(y, z), provided z, x and y obey the same individuation criteria.

The partial ordering on aspects ensures what we want about aspects, namely that they are maximal, as long as we're not counting the object of complex type relative to one individuation criterion and the aspects according to another. Thus, since bankers and people have the same individuation criteria, then there is only one aspect of John as a banker, which, recall, was a desideratum we placed on aspects. Furthermore, our partial ordering says that John as a banker in Toulouse is a subaspect of John the banker (tout court). On the other hand, if we look at physical aspects of books, then the aspects and the object of complex type may not obey the same individuation criteria. In that case, there may be several distinct physical aspects for a book that is individuated informationally (—e.g., there may be 3 physical copies of *the Bible*). If the book is individuated physically, however, there will be only one physical aspect corresponding to it, which is also as desired.

This leaves us with a very rich universe of objects when we consider complex types. But there is yet another complication. As we have seen, to count individuals we need a principle of counting given by a simple type. We cannot coherently count objects of • type except with respect to one of the constituent types' counting criterion, at least if the constituent types suggest two distinct criteria. So we mustn't count the physical aspect of a particular book and the informational aspect of a book as two different books. But we can count books either as informational or as physical objects (though not coherently as both).

- one way of counting: informational object with a physical realization
- another way of counting books: physical object with an informational content.

Consider the Jane Austen case again, where we have one physical volume containing Jane Austen's seven published novels together with three copies of the *Bible* on a shelf. We have a domain of objects of P • I type together with the I type and P type inhabitants. We should expect that given what we have determined, the principle of counting will determine the number of P • I inhabitants. We will have four with the P type criterion and eight according to the I criterion. In effect the counting criterion here determines the cardinality of the model! If we graph the Jane Austen situation using lines for the O-elab relation, we have for the case where we individuate the P • I inhabitants, $b_1, b_2, \ldots$ via the P type:

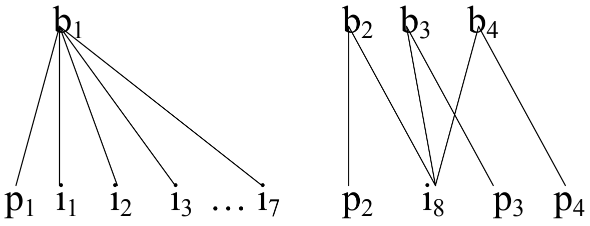


Figure 5.4: Books individuated physically

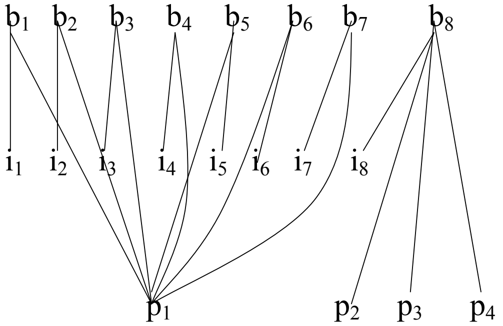


Figure 5.5: Books individuated informationally

Whereas if we individuate with respect to the I type, we have:

The number of P•I inhabitants is either in 1-1 correspondence with the physical aspects of the P • I objects or with the I inhabitants, the informational aspects of the P • I objects. We have in effect two models with two different sets of P • I inhabitants, depending on which criterion of individuation is chosen.

One way to think about this is from the perspective of relative identity.[10] To count one needs a criterion of identity and individuation, that is a criterion of *relative* identity. The bare objects of • type are counted and individuated relative to one of their constituent types. Such criteria of relative identity are typically understood as equivalence relations over the basic individuals.

[10] Thanks to Magda Schwager for this suggestion.

While this is a possibility, there remains the question what are the basic individuals? Can we say how many of them there are? Kit Fine in his paper on aspects (**?**) takes physical individuals to be basic. But as we've seen in some of our examples, there might not be enough physical individuals to get the right counting conditions for objects individuated in another way. And it's not clear that physical objects should be the basic sort of individuals. Finally, this makes our aspects complex set theoretic constructs, equivalence classes of basic objects, and this seems objectionable for the reasons that I've discussed earlier.[11]

What is really going on in the counting examples? It helps perhaps to consider these examples from a dynamic semantic point of view. In dynamic semantics successive sentences in a discourse refine an information state which consists of a set of world assignment pairs. Each formula in dynamic semantics is an action that does something to the information state: most formulas simply are tests and throw out world assignment pairs that don't satisfy them in the classical sense. Existential quantifiers modify or extend the assignment functions. When a discourse contains a question *How many books on the shelf are there?* and a response like *two four* or *eight*, the discourse context changes, and only those world assignment pairs where the number of books is the number that the response gives are left. However, some of these responses are intuitively compatible; they just seem incompatible because of the use of different criteria of individuation. So we should simply relativize the domain of quantification in a world to a criterion of individuation, an idea which is close to Geach's idea of relative identity.

It is not easy to switch counting criteria in mid discourse using expressions that refer back to the same objects.

(5.17) There are ten books on the shelf. Fred has read three books. Yet Fred has read all of the books on the shelf.

This discourse should be in principle coherent; we could have counted the 10 books using a physical individuation criterion and counting them again as three using a different individuation criterion. This of course makes sense if indeed we relativize the cardinality of the inhabitants of a complex type to a particular criterion of individuation. If we count the objects differently, we can't square the two countings together.

But once again things change when we give some contextual help, as in the counting examples earlier.

[11] Actually for informational objects this doesn't seem such a bad strategy in general, although it gets the wrong results for the Jane Austen example.

(5.18) There are three copies of the Bible, three copies of *Emma*, and four copies of *Formal Philosophy* on the shelf. Thus there are ten books on the shelf. Fred has read *Emma*, *the Bible*, and *Formal Philosophy*, that is three books. Yet Fred has read all of the books on the shelf.

What we've done is supplied the informatoin to construct the right sort of pull-backs to understand the shifts from one counting standard to another. Thus, we see that discourse context can shift the cardinality associated with the domain of a • type by shifting the criterion of individuation. Our analysis of types has led to another form of underspecification.

Not all complex types introduce distinct or even incompatible individuation and counting criteria. For instance, consider the complex types introduced by the as phrases *as a janitor* and *as a salesman on E-Bay*. We do not want to count John as a janitor and John as a salesman on E-Bay in two different ways, and we don't have to, since janitors and salesmen on E-Bay arguably have the same individuation and counting criteria as persons. Furthermore, we don't count these two aspects as two different people. Similarly, John as a doctor in village A and John as a doctor in village B are also not two different people. There is only one person John, though he has several, indeed perhaps many aspects. Aspects aren't counted as distinct objects from within the language at least unless they come with distinct individuation conditions. One such example is the following.[12] Suppose that a county must by law have five judges each with a different jurisdiction. John happens two fill two of those functions, the other three by three distinct people. Are there five judges? Has the law been fulfilled? It would seem so; there are five judges but only four people. But that is because the individuation conditions for a judgeship's being filled given by the law are different from the individuation conditions for personhood.

There doesn't seem to be any special criterion of individuation or counting for aspects in general. In fact if you stop to think how many aspects do objects have, it's in general impossible to answer this question. This is because aspects involve a conceptualization of the object and they can be created almost at will from any property that an object has in principle—this is what happens in fact with relative predication, as I've already said and will develop in detail in chapter 5. So in general it doesn't make sense to say that there are just so many thick individuals in our models. However, we could perhaps force such a counting by introducing a type ASPECT into the theory, but this is not part of the normal system of types—it's

[12] The example is due to Julie Hunter. Another example has to do with the much discussed statue/lump of clay object, Lumpel.

kind of like counting sakes.

While we don't count aspects, however, conjoining them forms a plural collection demanding plural agreement with its predicate.[13]

(5.19) John as a gay stripper and John as a banker are a weird combination.

This provides additional support that aspects are indeed objects in the model.

[13] Thanks for this example to David Nicolas.

Chapter 6

The Type Composition Logic TCL with • Types

This chapter provides a formalization of our theory of predication, It will provide a composition logic with complex types, a logic that tells us how to compose bits of logical form together to form the logical form for a clause or a sentence. This composition logic, Type Composition Logic or TCL, will enable us to derive the appopriate logical forms for predications that involve objects and properties of complex type. Here, I'll just concentrate on • types. In chapter 6, I'll revisit the issue of coercion and logical metonymy and extend TCL with another complex type.

In standard formal semantics, a composition logic is used to build logical forms for clauses or sentences of a natural language. It assumes a syntactic structure for the clause to be composed and basic lexical entries for each word, with perhaps some morphological decomposition. A composition logic exploits the rules of the lambda calculus to produce a normalized logical form, or lambda free expression (in most cases) giving the semantic content of the clause in a logically perspicuous language, usually that of first order or higher order intensional logic. The semantics of terms in the usual composition logic is just that of higher order intensional logic, and so the composition process is not isolated as a special system. Even first order intensional logic is of course undecidable. Without separating out the composition logic from the general semantic framework, we cannot prove that the composition process has any interesting computational properties.

On the other hand, if we separate out the compositional logic from the intensional semantics, and endow it with its own interpretation, then we can show that the process of constructing logical form does have interesting computational properties. Thus, formulas in our Type Composition Logic (TCL) will have an *external semantics*, where the terms refer to mind external properties and objects, given by the intended interpretation of the formulas of higher order intensional logic, but these formulas will also have an *internal semantics* via their types. This semantics is used to verify the composition process. The categorial interpretation of the types provide the internal semantics for our formulas. This internal semantics does not of course determine the external semantics, but it does capture certain general features, for instance, of the dual aspect nouns. We will check that the rules of TCL are sound with respect to the categorial interpretation. Because TCL concerns only type assignments and shifts in logical form due to type assignments, the logic has some interesting properties.

I now introduce the notion of a TCL formula and a TCL derivation from a syntactic parse and assignments of TCL terms to the words or basic morphological constituents in the parse tree via the lexicon. Like all formulations of the typed lambda calculus, the TCL language takes as fundamental the notion of a term of the lambda calculus together with a typing context C. The premises for a TCL derivation involve a set Δ of premises about the basic lexical entries, the type hierarchy (i.e., whether one type is a subtype of another or whether two types have a greatest lower bound that is or is not $\bot$) and about which lambda terms apply to which, as dictated by syntax. The other component in the derivation is the typing context, C, which is an assignment of types to variables and constants in all the terms mentioned in Δ. The TCL rules say how these assignments can be used to assign types to other terms in t or t itself. I shall write $\Delta, C \vdash \phi$ for information ϕ that follows from Δ and C; $\Delta \vdash \psi$ and $C \vdash \chi$ are interpreted analogously.

Some information is dependent only on Δ. For instance, which lambda term associated with a given word applies to which other such term is not affected by the typing context. Thus, where $t[t']$ indicates the application of a term t to another term t', where t and t' are lexically given, we can show

Lemma 1 *Let ϕ be any formula not involving a type assignment to a term. Then: $\Delta, C \vdash \phi$ iff $\Delta, C' \vdash \phi$, for any C and C'.*

This persistence property of derivations is important because as I mentioned earlier the type composition logic is dynamic in the sense that some rules of the logic will change the typing context. That is, we can expect to have sequences in a derivation that look something like this:

$$\frac{\Delta, C \vdash \phi}{\Delta, C' \vdash \psi}$$

TCL rules will change contexts with an input context C going to a new context C' (written $C \mapsto C'$) by adding new variables with their type assignments and by combining consistent typing constraints on variables already present in the typing context. In each of these cases, I will show that

Lemma 2 *If the application of a TCL rule, given an input context Δ, C, postulates $\Delta, C \mapsto \Delta, C'$, then Δ, C' is assured to exist.*

TCL derivations are designed to deliver normalizations for logical forms for clauses or sentences. They presuppose a given syntactic parse of the clause or sentence. The syntactic parse is a tree structure and dictates how the normalization procedure should go and also dictates what should be the types associated with the terms of the leaves: we begin with the leaves and then compute the result of combining lower nodes. The close correspondence between syntactic categories and semantic types in semantics, which has been in semantics since the beginning of Montague Grammar and is part of compositional interpretation, means that normalization will succeed unless an unresolvable type clash occurs during application.

6.1 Types and Basic Rules

In TCL, there is a collection, Type, of types consisting of:

- **Simple or Primitive Types:** These include e the general type of entities and T, shorthand for the type of propositions. The set of simple types, ST, together with the subtyping relation $\sqsubseteq$ form a lattice. I will assume that ST also contains a finite set of atomic types that are all subtypes of e.
- **Functional Types:** If σ and τ are types, then so is $(\sigma \Rightarrow \tau)$
- **Dot Types:** If σ and τ are types, then so is $(\sigma \bullet \tau)$
- **Underspecified types:** If $\alpha \in Type$, then $\alpha \bullet ?$ and $? \bullet \alpha$ are types.

Using $\sqsubseteq$ on the simple types, we can also define a greatest lower bound operation $\sqcap$ for elements of ST. $\sqcap$ has the usual properties—e.g., idempotence, commutativity, and $\alpha \sqsubseteq \beta$ iff $\alpha \sqcap \beta = \alpha$. TCL captures incompatibility between types in terms of their common join, $\perp$.

We can generalize $\sqcap$ to provide a notion of greatest lower bound that interacts with the complex types.

Definition 3

- $(\alpha \Rightarrow \beta) \sqcap^* (\gamma \Rightarrow \delta) = (\alpha \sqcap^* \gamma) \Rightarrow (\beta \sqcap^* \delta)$
- $(\alpha \bullet \beta) \sqcap^* (\gamma \bullet \delta) = (\alpha \sqcap^* \gamma) \bullet (\beta \sqcap^* \delta)$
- *otherwise,* $\alpha \sqcap^* \beta = \alpha \sqcap \beta$

From now on we take $\sqcap$ in our rules to reflect this generalized greatest lower bound. We can also define the useful notion of subtype with our extended notion of greatest lower bound.

Definition 4 *Subtyping:* $\alpha \sqsubseteq \alpha'$ *iff* $\alpha \sqcap \alpha' = \alpha$

I stipulate some facts about functional types and $\bot$ that make sense within semantic composition (and the set theoretic interpretation of types): $\alpha \Rightarrow \beta = \bot$, if α or $\beta = \bot$ These facts show that $\bot$ cannot be used to define negation as in Linear types Girard (1987). Our types are linear but correspond only to the negation free fragment. Negation in the lexical semantic type hierarchy does not have an obvious use or sense.

In order to ensure no unwanted interactions between simple types and • types, I assume that • types have a type lattice of their very own do not occur within the lattice of *simple* types, except ”at the very top”. That is. $\alpha \bullet \beta \sqsubseteq \gamma$ iff $\alpha, \beta \sqsubseteq \gamma$.

I also introduce a notion of ”lazy” greatest lower bound that I will use to define a version of unification in the lambda calculus that extends to • types. Lazy glb, $\sqcap^-$ is the same as $\sqcap^*$, except that when we take the lazy glb of a • type, $\alpha \bullet \beta$, and a type γ such that either $\alpha \sqcap^- \gamma$ or $\beta \sqcap^- \gamma$ is defined, we get $(\alpha \sqcap^- \gamma) \bullet \beta$ or $\alpha \bullet (\beta \sqcap^- \gamma)$). This allows us to factor ”inside the dot.”

We can quickly establish some facts about subtypes.

Lemma 3 *Subtype Facts about Complex Types*

a

$$\frac{\alpha \sqsubseteq \alpha'}{(\alpha \bullet \beta) \sqsubseteq (\alpha' \bullet \beta)}$$

b

$$\frac{\beta \sqsubseteq \beta'}{(\alpha \bullet \beta) \sqsubseteq (\alpha \bullet \beta')}$$

c

$$\frac{\beta \sqsubseteq \beta'}{(\alpha \Rightarrow \beta) \sqsubseteq (\alpha \Rightarrow \beta')}$$

d

$$\frac{\alpha \sqsubseteq \alpha'}{(\alpha \Rightarrow \beta) \sqsubseteq (\alpha' \Rightarrow \beta)}$$

6.2 Simple Rules of TCL

TCL rules divide into those that deal with types generally, the ”simple rules of TCL”, and those rules that are specific to a particular complex type. These rules operate on two levels: the type level, where they correspond to simple proof rules on types, and the level of the logical forms themselves. Type shifts may give rise to changes in the logical form, especially for complex types.

I now introduce the simple rules for TCL. Much of the TCL system is familiar to those who know about the λ calculus. For instance, TCL contains **Application**, which corresponds in terms of the type calculus to a rule of Modus Ponens for ⇒. It also has λ abstraction, which is the converse of **Application** and corresponds to a ⇒ introduction rule.

(6.1) **Application:**

$$\frac{\Delta, C \vdash \lambda x\phi[t]\colon \beta,\quad \Delta, C \vdash \lambda x\phi\colon \alpha \Rightarrow \beta,\quad \Delta, C \vdash t\colon \alpha}{\Delta, C \vdash \phi[t/x]\colon \beta}$$

(6.2) **Abstraction:**

$$\frac{\Delta, C \vdash x\colon \alpha,\ \ \Delta, C \vdash\ \ t\colon \beta}{\Delta, C \vdash \lambda xt\colon \alpha \Rightarrow \beta}$$

For applications whose typing contexts are not locally satisfied, but are consistent with the types available in the type semilattice from the lexicon (cf. Pustejovsky (1995),), we have the rule of *Type Accommodation*. This third rule covers what results in type unification, in which a supertype can unify with a subtype yielding the subtype as the result.

For applications whose typing context does not permit application, but is such that the types assigned are consistent in the sense that they have a meet that is not ⊥ in the type semilattice from the lexicon (cf. Pustejovsky (1995), Copestake and Briscoe (1995)), we have rules of **Type Accommodation**. This rule covers what results in type unification, in which a supertype can unify with a subtype yielding the subtype as the result. These rules make semantic sense; they are a way of combining consistent typing restrictions on objects from predicate and argument.[1]

(6.3) **Type Accommodation**

$$\frac{\Delta, C \vdash t[t']: \bot,\ \ \Delta, C \vdash t: \alpha \Rightarrow \gamma, \Delta, C \vdash t': \beta,\ \Delta \vdash \alpha \sqcap^{-} \beta \neq \bot}{\Delta, C(\frac{t:(\alpha \sqcap^{-} \beta) \Rightarrow \gamma}{t:\alpha \Rightarrow \gamma})(\frac{t':\alpha \sqcap^{-} \beta}{t':\beta}) \vdash t[t']: \gamma}$$

This is a dynamic rule. With Type Accommodation, we reset the type assignment, revising it to reveal the effects of accumulating the consistent type restrictions in predication. We can prove that such a type revision exists by induction on the type of t'.

These are the basic rules for computing with types.

6.3 • Types in TCL

I now turn to the first of our new complex types • and introduce the Exploitation and very limited introduction rules for such types. Exploitation rules tell us how to use constituent types of a complex type when the typing context of a predicational context demands it, while introduction rules tell us when contexts allow the coercion of a new complex type. Asher and Pustejovsky (2004) provide an extensive discussion of the • introduction and exploitation rules. The •-type, as our semantic analysis and rules show, resemble conjunctive types and our natural deduction rules for exploiting them and will resemble something like conjunction elimination at the type level. But the rules for • are quite a bit more complicated than the rules for simple conjunctive types, because they add new variables with new types as required by STA In the section where we examined models of • types, we saw that the introduction of new variables via the Separate Terms Axiom (STA) to given a coherent analysis of • types. The complexity of the type shifting rules thus reflect the theoretical analysis and model of • that we developed in the previous chapter.

[1] See Barr and Wells (1990) for a discussion of unification within a category theoretic framework.

These changes in the type context then give rise to changes in the logical form itself via separate *transfer rules*. Transfer rules take us from the type level, which is concerned with predication and hence how the logical form is ”put together” to the level of information content, which is concerned with the nature of the relations and objects involved in the predication. The transfer rules encode an information flow from the level of information packaging to the level of information content, or from types to formulas.[2]

The • and other complex type exploitation rules may shift the type on a given variable, as we shall see. This is something that the simple rules of TCL already do, but there is an important difference between • Exploitation and say Type Accommodation. The latter rule simply add information about the type of a given term by assigning it a more specific type. That is, Accommodation moves the type of a term t from a type α to a type β where $\beta \sqsubseteq \alpha$. Given the meaning of the subtyping relation, this means of course that t is still of type α; we just know now that it is of a specific subtype of α. The typeshifts involved with • Exploitation are different because they move us from $t : \alpha \bullet \beta$ to $t : \alpha$ say. And we've seen that $\alpha \bullet \beta \sqcap \alpha = \bot$. Thus if we have some predicate *phi* one of whose arguments is x with a basic typing $x : \alpha \bullet \beta$ given in virtue of its being an argument of ϕ, then to shift to $x : \alpha$ will make the semantics of the predicate inconsistent. For example if we have a formula book(x) where we understand books to be of $\textsc{p} \bullet \textsc{i}$ type and shift x to $\textsc{p}$ type, we're violating the semantics of the word. Thus, whatever type shifts we do, we must not violate the basic types for predicates given in the lexicon. This principle is encoded in the following hypothesis:

(6.4) **Lexical Primacy Hypothesis**: Type shifts do not violate the basic types given in the lexicon.

Let us examine what is involved when we want to predicate something of a •-type that forces some sort of type shift. Consider, for example, the compositional interpretation of the noun phrase in (6.5).

(6.5) a heavy book

The interpretation of interest is the predication of being heavy to the book *qua* physical object. Let us suppose that the adjective *heavy* is understood as an intersective adjective. The lexical information yields the lambda term in (6.6) with its

[2] Asher and Pustejovsky (2001) proposed rules for manipulating • types. This version of TCL differs from Asher and Pustejovsky by separating out the contributions of the type logic to logical form from the transfer rules, as we'll see below.

associated type along with the types of the various subterms. I display the typing context to the left of the $\vdash$ even though this information is also derivable from Δ. The type P below abbreviates the type PHYSICAL-OBJECT.

(6.6) $\Delta, \langle x : \textsc{p},\ \ P\colon \textsc{p} \Rightarrow \textsc{t}\rangle\ \vdash\ \lambda P \lambda x(\text{heavy}(x) \wedge P(x))\colon (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow (\textsc{p} \Rightarrow \textsc{t})$

Since P has the type of an arrow from PHYSICAL-OBJECT to propositions, the term for the adjective phrase itself has the type assigned to it on the right of the $\vdash$. This must combine with the semantic expression for *book* to create a common noun phrase or NP in the DP analysis of syntax that will then combine with the determiner.

The word *book* introduces a lambda term $\lambda v\text{book}(v)$ where the variable v is conventionally determined to range over objects with both a physical and an informational aspect. So we have the following logical form and typing context.

(6.7) $\Delta, \langle v\colon \textsc{p} \bullet \textsc{i}\rangle\ \vdash\ \lambda v\text{book}(v)\colon \textsc{p} \Rightarrow \textsc{t}$

where $p \bullet i$ is short for the type PHYSICAL-OBJECT • INFORMATION. So the term $\lambda v\text{book}(v)$ itself has type $(\textsc{p} \bullet \textsc{i})\ \Rightarrow \textsc{t}$. Putting all this information together following the syntax of this DP, we get the following term with its type assignment.

(6.8) $\Delta, \langle x\colon \textsc{p}, P\colon \textsc{p} \Rightarrow \textsc{t},\ \ v\colon \textsc{p} \bullet \textsc{i}\rangle\ \vdash\ \lambda P \lambda x(\text{heavy}(x) \wedge P(x))[\lambda v\text{book}(v)]$

Now, however, this presents us with a type clash between the adjective's type and the noun's type; in effect Δ and the typing context specified yields the absurd type for $\lambda P \lambda x(\text{heavy}(x) \wedge P(x))[\lambda v\text{book}(v)]$. We cannot combine these two lambda terms via lambda conversion because the type of the lambda abstracted variable P and the term that is to replace P don't match. Nor are they subtypes of each other, given our analysis of •. So we can't use type Accommodation.

Three questions arise in the context of this mismatch. First, should we make a type adjustment? If so, where should the type adjustment in this construction take place, on the type of the adjective itself, on the noun, or on some lower variable? Finally, what sort of type adjustment should be made?

The first question has an obvious answer: since a phrase like *heavy book* is clearly felicitous, some sort of type adjustment should be made to allow lambda conversion to take place so as to construct a logical form for the NP. So we must make a type adjustment somehow. But where? Should the type adjustment in this construction take place, on the type of the adjective itself, on the noun, or on some lower variable?[3] Finally, what sort of type adjustment should be made?

For the second question there also seems to be a principled answer in the Head Typing Principle, formulated in the following fashion in Asher and Pustejovsky (2004).

(6.9) **Head Typing Principle:** Given a compositional environment X with constituents A and B, and a type assignment C such that $C \vdash A{:}\,\alpha,\ B{:}\,\beta$ with $\alpha \sqcap \beta = \bot$, then if A is the syntactic, lexical head in the environment, then the typing of A must be preserved in any composition rule for A and B producing a type for X

The Head Typing Principle answers our second question, as to where the type shift should be made: the head type in the construction should remain. When combining maximal projections of lexical heads together with functional heads, it will always be the lexical head's type that will percolate up to the higher construction by using Type Accommodation. This means that in the case of (6.5), we should adjust the adjective's type, given that the noun is the relative head in the construction. Similarly, when we combine a DP in object position with a governing verb to form a VP or with a governing preposition to form a PP, we want the verb's or preposition's categorization to affect the way the DP, and its constituent NP, is interpreted, given our principle that the head of the category should win out. For subjects of a sentence, given the Head Typing Principle, we need to establish what the head of the IP is. If we take standard $\bar{X}$-syntax as our guide, it is the inflection node which introduces an event to saturate the VP, which is its complement. By Type Accommodation, the result will then have the type of the VP. So the Head Typing Principle tells us that we must change the type of the subject DP in order for it to conform to the typing of the I'.[4] it appears as though the VP's type will win out, forcing us to change the type of the subject if there is a type clash. Finally, for coordinate constructions, the Head Typing Principle doesn't determine how types should adjust. But if the type of the coordinated VP is distinct from that for the types of the coordinate VPs then we may expect the subject DP that

[3] Classic GL analyses (e.g., Pustejovsky, 1995) have argued that adjectival subselection selects for a particular qualia role or the corresponding type for a quale within the feature structure of the nominal semantics. That is, they are typed to modify the particular qualia role of the noun in a specific construction. We compare this analysis to the present one below.

[4] Results are largely equivalent if we choose HPSG as our syntactic guide; there the verb will be the lexical head and will once again force us to change the NP's type.

enters into a predication relation with the coordinate VP may give a type to the coordination that will distribute over the coordinated VPs.

Now what should the adjustments be so that we get a variable of the appropriate type? If we go back to our metaphysical underpinnings, then what complex types allow us to do is to predicate properties of aspects of individuals. But if an aspect of a thing exists, then the thing itself must exist as well; in this respect as well as others, aspects differ from parts. So in retyping a variable to represent an aspect of a thing, we should also have a variable representing the thing itself as demanded by STA, and we need to make sure that we link the variable representing the aspect to the variable representing the thing via our O-Elab (object elaboration) relation. In other words we need to *exploit* the complex type P • I by introducing a new term. One of our terms must be of type PHYSICAL so that it can combine with the adjective and the other must be of type P • I and remain the argument of *book*.

According to the Head Typing Principle, the type of *book* should project up to the type of the lambda term associated with the DP. Since *heavy* selects for the type P, if we wish to preserve the type of *book* when we combine it with the adjective, we have several options. The first involves shifting the type of the adjectival phrase so that it will match the type of the noun phrase. But this would require a rule of • introduction, which is not safe in the sense that the move from having something of type α to then of type $\alpha \bullet \beta$ doesn't look as though it preserves truth in general. We could change the logical forms for nouns, but that isn't that attractive either.

What we need to do is to attend to the implementation of STA. STA tells us that we need separate terms associated the • type and for the component types. But it doesn't tell us which type the original variable with the complex type should have. In this case, we need to keep the • type of the λ bound variable associated with *book* and introduce a variable of type P, which can then combine with the adjective. This means that where the head of the construction has the term of complex type, we need to implement STA in a particular way—namely if a typing context C types a variable or constant x with a • type and that term is part of the Head of the construction, then it retains its type but a new variable is introduced with a simpler type.

I formalize this as a rule about TCL deductions; it is a rule that allows us to shift the type context—in particular, expand it, when: (i) a variable or constant of • type is present, with a new variable z that has type α or alternatively type β (which I write $y\colon \alpha(\beta)$), $\underline{\text{and}}$ (ii) the term in which the bullet typed constant or variable is present does not have a consistent type. Thus, the type shifting rule can come into force only when the normal rule of application cannot apply.

- • Exploitation Rule (• Exploitation):

$$\frac{\Delta, C \vdash t\colon \bot,\quad C \vdash x\colon \alpha \bullet \beta}{\Delta, C + \langle z\colon \alpha(\beta)\rangle \vdash t}$$

Clearly if C exists and is a consistent typing context, then so is $C + \langle z\colon \alpha(\beta)\rangle$. The categorial semantics assures us of inhabitants in α and β if there are inhabitants of $\alpha \bullet \beta$.

Matters cannot remain here, however. The new variables z has to be linked with the variable v involving book in the logical form. Further, we must relate the two variables to say that one denotes a trope of the other. Given our discussion of non-functionality with respect to objects of constituent and • types, we must introduce a new term for the physical aspect of *book* and say that it is a physical trope of the book. I use the *O-Elab* relation for that. But where do we introduce this information? To get the truth conditions right, it must be on the smallest term that is not in the head which is responsible for the type clash —i.e. $\lambda P \lambda x(\text{heavy}(x) \wedge P(x))$. So what we write is a function on the adjective meaning. This function should say that there is a physical aspect of something with a complex type that can combine with the adjective *heavy*; that is, for (6.5), we want to say that it is the physical aspect of the book that is heavy, but we don't want this type to percolate up from the Adjectival Phrase into the main predication involving the common noun phrase (NP) let alone the whole determiner phrase (DP). So the variable for the physical aspect must be existentially quantified over in our function. Our function is a lambda expression taking the adjective's logical form as an argument, and it returns an expression ready to combine with the term that has the complex type. It also uses the newly introduced variable z with its type $\textsc{p}$. We redo now the derivation starting with the information in Δ and adding type constraints for z guaranteed by • Exploitation—i.e., $\mathrm{z}\colon \textsc{p}$ and add a typing for the variables in the function term (all but one of these will disappear during the application of this function term to its argument from the original derivation).

(6.10) $\Delta, \langle \mathcal{P}\colon (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow (\textsc{p} \bullet \textsc{i} \Rightarrow \textsc{t}),\ Q\colon \textsc{p} \bullet \textsc{i} \Rightarrow \textsc{t},\ \mathrm{z} : \textsc{p} \bullet \textsc{i},\ z'\colon \textsc{p}\rangle\ \vdash$
$\lambda\mathcal{P}\lambda Q\lambda w\exists \mathrm{z}[\mathcal{P}(\lambda y \text{O-elab}(y, w) \wedge Q(w))](\mathrm{z})\colon \textsc{type}(\mathcal{P}) \Rightarrow (\textsc{type}(Q) \Rightarrow ((\textsc{p}\bullet$
$\textsc{i}) \Rightarrow \textsc{t}))$

We now combine this with the adjective's logical form and redo the derivation. Adding the types for the adjective meaning , we have :

(6.11) $\Delta, \langle \mathcal{P}\colon (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow (\textsc{p}\bullet\textsc{i} \Rightarrow \textsc{t}),\ Q\colon \textsc{p}\bullet\textsc{i} \Rightarrow \textsc{t},\ w : \textsc{p}\bullet\textsc{i},\ \ x, y\colon \textsc{p},\ P\colon \textsc{p} \Rightarrow \textsc{t}\rangle \vdash \lambda\mathcal{P}\lambda Q\lambda w\exists \mathrm{z}[\mathcal{P}(\lambda y(\text{O-elab}(y, w)\wedge Q(w)))](\mathrm{z})[\lambda P\lambda x(\text{heavy}(x)\wedge P(x))]\colon \textsc{type}(Q) \Rightarrow ((\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t})$

A use of Application yields:

(6.12) $\Delta, \langle Q\colon \textsc{p}\bullet\textsc{i} \Rightarrow \textsc{t},\ w : \textsc{p}\bullet\textsc{i},\ \ x, y\colon \textsc{p},\ P\colon \textsc{p} \Rightarrow \textsc{t}\rangle \vdash \lambda Q\lambda w\exists \mathrm{z}(\lambda P\lambda x(\text{heavy}(x)\wedge P(x))(\lambda y(\text{O-elab}(y, w) \wedge Q(w))))(z))\colon \textsc{type}(Q) \Rightarrow ((\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t})$

Another three uses of Application yield:

(6.13) $\Delta, \langle Q\colon \textsc{p} \bullet \textsc{i} \Rightarrow \textsc{t},\ \ w\colon \textsc{p} \bullet \textsc{i}\rangle \vdash \lambda Q\lambda w\exists \mathrm{z}(\text{heavy}(z) \wedge \text{O-elab}(z, w) \wedge Q(w))\colon \textsc{type}(Q) \Rightarrow ((\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t})$

We can now apply (6.13) to the term for common noun phrase without any difficulty, yielding

(6.14) $\Delta, \langle w\colon \textsc{p} \bullet \textsc{i}\rangle \vdash \lambda w\exists \mathrm{z}(\text{book}(w) \wedge \text{heavy}(z) \wedge \text{O-Elab}(z, w))\colon ((\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t}$

This now combines with the determiner meaning for *a*, $\lambda Q\lambda P\exists x_1(Q(x_1) \wedge P(x_1))$ with the type context $\langle x_1\colon e,\ \ P, Q\colon e \Rightarrow \textsc{t}\rangle$ via Accommodation on x_1, setting the type of x_1 to $\textsc{p} \bullet \textsc{i}$. We now get the correct logical form with the right type:[5]

(6.15) $\Delta, \langle x_1\colon \textsc{p} \bullet \textsc{i},\ \ \ Q\colon (\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t}\rangle \vdash \lambda Q\exists x_1\exists \mathrm{z}(\text{book}(x_1) \wedge \text{heavy}(z) \wedge \text{O-Elab}(z, x_1) \wedge Q(x_1))\colon ((\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}$

Let us look at another representative example of a type mismatch involving a dot object, where the subject is a complex type and the predicate selects for one of the constituent types. Consider the predication in (6.16) below.

(6.16) The book is heavy.

The Head Typing Principle once again is our guide. It tells us that we have to change the type of the subject DP, while the type of the VP remains unchanged. Confirmation of the Head Typing Principle comes from this implication about changing the type of the DP. To see why, suppose that the quantification in the DP in (6.16), and more importantly in (6.17) is over dot objects. Suppose that we assert (6.17) in a context in which some books from the library have been stolen and others borrowed.

[5] Note that I have simplified the determiner's contribution ot logical form, avoiding irrelevant complications about presuppositions.

(6.17) Every book is now back in the library.

Suppose in addition that the library contains five copies of *Anna Karenina*, six copies of *The Possessed* and four copies of *Madame Bovary* but only one copy of each has been returned. Universally quantifying over objects of • type with the informational counting principle implies that (6.17) is true in that case. Indeed dot objects are difficult to "count", but we can individuate at least some of them, viz. books, in terms of the individuation conditions of either constituent type. Our intuitions, however, dictate that (6.17) is neither ambiguous nor indeterminate but false in this context. To avoid such "sloppy" individuation conditions, we need to resort to simple types. The Head Principle dictates that the lexical typing from the VP must percolate up to the main predication. So we need to type the DP in (6.17) so that it quantifies over physical objects. If we quantify over every physical book in the library, then this will make (6.17) false in the context we have specified.

We have seen that we need to shift the type of the logical form of the DP so that it has a simple type. We could do this in several ways. But the Lexical Primacy Hypothesis constrains our options. We cannot change the type of the variable associated with *book* in (6.17) without violating that constraint. So what we must do is to change the head variable in the DP. So, if our DP looks like

(6.18) $\lambda P \forall x(\psi \rightarrow P(x))$

where $x\colon \alpha \bullet \beta$, then we need to use our new variable introduced by the • Exploitation Rule. So for instance if the new variable is y with the type α. We replace (6.18) with (6.19):

(6.19) $\lambda P \forall y(\exists x(\psi \wedge \text{O-Elab}(x, y) \rightarrow P(y))$

This is a rather different strategy for integrating the new variable introduced by the Exploitation Rule from the one we used for (6.5), but it makes sense that in virtue of the Head Typing Principle. Our transfer rule should be sensitive as to whether the • typed variable is in the head of the construction or not.

Let's now look this in more detail. To compute a logical form for (6.16), we first combine the term $\lambda v \text{book}(v), \langle v : \textsc{p} \bullet \textsc{i} \rangle$
with the determiner's logical form. I omit Δ and the sequent form, giving just the formula and its typing context, in the interests of readability. $\lambda Q \lambda P \exists x(Q(x) \wedge P(x)), \langle x\colon e, \quad Q, P\colon e \Rightarrow \textsc{t}$. With Accommodation and Application, we get for *the book*:

(6.20) $\lambda P \exists x(\text{book}(x) \wedge P(x)), \langle x\colon \textsc{p} \bullet \textsc{i},\ P\colon (\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t} \rangle$

(6.21) The logical form for *is heavy* is the following: $\lambda u \mathrm{heavy}(u)$, $\langle u\colon \textsc{p}\rangle$

(6.22) Putting these two together via syntax we get:
$\lambda P \exists x(\mathrm{book}(x) \wedge P(x))[\lambda u \mathrm{heavy}(u)]$, $\langle P\colon (\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t}, x\colon \textsc{p} \bullet \textsc{i}, u\colon \textsc{p}\rangle$

(6.23) By •-Exploitation: $\lambda P \exists x(\mathrm{book}(x) \wedge P(x))[\lambda u \mathrm{heavy}(u)]$,
$\langle P\colon \textsc{p} \Rightarrow \textsc{t}, u\colon \textsc{p},\ \ v\colon \textsc{p}, x\colon \textsc{p} \bullet \textsc{i}\rangle$

Let's pause and take stock. We have arrived at a point in the derivation where we must once again link the newly introduced variable with the original one that has been shifted. It's obvious that we have to change the formula for the quantifier—ie. once again the non head construction, because *book* is applying to a variable of the inappropriate type; the argument of *book* is specified in its lexical entry as being of • type and we must respect this lexical information. Once again we introduce the information here about O-elab on the smallest term responsible for the typing of x, which here is not in the head. That must be the term for *book* when we look back at the derivation. At this point then we must apply the appropriate transfer operator in (6.24) to *book* with the following result in (6.25), which yields (6.26) after λ reduction:

(6.24) $\lambda P \lambda v \exists w(P(w) \wedge \text{O-Elab}(v, w))$, $\langle P\colon (\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t},\ w\colon \textsc{p} \bullet \textsc{i},\ v\colon \textsc{p}\rangle$

(6.25) $\lambda P \lambda v \exists w(P(w) \wedge \text{O-Elab}(v, w))[\lambda v_1\, \mathrm{book}(v_1)]$, $\langle P\colon (\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t}, w, v_1 : \textsc{p} \bullet \textsc{i},\ v\colon \textsc{p}\rangle$

(6.26) $\lambda v \exists w(\mathrm{book}(w) \wedge \text{O-Elab}(v, w))$, $\langle w : \textsc{p} \bullet \textsc{i},\ v\colon \textsc{p}\rangle$

We can now combine this term with the determiner to get back a new DP meaning that is to combine with the predicate meaning as in (6.27). Note that the type of the higher order variable P will have changed because of the shift in type to x in the resetting of the type context prompted by •-Exploitation NH. That is, $P : \textsc{p} \Rightarrow \textsc{t}$ in the updated context. This allows us to produce the final logical form for (6.16) in (6.28).

(6.27) $\lambda P \exists x(\exists w(\mathrm{book}(w) \wedge \text{O-Elab}(x, w)) \wedge P(x))[\lambda u \mathrm{heavy}(u)]$, $C + \langle w : \textsc{p} \bullet \textsc{i},\ u\colon \textsc{p}\rangle$

(6.28) $\exists x \exists v(\mathrm{book}(v) \wedge \text{O-Elab}(x, v)) \wedge \mathrm{heavy}(x))$, $\langle x\colon \textsc{p}, v\colon \textsc{p} \bullet \textsc{i}\rangle$

Type conflicts involving a complex dot type and a constituent type might in principle occur between terms of arbitrarily complex type. This would make it very difficult to state transfer rules for the language of logical form. However, a few observations about the nature of predication show us that the task is not at all hopeless. Generalizing from these two examples, we see that transfer information always applies to a term that is not part of the head in the predication. But transfer typically involves only a subformula of the non head term involved in the predication. We must add the linking information between the two variables introduced by the • exploitation rules at the point at which the original typing of the variable of complex type occurs. This typing can be introduced only by five lexical classes: nouns, prepositions, verbs, adjectives and adverbs. In addition, the Head Typing Principle tells us to add the transfer information only in those formulas that are not in the head. Since these formulas are not in the head, they cannot change the type of the head with which they combine. Hence, we can get by introducing formulas encoding the transfer just over expressions that are a subtype of $e \Rightarrow \textsc{t}$ (the case of nouns or intransitive verbs and predicative phrases), of type $(e \Rightarrow \textsc{t}) \Rightarrow (e \Rightarrow \textsc{t})$ (the case of adjectives, prepositions and adverbs) or of type $((e \Rightarrow \textsc{t}) \Rightarrow (e \Rightarrow \textsc{t})$ (the case of transitive verbs). Our transfer information will take the form of functions that transform these types into types that will fit eventually with the head of the construction—replacing the complex types with simple constituent ones. These functions correspond to the object level translations of our functors over categories.

The approach taken to transfer here requires us to go back up the derivation tree to use the transfer operators and redo the derivation. This clearly adds complexity to the system but is needed to preserve the Head Typing Principle. As we'll see, the gain in complexity is not all that much. It's also needed to restore a consistent type to a term to which we have applied • Exploitation. • Exploitation is triggered in part by a failure to assign a consistent type, but while the type rule introduces the wherewithal to repair the type inconsistency, it is the transfer rule that actually eliminates the inconsistency, and it does so in a minimal and monotonic way.

Let's now formalize our tranfer rules. The transfer rules rely on the presence of an application of one of the type shifting • exploitation rules or more generally on the derivation of one sequent from another; $\Delta, C' \vdash t \mapsto \Delta, C \vdash t$ means that there was a derivation from the sequent $\Delta, C' \vdash t$ to the sequent $\Delta, C \vdash t$. **Non**

Head Transfer:

Suppose (i) $\Delta, C' \vdash t\colon \bot \mapsto \Delta, C \vdash t\colon \bot$, (ii) $C \vdash x\colon \alpha \bullet \beta, C \vdash z\colon \alpha(\beta),\quad C' \nvdash z\colon \alpha(\beta))$, (iii) $\Delta \vdash x \notin$ Head(t).
Let ϕ be the smallest term in Δ responsible for the typing of x.

Then:

- if $\phi\colon (\alpha \bullet \beta) \Rightarrow \textsc{t}$, then:
  $\Delta, \langle P\colon (\alpha{\bullet}\beta) \Rightarrow \textsc{t},\ \mathrm{z}\colon \alpha(\beta),\ v\colon \alpha{\bullet}\beta\rangle \ \vdash\ \lambda P \lambda \mathrm{z} \exists v (P(v) \wedge \text{O-elab}(\mathrm{z}, v))[\phi]\colon \alpha(\beta) \Rightarrow \textsc{t}$

- if $\phi\colon ((\alpha \bullet \beta) \Rightarrow \textsc{t}) \Rightarrow (\alpha \bullet \beta \Rightarrow \textsc{t})$, then:
  $\Delta, C'' \ \vdash\ (\lambda \mathcal{P} \lambda Q \lambda \mathrm{z} \exists v (\mathcal{P}(\lambda u (\text{O-elab}(v, u) \wedge Q(v)))(\mathrm{z}))[\phi]\colon (\alpha(\beta) \Rightarrow \textsc{t}) \Rightarrow (\alpha(\beta) \Rightarrow \textsc{t})$
  where $C'' = \langle \mathcal{P}\colon ((\alpha{\bullet}\beta) \Rightarrow \textsc{t}) \Rightarrow ((\alpha{\bullet}\beta) \Rightarrow \textsc{t}),\ Q\colon \alpha(\beta) \Rightarrow \textsc{t}, \mathrm{z}\colon \alpha(\beta), u, v\colon \alpha \bullet \beta\rangle$

- if $\phi\colon ((\alpha \bullet \beta) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}) \Rightarrow (e \Rightarrow \textsc{t})$, then
  $\Delta, C'' \vdash \lambda \Pi \lambda \mathcal{P}' \lambda u \mathcal{P}' [\lambda \mathrm{z} [\Pi (\lambda Q \exists v (Q(\mathrm{z}) \wedge \text{O-elab}(\mathrm{z}, v)))(\phi)(u)]]\colon ((\alpha(\beta) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}) \Rightarrow (e \Rightarrow \textsc{t})$,
  where $C'' = \langle \Pi\colon \textsc{type}(\phi) \Rightarrow (((\alpha(\beta) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}) \Rightarrow (e \Rightarrow \textsc{t})),\ \mathcal{P}'\colon (\alpha(\beta) \Rightarrow \textsc{t}) \Rightarrow \textsc{t},\ Q\colon (\alpha \bullet \beta) \Rightarrow \textsc{t},\ \mathrm{z}\colon \alpha(\beta),\ v\colon \alpha \bullet \beta,\ u\colon e\rangle$

The other transfer rule we need is where the • type is in the Head:

Head Transfer: (i) $\Delta, C' \vdash t\colon \bot \mapsto \Delta, C \vdash t\colon \bot$, (ii) $C \vdash x\colon \alpha \bullet \beta, C \vdash \mathrm{z}\colon \alpha(\beta),\ \ C' \nvdash \mathrm{z}\colon \alpha(\beta))$, (iii) $\Delta \vdash x \in$ Head(t).
Let ϕ be the smallest term in Δ responsible for the type clash that led to the shift from C' to C.
Then:

- If $\phi\colon \alpha(\beta) \Rightarrow \textsc{t}$,
  $\Delta, \langle P\colon \alpha(\beta) \Rightarrow \textsc{t},\ v\colon \alpha \bullet \beta,\ \mathrm{z}\colon \alpha(\beta)\rangle) \ \vdash\ \lambda P \lambda v \exists \mathrm{z} (P(\mathrm{z}) \wedge \text{O-elab}(\mathrm{z}, v))[\phi]\colon (\alpha \bullet \beta) \Rightarrow \textsc{t}$

- If $\phi\colon (\alpha(\beta) \Rightarrow \textsc{t}) \Rightarrow (\alpha(\beta) \Rightarrow \textsc{t}))$, then
  $\Delta, C'' \ \vdash\ \lambda \mathcal{P} \lambda Q \lambda v \exists \mathrm{z} (\mathcal{P}(\lambda u (\text{O-elab}(u, v) \wedge Q(v)))(\mathrm{z}))[\phi]\colon ((\alpha \bullet \beta) \Rightarrow \textsc{t}) \Rightarrow ((\alpha \bullet \beta) \Rightarrow \textsc{t}))$,
  where $C'' = \langle \mathcal{P}\colon (\alpha(\beta) \Rightarrow \textsc{t}) \Rightarrow (\alpha(\beta) \Rightarrow \textsc{t}),\ Q\colon \alpha \bullet \beta \Rightarrow \textsc{t}, u\colon \alpha(\beta), v\colon \alpha \bullet \beta\rangle,\ \mathrm{z}\colon \alpha(\beta)\rangle$,

- if $\phi\colon ((\alpha(\beta) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}) \Rightarrow (e \Rightarrow \textsc{t})$, then
  $\Delta, C'' \ \vdash \lambda \Pi \lambda \mathcal{P}' \lambda u (\mathcal{P}' [\lambda v [\Pi (\lambda Q \exists \mathrm{z} (Q(\mathrm{z}) \wedge \text{O-elab}(\mathrm{z}, v)))]][\phi](u)\colon (((\alpha \bullet \beta) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}) \Rightarrow (e \Rightarrow \textsc{t})$
  where $C'' = \langle \Pi\colon \textsc{type}(\phi) \Rightarrow (((\alpha \bullet \beta) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}) \Rightarrow (e \Rightarrow \textsc{t})),$
  $\mathcal{P}'\colon (\alpha \bullet \beta) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, Q\colon \alpha(\beta) \Rightarrow \textsc{t},\ v\colon \alpha \bullet \beta,\ \mathrm{z}\colon \alpha(\beta),\ u\colon e\rangle$

The transfer rules fix the interpretation of the new variables introduced in the type context by the • Exploitation rule. We've already seen that the transformed type contexts postulated by these rules must exist if the input contexts exist. Our transfer rules fix the relation between the variables whose types stand in a projection relation ($\alpha \bullet \beta \mapsto \alpha(\beta)$) to be the one provided for by the categorial semantics. Given types α and β, then there is a pullback $\alpha \times_{\beta} \beta$ and $\alpha \times_{\alpha} \beta$ with projections onto α and β. And so for any object of the complex type, there will be aspects of the simple constituent types. This is precisely what the transfer function together with the • exploitation rules determine. Given our model of an object of complex type modelled as a set of pairs all agreeing on a common element as defined by the pullback, our transfer functions say that the aspects are related to the object in the appropriate way and are thus sound

6.3.1 Spurious Ambiguity and Other Problems

There is perhaps a question as to why I formulated exploitation rule so that it is restricted to cases where the typing context shows that a predication cannot proceed—that is, the term t has type $\perp$. Derivations in TCL of λ free formulas correspond to readings of semantically and syntactically well formed clauses of a natural language like English. Without the restriction on the exploitation rule, is that our rule would not be constrained enough, and as a result we could derive readings for natural expressions that are not there. The system provides, in other words, for spurious ambiguities.

Here is an example of the problem. Consider the noun phrase

(6.29) a readable book

If TCL's • exploitation rule was not guarded by the requirement that there is an inconsistency in the typing context, it would permit the derivation of a reading of (6.29) where the existential quantifier introduced by the indefinite ranges over physical objects only, a reading where the quantifier ranges over informational objects only and a reading where it ranges over objects of type $\text{P} \bullet \text{I}$. But the first two readings aren't there; the theory thus predicts that (6.29) is ambiguous and wrongly so.

TCL's • exploitation rule precludes such a derivation and brings out an important feature of type shifting in natural language semantics: conservativity. These exploitations are triggered as a last resort in order to rescue a predication that would otherwise fail because of a type clash. In general in computing a litteral meaning for a clause (or what passes for such), we want to avoid changes in the lexical entries during the composition process unless we have to. The guarded exploitation rule eliminates the spurious ambiguities and obeys the principle of conservativity just observed.

These rules improve upon the rules of Asher and Pustejovsky (2004) considerably. They are simpler and demonstrably sound. They enable us to prove versions of the substitution rules of Asher and Pustejovsky (2004). For instance, where δ, in t is responsible for the original typing of x (x is not in the head of the construction) we have the following lemma (this was a primitive rule of Asher and Pustejovsky (2004)):

Lemma 4

•-Exploitation Transfer I (•E Tr I):

$$\frac{\Delta, C\frac{x{:}\alpha(\beta)}{x{:}\alpha\bullet\beta} + v{:}\,\alpha \bullet \beta \ \vdash \ t}{\Delta, C\frac{x{:}\alpha(\beta)}{x{:}\alpha\bullet\beta} + v{:}\,\alpha \bullet \beta \ \vdash \ t[\frac{\delta(x)[\frac{v}{x}]\wedge \textit{O-Elab}(x,v))}{\delta(x)}]}$$

In words, we substitute in for $\delta(x)$, $\exists v\Delta(v) \wedge$ O-Elab(x, v) or $\lambda v\delta(v) \wedge$ O-Elab(x, v), depending on whether we are at a maximal projection or not.

This lemma reveals a problem concerning proper names that are supposed to undergo type shifts. An example of this is something like

(6.30) *War and Peace* is over 500 pages long.

Assuming that *is over 500 pages long* is a property of the book in its physical aspect and that the name *War and Peace* is of P • I type, then the Head Principle says to exploit the complex type and use Transfer. How we carry this out depends exactly on how we translate proper names into logical form. If we take the standard Montegovian treatement and say that the proper name introduces a quantifier $\lambda PP(wp)$, where 'wp' is the constant introduced by the name, then that will force us to shift wp itself. In this case there is no Δ aside from the constant itself that is responsible for the typing of the term. If we treat a name presuppositionally, as simply introducing a variable into the proferred logical from, we have the same problem. But we do not want to shift the type of *War and Peace* itself, only what it contributes to the predicate. To do otherwise would violate the Lexical Primacy Hypothesis. We also don't want to shift the type of the predicate, which would also violate the Lexical Primacy Hypothesis. So what should we do? It seems that we must assimilate this case to one in which we do not shift the type of wp itself. Of course if we considered the contribution of a name to logical form to be something more like what Discourse Representation Theory (Kamp and Reyle (1993)) suggests— $\lambda P \exists x(x = wp \wedge P(x))$, the problem is easily solved. In this case δ must be $x = wp$. In the case of (6.30), Transfer would have the following effect after we shift types:

(6.30) $\lambda P \exists x \exists v(v = wp \wedge \text{O-Elab}(x, v) \wedge P(x))$

6.3.2 Some Derivations

The rules for TCL presented so far constitute the basic logic of TCL and provide a treatment of complex examples like those given in the quantificational puzzle, as well as the basics of a treatment of copredication.

The rule of •-Exploitation lets us take any modifier of a noun that would force a dot type (the adjective *readable* would be one such example) and apply it to a noun with a simple type that is the constituent of the modifier's type. We could then combine the two together to get a noun phrase of the simple type as required. Thus if we have a sentence such as (6.31) below:

(6.31) John turned off every readable screen,

our Non Head Transfer rule will produce a noun phrase that looks like the following, since the adjective introduces the • type but is not in the head.

(6.32) $\Delta, \langle x\colon \textsc{p}, v\colon \textsc{p}\bullet\textsc{i}\rangle \vdash \lambda x(\exists v(\text{readable}(v)\wedge\text{O-Elab}(x, v)\wedge\text{screen}(x))\colon \textsc{p}Rightarrow\textsc{t}$

When the determiner meaning is applied, we will get a quantification over all physical screens, which is what is intuitively required.

When a • type is exploited and Non Head Transfer is used, the variable with the • type does not percolate up in the predication. In this case quantification over objects of the constituent types always has scope over the quantification over objects of • type.

Is this scope prediction in fact correct? Consider for instance,

(6.33) Three books by Tolstoy are heavy.

Following the derivation above, we would get a logical form for this sentence according to which on a distributive reading there are three physical aspects p_1, p_2, p_3 each of which have to satisfy the formula $\exists v(\text{book* by Tolstoy}(v) \wedge \text{O-Elab}(x, v))$, where $x : \textsc{p}$ and $v : \textsc{p} \bullet \textsc{i}$, and each of which are heavy.[6] Nothing in our semantics

[6] 'book*' is a predicate that can apply either either to individuals or to collections thereof. See Link 1983.

forces the three aspects to be part of the *same* book, as it should be. In fact the opposite is true. The predication leads us to think of the objects of complex type as individuated in a certain way—in particular we are inclined to individuate books physically. If books are individuated physically, then the constraints on O-elab make an interpretation where we have three physical aspects of one book incoherent. In particular it is the partial ordering of aspects and which itself entails that predications pick out aspects of objects of complex type that are maximal that entails the incoherence. Below is repeated the relevaant consequence of the partial ordering principle.

(6.34) $\text{O-elab}(x, y) \wedge \text{O-elab}(z, y) \wedge x, z : \alpha) \rightarrow x = z$

Given (6.34), if we have O-elab(p_1, b) and O-elab(p_2, b), $p_1 = p_2$, which contradicts the meaning of the quantifier. On our semantics of O-elab, this subformula of the logical form of (6.33) can only be satisfied if there is a distinct book (understood physically) for each distinct physical aspect. Though there is a collective reading of the DP (the three books together are heavy), our semantics precludes having a collective reading of the formula in the restrictor of the quantifier.[7] Thus, we end up predicting that (6.33) is true only if there are three distinct books each with its own physical aspect that is heavy. Because of the particular dependency of aspects on the substances to which they belong, there is a quantificational dependency between variables for aspects and variables ranging over the substances of which they are parts.[8]

Yet another instance of •-exploitation concerns one where the complex type/simple type conflict occurs between an expression that has a generalized quantifier as an argument and a generalized quantifier. Given Fact 2, we cannot change the type of verbal predicate; we must change the type of the generalized quantifier An example of this situation is one where a verb types its argument as a physical object but the noun in the complement types its argument as a complex type, say $\text{P} \bullet \text{I}$. Consider for example,

(6.35) John's mother burned the book on magic before he mastered it.

[7] For details on how such distributive and cumulative readings together are possible, see Asher and Wang 2003.

[8] There is also a reading on which the book is individuated informationally, and this allows for the other scope reading on which there is one book with three physical aspects—i.e. three copies of a single informational book by Tolstoy are heavy. This reading is rather forced but I think is there in the right context.

The verb *burn*'s object argument must be a physical object, and as the Head Typing Principle dictates, although the object DP enters the composition with type $\textsc{p} \bullet \textsc{i}$, there must be some way to coerce it into having the right type, to satisfy the typing context and thereby allow the λ-conversion from the verb to go through. The way we do this is to apply •-Exploitation on the generalized quantifier to coerce it into the right type.

Let us look at the details. In (6.35), we see a problem with the typing of the expressions we are trying to compose (recall that $\textsc{p}$ (physical-object) in this context is a subtype of e in the semilattice structured domain of entities, i.e., $\textsc{p} \sqsubseteq e$):

(6.36) $\lambda\mathcal{P}\lambda w\mathcal{P}[\lambda u(\mathrm{burn}(w,u))](\lambda Q\exists x(\mathrm{book}(x)\wedge Q[x]),\ \langle Q\colon (\textsc{p}\bullet\textsc{i}) \Rightarrow \textsc{t},\ \mathcal{P}\colon (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, u\colon p, w\colon \textsc{p},\ x\colon \textsc{p} \bullet \textsc{i}\rangle$

Because we are not changing the sense of the predicate in any way (that is, *burn* should still mean *burn*) it is undesirable to change the type of the variable $\mathcal{P}$ over DP denotations. Rather, we want to change the type of the object itself. Since *burn* is the lexical head of the VP construction, we need to change the type of the DP. By • Exploitation,

(6.37) $\lambda\mathcal{P}(\lambda w\mathcal{P}[\lambda u(\mathrm{burn}(w,u))])[\lambda P\exists x(\mathrm{book}(x)\wedge P[x]),\ \langle\mathcal{P}\colon (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, u\colon p, w\colon \textsc{p},\ P\colon \textsc{p} \Rightarrow \textsc{t}, y : \textsc{p} \bullet \textsc{i}, x : \textsc{p}\rangle$

By Transfer over *book* and redoing the derivation

(6.38) $\lambda\mathcal{P}(\lambda w\mathcal{P}[\lambda u(\mathrm{burn}(w,u))])[\lambda P\exists x\exists y(\mathrm{book}(y)\wedge\text{O-Elab}(x,y)\wedge P[x]),\ \langle\mathcal{P}\colon (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, u\colon p, w\colon \textsc{p},\ P\colon (\textsc{p} \Rightarrow \textsc{t}, y : \textsc{p} \bullet \textsc{i}, x : \textsc{p}\rangle$

Using *Application*, we get the following expression, where c' denotes the typing context derived in (6.37)

(6.39) $\lambda w\ \lambda P\ \exists x\exists y(\mathrm{book}(y) \wedge \text{O-Elab}(x,y) \wedge P[x]))[\lambda u(\ \mathrm{burn}(w,u))],$
c'

We can now continue the λ-reductions with *Application* to get:

(6.40) $\lambda w\exists x\exists y(\mathrm{book}(y) \wedge \text{O-Elab}(x,y) \wedge\ \mathrm{burn}(w,x)),\ c'$

When we apply this to the subject DP, we get the desired reading: namely, that the physical manifestation of the book has been burned, though the dot object book remains for discourse binding.[9]

I now return to the quantificational puzzles. Recall that using different predicates made a difference in how one counted objects under quantification.

[9] Given the Head Typing Principle, we do not need any other rules for such examples. including the Type Shifted versions of • exploitation of Asher and Pustejovsky (2001).

(6.41) a. The student read every book in the library.

b. The student carried off every book in the library.

First let's tackle (6.41a). Below the type $\textsc{p} \bullet \textsc{l}$ is short for the complex type, PHYSICAL-OBJECT • LOCATION, which is the type of libraries. I assume that the preposition *in* types its first argument as physical and takes an argument of $p \bullet \textsc{l}$ as its second argument. So for *book in the library* we get:

(6.42) $[\![\text{book in the library}]\!] = \lambda Q \lambda y \exists x(\text{library}(x) \wedge \text{in}(y, x) \wedge Q(y))[\lambda v \text{book}(v)]$,
$\langle Q : \textsc{p} \Rightarrow \textsc{t},\ x{:}\,\textsc{p} \bullet \textsc{l},\ y{:}\,\textsc{p}, v{:}\,\textsc{p} \bullet \textsc{i}\rangle$

There is obviously a type conflict between *book* and the λ bound variable Q. Thus at this point in our derivation is trouble; for the typing context C and the term t for *book in the library*, $C \vdash t{:}\bot$. But the type conflict can be resolved by • Exploitation on the type of v:

(6.43) $\lambda Q \lambda y \exists x(\text{library}(x) \wedge \text{in}(y, x) \wedge Q(y))[\lambda v \text{book}(v)]$,
$\langle Q : \textsc{p} \Rightarrow \textsc{t},\ x{:}\,\textsc{p} \bullet \textsc{l},\ y{:}\,\textsc{p}, v{:}\,\textsc{p} \bullet \textsc{i}, z{:}\,\textsc{p}\rangle$

Because v is in the head of the construction, we use Transfer Head, which tells us that we use v and a fresh variable with a constituent type within a functor that will take scope over the prepositional phrase meaning. The functor looks like this:

(6.44) $\lambda \mathcal{P} \lambda Q \lambda v' \exists \text{z}(\mathcal{P}(\lambda u(\text{O-elab}(u, v') \wedge Q(v')))(\text{z}), \langle \text{z}{:}\,\textsc{p},\ v'{:}\,\textsc{p} \bullet \textsc{i}\rangle$

Transfer and Type Accommodation allow us to rewrite the term of the non head in the predication as follows:

(6.45) $\lambda Q \lambda v' \exists \text{z} \exists x(\text{library}(x) \wedge \text{in}(z, x) \wedge \text{O-Elab}(z, v') \wedge Q(v'))$, $\langle Q{:}\,(\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t}, v'{:}\,\textsc{p} \bullet \textsc{i}, z{:}\,\textsc{p}, x{:}\,\textsc{p} \bullet \textsc{l}\rangle$

Using (6.45), we can now use Application with the lexical entry for *book* to get (6.46), abbreviating the whole type context as C:

(6.46) $\lambda v' \exists \text{z} \exists x(\text{library}(x) \wedge \text{in}(z, x) \wedge \text{book}(v') \wedge \text{O-Elab}(z, v'))$, C

Now applying the determiner meaning we get:

(6.47) $\lambda P \forall v(\exists \text{z} \exists x(\text{library}(x) \wedge \lambda w \text{in}(z, x) \wedge \text{book}(v) \wedge \text{O-Elab}(z, v)) \rightarrow P(v)), C$

I assume that the first argument of *read* is typed as AGENT (A) and the second argument is typed as $\textsc{p} \bullet \textsc{i}$. This means that the types of the DP (whose λ term we construct by Accommodation and Application) and the transitive verb match and so again by Application we get:

(6.48) $[\![\text{read every book in the library}]\!] =$
$\lambda u[\forall v(\exists z\exists x(\text{library}(x) \wedge \text{in}(z,x) \wedge \text{book}(v) \wedge \text{O-Elab}(z,v)) \rightarrow \text{read}(u,v)),$
$\langle u : \textsc{a}, v\!: \textsc{p} \bullet \textsc{i},\ x\!: \textsc{p} \bullet \textsc{l}, y\!: \textsc{p}\ \rangle$

After application we get the desired quantificational reading for the VP. That is, quantification is over $\textsc{p} \bullet \textsc{i}$ objects, not physical objects only, as desired. We now have to choose the criterion of counting and individuation appropriate to the quantificational domain. If we choose the *i* criterion, which is perhaps more plausible, then we get the reading that the student read every (informational) book of which there is a physical copy in the library. Note that we predict, as seems plausible, that the student could have read every book in the library even without having opened a single physical copy that is in the library. On the other hand, we could also take the criterion of counting and individuation provided by physical objects. This yields the less plausible reading that the student read every physical volume in the library.

Now let us contrast this with the sentence in (6.41b). The derivation is the same down to the application of the transitive verb to its object. By the Head Principle, the type on the verb must win, and *carry off* types its object as physical as in (6.49):

(6.49) $\lambda \mathcal{P} \lambda w \mathcal{P}(\lambda u\ \text{carry-off}(w,u)), \langle w : \textsc{a},\ \mathcal{P}\!: (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, u\!: p\rangle$

Syntax tells us that we must combine this with *book in the library*. I abbreviate the whole typing context again to$C$.

(6.50) $\lambda \mathcal{P} \lambda w \mathcal{P}(\lambda u\ \text{carry-off}(w,u))[\lambda P \forall v(\exists y \exists x(\ \text{library}(x) \wedge\ \text{in}(y,x) \wedge\ \text{book}(v) \wedge$
$\text{O-Elab}(y,v)) \rightarrow P(v)),\ C$

Since the functor provided by *carry off* here is the head of the construction, we must retype and rewrite the λ-term for the DP, more specifically its constituent NP. Use • Exploitation NH on *v*, and the Transfer Non Head rule, we get. C is now replaced by $C\frac{z:\text{P}\bullet\text{I}, v:\text{P}}{v:\text{P}\bullet\text{I}}$. Now by Transfer we can apply the following functor to *book*:

(6.51) $\lambda P \lambda v \exists z(P(z) \wedge \text{O-elab}(v,z)),\ \langle P\!: \textsc{p} \bullet \textsc{i} \Rightarrow \textsc{t},\ z\!: \textsc{p} \bullet \textsc{i}, v\!: \textsc{p}\rangle$

which yields, when applied to the term for *book*:

(6.52) $\lambda v \exists z(\text{book}(z) \wedge \text{O-elab}(v, z)),\ \langle z\colon \textsc{p} \bullet \textsc{i}, v\colon \textsc{p}\rangle$

We can now retrace our steps in the derivation to get back to the point we were earlier:

(6.53) $\lambda\mathcal{P}\lambda w\mathcal{P}(\lambda u\text{carry-off}(w, u))[\lambda P\forall v\exists z\exists y(\exists x(\text{library}(x)\wedge\text{in}(y, x)\wedge\text{book}(z)\wedge \text{O-Elab}(y, z)\wedge\text{O-Elab}(v, z)) \rightarrow P(v))], \langle w : \textsc{a},\ \mathcal{P}\colon (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow \textsc{t},\ u\colon p,\ z\colon \textsc{p}\bullet \textsc{i}, v\colon \textsc{p}, P\colon \textsc{p} \Rightarrow \textsc{t})\rangle$

Now by application we get the following result with the desired quantificational reading, It states that all physical copies of all books that are in the library were carried off.

(6.54) $\lambda w\forall v(\exists x\exists y\exists z\exists v(\text{library}(x)\wedge\text{in}(y, x)\wedge\text{book}(z)\wedge\text{O-Elab}(y, z)\wedge\text{O-Elab}(v, z)) \rightarrow \text{steal}(w, v)),$
$\langle w : a,\ y\colon p, v\colon \textsc{p}, z :: \textsc{p} \bullet \textsc{i},\ x\colon \textsc{p} \bullet \textsc{l}\rangle$

Now this doesn't say that all the copies in the library were stolen though it entails this. We can recover this reading completely by identifying the copies that are in the library with the physical aspects quantified over. How might this take place? If the PP helps to determine the domain of quantification for the DP, then this identification is appropriate and we can simplify (6.54) to the following:

(6.55) $\lambda w\forall v(\exists x\exists z(\text{library}(x)\wedge\text{in}(y, x)\wedge\text{book}(z)\wedge\text{O-Elab}(v, z)) \rightarrow \text{carry-off}(w, v)),$
$\langle w : a,\ y\colon \textsc{p}, v'\colon \textsc{p} \bullet \textsc{i},\ x\colon \textsc{p} \bullet \textsc{l}\rangle$

Had we used *master* instead of *read* in (6.41b), we would have quantified over informational objects only. The TCL formalism thus captures the subtleties of the quantificational puzzles.

6.3.3 Copredication Revisited

We now return to the copredication examples. Let us look at a classic example of copredication.

(6.56) John picked up and mastered three books.

We suppose that the phrasal verb *pick up* must take a physical object as its argument and that the verb *master* must take an informational object. The copredication itself enforces an identity of types between the two conjoined phrases since the conjoined verbs have common arguments. To look at some more details, let us suppose that the syntax of coordinated conjunction is not itself symmetric, but rather of the form [X [and/or X]]. So the coordinate construction must apply first to the second V and then to the first. As is well known, when transitive verbs are conjoined we have to ensure that they have common *arguments*. One standard way to do this is to suppose as did Montague a special coordination rule—for transitive verbs, it looks like this:

- Coordination Rule:
  $\lambda\mathcal{P}\lambda u\mathcal{P}(\lambda v\phi) \wedge (\vee)\lambda\mathcal{P}\lambda u'\mathcal{P}(\lambda v\psi) = \lambda\mathcal{P}\lambda u\mathcal{P}[\lambda x(\lambda v\phi[x] \wedge (\vee)\lambda v\psi[x])]$, provided type($u$) $\sqcap$ type(u') $\neq \bot$

Working up from the bottom, the type equation introduced by the coordinate construction must apply first to the second V and then to the first. We will assume a generalized conjunction operator that allows us to conjoin two λ-terms of any arity as long as they have the same arity. We now return to our example from above. Syntax gives us the following:

(6.57) $\lambda\mathcal{P}\lambda u\mathcal{P}[\lambda v\text{pick-up}(u,v)]\wedge\lambda\mathcal{P}\lambda u'\mathcal{P}[\lambda v'\text{master}(u',v')], \langle u, u'\colon \textsc{a}, x : e, \mathcal{P}\colon (e \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, v\colon \textsc{p}, v' : \textsc{i}\rangle$

Using the coordination rule we get:

(6.58) $\lambda\mathcal{P}\lambda u\mathcal{P}[\lambda x(\lambda v\text{pick-up}(u,v)[x]\wedge\lambda v'\text{master}(u,v')[x])], \langle u\colon \textsc{a}, x : e, \mathcal{P}\colon (e \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, v\colon \textsc{p}, v' : \textsc{i}\rangle$

If we try to apply the two verbs to the argument x, we'll get a type conflict. But we could delay those applications and apply this term to the object DP *three books*. After Type Accommodation we have:

(6.59) $\lambda\mathcal{P}\lambda u\mathcal{P}[\lambda x(\lambda v\text{pick-up}(u,v)[x] \wedge \lambda v\text{master}(u,v)[x])](\lambda P\exists_3 w(\text{book}(w) \wedge P[w])$,
$\langle u\colon \textsc{a}, x\colon \textsc{p}\bullet\textsc{i}, \mathcal{P}\colon ((\textsc{p}\bullet\textsc{i}) \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, v\colon \textsc{p}, P\colon (\textsc{p}\bullet\textsc{i}) \Rightarrow \textsc{t}, v' : \textsc{i}, w : \textsc{p}\bullet\textsc{i}\rangle$

It is here where things get a bit tricky. What is the head construction here? As I intimated earlier, both constituents should contribute to the type conflict resolution. And so from that perspective, the head is the coordinated VP and *not* the separate Vs. That means that x has the type to be conserved in resolving the type conflict in the predications $\lambda v\text{pick-up}(u, v)[x]$ and $\lambda v\text{master}(u, v)[x]$.

This means that we must use • Exploitation twice on x and so introduce two new variables of the types appropriate to combine with the expressions corresponding to *pick-up* and *master*, thus changing the typing context from the one above by extending the old typing context with $z\colon \textsc{p}, z'\colon \textsc{i}\rangle$—call this new context C. The Non Head Transfer Rule tells us that we have to now apply the appropriate functors to the two individual verbs, redoing the derivation from that point on. After incorporation of the functors, our coordinated entries look like this:

(6.60) $\lambda\mathcal{P}\lambda u\mathcal{P}[\lambda v\exists z(\text{ pick-up}(u, z) \wedge \text{ O-Elab}(z, v))], \langle\mathcal{P}\colon (\textsc{p} \bullet \textsc{i} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, v\colon \textsc{p} \bullet \textsc{i}, z\colon \textsc{p}\rangle$

(6.61) $\lambda\mathcal{P}\lambda u'\mathcal{P}[\lambda v'\exists z' \text{ master}(u', z')\wedge \text{ O-Elab}(z', v')],\ \langle\mathcal{P}\colon (\textsc{p}\bullet\textsc{i} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, v'\colon \textsc{p}\bullet \textsc{i},\ z'\colon \textsc{i}\rangle$

Redoing the conjoined VP, we get the following now, abbreviating the whole typing context now to C.

(6.62) $\lambda\mathcal{P}\lambda u\mathcal{P}\lambda x(\lambda v\exists z(\text{ pick-up}(u, z) \wedge \text{ O-elab}(z, v))[x] \wedge \exists z'(\text{master}(u, z') \wedge \text{O-Elab}(z', v')[x])),\ C$

Using Application and applying the DP meaning to the VP meaning, we get the desired result:

(6.63) $\lambda u\exists_3 w(\text{book}(w) \wedge \exists z\exists z_1(\text{ pick-up}(u, z) \wedge \text{ O-elab}(z, w)) \wedge (\text{ master}(u, z_1)) \wedge \text{ O-Elab}(z_1, x)),\ C$

6.3.4 The Complexity of TCL Rules

The TCL rules come in two kinds. One kind concerns operations on types; the other type embodies a ”reflection” of the type operation on the logical form itself. The operations on types are all sound relative to our background semantics. They also correspond to logical rules. The exploitation rules for • correspond to a guarded non destructive conjunction exploitation. They correspond to obviously valid inferences of intuitionist logic. Application corresponds to modus ponens. Application corresponds to conditional proof. In computing the lambda free term for the meaning representation of a clause, there is very limited use of introduction rules for complex types; we can even assume that abstraction is used ”off-line” from the normalization process, as all of the types of the terms (even the complex ones) are given in the lexicon or by Transfer. And so the type system has considerably less flexibility that the implicational fragment of intuitionistic logic. Type Accommodation is a restricted $\wedge$ introduction rule, for instance, that can only be invoked when we can and must unify the type of an argument and predicate to make application work. We cannot introduce such types unless predication demands it. And with • exploitation, we cannot introduce new variables unless predication demands it as well.

TCL deductions are linear in the sense that once types are combined using Application, they are not used further. Furthermore syntax completely determines the order of application and which terms apply to which. While λ operators introduced in the lexicon may capture some variables during the composition process, apart from that there are no new bindings in the conversion process. Another point is that while while adjectives, determiners, reflexives may introduce some non linear terms into a deduction as can constructions like coordination, secondary predication or relative clauses, these non linear terms do do not increase the number of applications or the complexity of the derivation; the higher order variables are linear and the complexity of the terms never *increases* when doing application with these terms, and the duplicated terms are all specified lexically to be of no higher type than e. Finally, given a syntactic tree for a clause with n words,[10] there are at most n-1 predications, one for each node of the complex tree. It is quite clear then that in virtue of the results of Aehling and Schwichtenberg (2000), TCL deductions that use only the basic rules deduce a complete normalization in polynomial time.

Nevertheless, TCL allows resolutions of type clashes by extending the type context with a fresh variable by using • exploitation. This certainly makes things more complex. Together with • exploitation comes the use of one of two the transfer rules, which involve backtracking and redoing lambda computation. In effect we have to redo the TCL deduction. The transfer rules also add complexity to the lambda terms by adding new variables. However, there is an upper bound on the number of new variables added—three in the worst case for a transfer by a particular occasion of • exploitation. This adds then at most 4 extra Application steps to the new deduction.

There is also the possibility of exploiting all the various constituent types of a • type. TCL assumes that these • types are lexically given, however, and so we can list in advance what these and their lexically realized constituent types

[10] If we assume that morphemes of words or even phonologically empty nodes can occur in the tree and that these have semantic content, then one will have to count these as well.

are. There are of course all the simple constituent types, but not every possible • combination of simple constituent types of a • type are lexically realized. For instance, while we have NEWSPAPER $\sqsubseteq$ P • I • INSTITUTION • PUBLISHER, we do not have I • INSTITUTION as a lexically realized type. For each • type that occurs in a predication k, let m_k be the number of lexically realized types relevant to k. Given that each single deduction without resetting occurs given the results of Aehling and Schwichtenberg (2000) within polynomial time and given that there are at most $\Pi_{1 \leq k \leq n-1} k(m_k)$ resettings adding at each resetting at most three extra variables, the entire TCL normalization process remains within a time that is polynomial within the number of words in the clause and the number of lexically realized subtypes of all the • types given by words in the clause. Thus,

Fact 3 *Whether a clause ϕ has a complete normalization or not, given a syntactic tree for ϕ and basic typings for λ terms for the words in ϕ, is computable in TCL within a time that is a polynomial function of the number of words ϕ and the number of lexically realized subtypes of the complex types in ϕ.*

6.4 Introduction Rules for • Types

As illustrated earlier, a verb like *read* not only selects for a dot object (5.15a); it also coerces a lower type to dot object status (5.15b-c).

(5.15) a. Mary read the book.

b. John read the rumor about his ex-wife.

c. Mary read the subway wall.

d. Mary read the stone.

e. Mary illuminated and then read the subway wall.

In these examples, the use of the verb *read* entails that the rumor about the ex-wife must be printed or written down in some physical form and that the subway wall has some information content inscribed upon it. The reason for this is a lexical presupposition of the verb; *read* involves necessarily both an event of scanning or being in contact with some sort of physical object So these objects, one of informational type the other of physical type, would seem to take on a second aspect; these objects become both informational and physical objects—or objects of • type. *Read* also enters into copredications that exploit one of the constituent types of the • as seen from (5.15d)/

One should note that this phenomenon is peculiar to particular words. • type introducers are rare. Besides *read*, there are closely related words like *decode, decipher, peruse, make out, pore over, scan, study, translate* that arguably induce complex types for their internal arguments.

For some quite closely related verbs like *master* the possibility of coercion to a complex type is degraded. For instance, (6.64b) is a lot less felicitous than its (5.15) counterpart:

(6.64) a. Mary mastered the book.

b. ??Mary mastered the subway wall.

Furthermore, coordinate predication or copredication, which often requires a • type as an argument, doesn't license the introduction of such a type:

(6.65) John built and mastered the wall.

More striking still is the way the verb *Write* fails to pattern with *read*. Although what writing produces is an object of complex type P • I, it cannot easily convert to • type internal arguments that are lexically typed of physical type:[11]

(6.66) a. Mary wrote the book.

b. John wrote the rumor about his ex-wife.

c. #John wrote the subway wall.

d. #John wrote the stone, the typewriter paper.

This asymmetry is quite remarkable, and the explanation for why this transformation is possible with arguments of *read* but not with *write* is quite involved. Recall that *write* produces an object of *bullet* type; but as it is a verb of creation, the object does not exist (physically) beforehand and so cannot be typed P prior to the action of verb. Thus, the P type of the direct object of *write* must apply to the object after its creation. But at this point we have an unrecoverable type clash with direct objects of simple type P. It appears far easier to assume that something of type I has a physical instantiation. This is because of the nature of writing. Writing is a process that begins with an informational object (perhaps not completely realized either) and then issues in the creation of a physical instantiation of the informational object. Simply physical types have incompatible realization

[11] You can also write a rumor, however. A quick check on Google reveals several attested instances.

conditions with those of *write*. On the other hand, with *read*, we make a presupposition as to the type of its object —it must be of type P • I. This presupposition can be accommodated in many different ways, as many sorts of objects can have information stored on them (people write on stones and on subway walls) so that the process of reading can take place. Thus, the validity of this coercion depends on the possibility of accommodating lexical presuppositions.

Very few words have such presuppositions. Verbs like *read* seem to be largely lacking for the other • types that instantiated in the lexicon: there are no EVENT • I or PRIZE • AMOUNT OF MONEY, PORTION OF MATTER • ARTIFACT introducers as far as I am aware. One must be careful to distinguish lexical • type introducers from coercive predicates, which are in great abundance in most languages. There are a few other grammatical constructions that introduce • types. In chapter 8, I'll argue that resultative constructions like (6.67) below introduce complex types involving both an object and an aspects of it that results after the completion of some process.

(6.67) Sam wiped the table clean.

I'll also claim that depictives like (6.68) introduce complex types of a particular sort

(6.68) Sam drove home drunk.

In any case, it appears that we should hypothesize that lexically induced • introductions depend on very particular lexical properties—viz., lexical presuppositions. How does this affect the way the rule operates at logical form? One telling mark has to do with quantification. We've seen that objects that are of • type have complex individuation conditions that can be checked in examples like those given with the quantificational puzzle. Does this mean that when a lexical item like *read* introduces a • type on an object that we can individuate that object relative to both informational and physical individuation criteria? In other words can such predications give rise to versions of the quantificational puzzle?

The answer is no, which is somewhat surprising, given that terms that are lexically typed as dual aspect nouns permit us to use criteria of individuation appropriate to each constituent type of the •. Consider the following examples.

(6.69) a. Mary read every screen.

b. Mary read three books.

c. Mary read three rumors about her ex husband.

It seems she has to read all the physical screens, not only all the informational ones, in order for (6.69a) to be true. That is, in a situation where many screens have the same information on them, Mary must have read each and every physical screen, not just the few screens that jointly convey all the information on the screens in the situation, in order for (6.69a) to be true. Similarly Mary must read three informationally distinct rumors about her ex in order for (6.69b) to be true. In other words (6.69b) cannot have the meaning *Mary read a (the) rumor about her ex three times*. On the other hand, for Mary to have read three books, it is preferred that they be individuated informationally but that is not necessary, as we have seen in other cases.

Furthermore, the individuation of the objects carries over to an individuation of the events of reading. Reading three books comprises three events, one for each reading of each book. But these events may be individuated either in terms of three physical books or three informational books (without regard to how many physical books were involved in the reading events). This is not true for (6.69a,c). The events are individuated relative to the set individuation conditions of the ”unshifted” type of the arguments. This means that physical screens and the informational rumors have to be involved in the reading events themselves somehow in order to individuate the events in the appropriate way.

Thus • introductions, when mandated by a lexical verb, do not change the counting principles or the individuation criteria for the objects whose types are shifted. The story is the same, as we shall see, for the grammatical constructions of relative predication, depictives and resultatives, which also license the introduction of • types and which we will study in much more detail in the next chapter. As we'll see, relative predications also don't change counting principles for the objects shifted.

There is a question as to whether such • introductions amount to coercions. We haven't looked at coercions in TCL yet, but in principle we can make a distinction between actually changing the type of the argument and adding another, appropriately related variable to logical form. The use of a lexical predicate really does in some sense tell us more about the type of its argument; if one reads a stone, then the stone must have an informational aspect. The stone is no longer a mere stone, but a conveyor of information. Something like the Rosetta Stone, a very famous stone conveying information, can be destroyed simply by wiping out the inscriptions on it, whereas a normal stone can remain the same stone after considerable erosion.

Contrast the case of the Rosetta stone with a classic example of coercion. To say that Mary enjoyed her glass of wine doesn't in any way tell us more about the intrinsic nature of the glass of wine, only that Mary enjoyed doing something with it, presumably drinking it. The latter is a classic case of coercion and one can see a subtle if unmistakable distinction between the two cases. There are many more cases of coercion than • introduction. In Asher (2007) I argued that the preposition *at* might introduce a complex P • L (L is recall short for the type LOCATION) in constructions like:

(6.70) Mary is at the chair.

(6.70) is possible in contexts where Mary is playing some sort of game where pieces of furniture are used as waypoints or locations. I now think that *at* is probably not a • type introducer but functions rather more like the coercions with the aspectual verbs or *enjoy*. In this situation the chair remains a chair, while also serving as a location. Here the intrinsic nature of the chair is not altered.

The introduction of • types behaves quite differently from the exploitation of lexically given • types. The introduction rules involve a local adjustment of types for the purposes of the predication, but not percolating up to the full DP and thus affecting the quantificational domain, in the way we have seen • exploitation does with the quantificational puzzles. In addition, the introduction of • type is licensed only by particular lexical items or grammatical constructions. We need two versions in fact, particular lexical rules for predicates like *read*), and a structural rule. I give the lexical rule here.

(6.71) Lexical **•-Introduction** (•I) for a lexical term *t* that must combine with t':

$$\frac{\Delta, C \vdash t[t']\colon \bot, C \vdash y\colon \alpha(\beta),\ \ \Delta \vdash y \text{ in } t'}{\Delta, C + \langle z\colon \alpha \bullet \beta\rangle \vdash t[t'],}$$

Our transfer rules for • will transfer the information implicit in the type shift of these rules into the logical form. We will use the same operators as before. But to which term should they apply? They should not apply to the terms denoting the objects of simple type, because otherwise we would predict instances of the quantificational puzzles, where there are none.

There are two ways in which we can implement the theory so as to derive the observations. The first possibility is that the transfer functions of a now famliar sort must apply in the case of lexical • introduction to the smallest term responsible for the typing on the variable of complex type. This leads to a violation of the Head Typing Principle, but it is in keeping with our approach of changing types at the point at which they are introduced into the derivation. On the other hand, we could postulate a more local type shift but at a higher level—that is, the entire DP logical form is shifted with the following sort of functor:

(6.72) $\lambda\mathcal{P}\lambda Q\mathcal{P}(\lambda u\exists \mathrm{z}(\text{O-elab}(u,z)\wedge Q(\mathrm{z})))$,

where $\mathcal{P}$ has the type of a DP or generalized quantifier, z has the complex type as dictated by the lexical • introduction rule and Q has the obvious property type. This second approach also implements the idea that this is a type shift that is done locally without large scale effects like for example a shifting of the quantificational domain.

To give a sample derivation using the first approach, *read the subway wall* looks like this:

(6.73)

a. By the usual methods we have:
$\lambda\mathcal{P}\lambda u(\mathcal{P}[\lambda v\mathit{read}(u,v)])[\lambda P\exists x(\text{subway-wall}(x)\wedge P(x))], \langle u\colon \textsc{a}, v : \textsc{p}\bullet \textsc{i}, x : \textsc{p}, P : \textsc{p}\Rightarrow\textsc{t}, \mathcal{P} : (\textsc{p}\bullet\textsc{i}\Rightarrow\textsc{t})\Rightarrow\textsc{t}\rangle$

b. By the lexical rule for • intro, we get:
$\lambda\mathcal{P}\lambda u(\mathcal{P}[\lambda v\mathit{read}(u,v)])[\lambda P\exists x(\text{subway-wall}(x)\wedge P(x))], \langle u\colon \textsc{a}, v : \textsc{p}\bullet \textsc{i}, x : \textsc{p}\bullet\textsc{i}, y : \textsc{p}, P : \textsc{p}\Rightarrow\textsc{t}, \mathcal{P} : (\textsc{p}\bullet\textsc{i}\Rightarrow\textsc{t})\Rightarrow\textsc{t}\rangle$

c. By Transfer applied to *read* and Application:
$\lambda\mathcal{P}\lambda u(\mathcal{P}[\lambda v\exists z\mathit{read}(u,z)\wedge\text{O-elab}(v,z)])[\lambda P\exists x(\text{subway-wall}(x)\wedge P(x))], \langle u\colon \textsc{a}, \mathrm{z} : \textsc{p}\bullet\textsc{i}, x, v : \textsc{p}P : \textsc{p}\Rightarrow\textsc{t}, \mathcal{P} : (\textsc{p}\Rightarrow\textsc{t})\Rightarrow\textsc{t}\rangle$

d. By Application and setting the whole type context to C:
$\lambda u\lambda P\exists x(\text{subway-wall}(x)\wedge P(x))[\lambda v\exists z(\mathit{read}(u,z)\wedge\text{O-Elab}(v,z))],\ C$

e. And by Application again twice:
$\lambda u\exists x(\text{subway-wall}(x)\wedge\exists \mathrm{z}(\text{O-Elab}(x,z)\wedge\text{read}(u,z)),\ C$

Notice that in this derivation, there is a local adjustment on the verb's type. Thus, quantification in the DP remains over whatever it was lexically specified to be by the head of the common noun phrase. This is typical of coercion effects that we will study in the chapter after next. Coercion effects are quite different from the type adjustment in copredication or • exploitation, because they do not really change the fundamental character of the object whose type is changed. They don't change quantificational domains. But that makes sense: saying that a physical object has some information contained in it does not change that object's identity conditions. We get this conclusion because of the local type shift; our formula says that there is an object of complex type of which the subway wall is the physical aspect, but we do not change the type or individuation conditions of subway walls in so doing. This formalisation also predicts that for something like (6.69a) there will be three events, one for each physical screen. However, we do do violate the Head Typing Principle.

The alternative proposal keeps the same derivation until the Transfer step, which in this case gives us for the DP meaning

(6.74) a. $\lambda\mathcal{P}\lambda Q\mathcal{P}(\lambda u\exists \mathrm{z}(\text{O-elab}(u,z)\wedge Q(\mathrm{z})))[\lambda P\exists x(\text{subway-wall}(x)\wedge P(x))], \langle\mathcal{P} : (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow \textsc{t},\ Q\colon (\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t},\ P : \textsc{p} \Rightarrow \textsc{t}, u, x : \textsc{p},\ \mathrm{z}\colon \textsc{p} \bullet \textsc{i}\rangle$

b. after several uses of Application:
$\lambda Q(\exists x(\text{subway-wall}(x) \wedge \exists \mathrm{z}(\text{O-elab}(x,z) \wedge Q(\mathrm{z})))), \langle Q\colon (\textsc{p} \bullet \textsc{i}) \Rightarrow \textsc{t}, x : \textsc{p},\ \mathrm{z}\colon \textsc{p} \bullet \textsc{i}\rangle$

This can now combine with the lambda term for *read* giving equivalent results but without violating the Head Principle.

This completes my introduction to • types and copredication. In the next chapter I consider a type of predication, restricted predication that makes essential use of aspects. It is a way of introducing aspects grammatically and then attributing properties of them.

6.5 • Types and Accidentally Polysemous Terms

In introducing TCL, I noted that one might treat accidentally polysemous terms like *bank* using another complex type restrictor akin to disjunction. Let us take a look at that proposal a bit more closely.

(6.75) Disjunctive Types: If α, β are types, then $\alpha \vee \beta$ is a type.

Extending our notion of extended greatest lower bound to disjunctive types, we have:

- $(\alpha \vee \beta) \sqcap^* (\delta \vee \gamma) = (\alpha \sqcap^* \delta) \vee (\beta \sqcap^* \gamma) \vee (\alpha \sqcap^* \gamma) \vee (\beta \vee \delta)$

This definition permits $\vee$ to be a symmetric type constructor. That is, there is no difference between the type $\alpha \vee \beta$ and the type $\beta \vee \alpha$.

This leads to some straightforward observations about subtypes with disjunctive types:

Lemma 5 *Subtypes with disjunctive types:*

a $\alpha \sqsubseteq \alpha \vee \beta$

b $\beta \sqsubseteq \alpha \vee \beta$

The idea is that a word like *bank* will have some sort of a disjunctive type. More precisely the λ bound variable x in λxbank(x) would have the type constraint x: FINANCIAL INSTITUTION $\vee$ P • L, assuming that river banks are physical objects as well as locations. A predication like

(6.76) The bank specializes in IPO's.

would involve an operation of disjunction elimination among types, which is simply a feature of Type Accommodation in TCL in virtue of the observations about $\sqsubseteq$ and disjunctive types. That is, the predicate *specializes in IPO's* types its argument as FINANCIAL INSTITUTION, and this can combine with the accidentally polysemous subject. Thus, Type Accommodation can be understood as a type disambiguator.

Accidentally polysemous expressions are vastly different in their logical behavior from their logically polysemous cousins. Type accommodation does not add any variables to logical form or have anything like the complexity of the • exploitation rules. Thus, we predict that accidentally polysemous terms will share little in the behavior of their logically polysemous cousins: we should not get the same patterns of anaphora resolution or of copredication. Consider the situation with anaphora first and compare the anaphoric possibilities with accidentally polysemous expressions with those that are logically polysemous

(3.41) John's Mom burned the book on magic before he could master it.

(6.77) The bank specializes in IPO's but it has eroded considerably due to the recent floods.

Once their type has been specified from disjunctive type $\alpha \vee \beta$ to the type of one of the disjuncts, say α because of predication, accidentally polysemous expressions do not support anaphoric reference to objects of type β. This is predicted if Type Accommodation is used to make the specification. In the logical form for (6.77) there is no bound variable of the appropriate type from the first clause to serve as the binder for the pronoun *it*. This contrasts with TCL's treatment of (3.41) where • exploitation furnishes just such a binder for the pronoun.

As to copredication, recall that it is not the complex predicate that confers a • type on the argument but rather it is only the fact that the logically polysemous argument is of • type that we can successfully combine it with the predicate constituents of the complex. Let us now see what happens to an attempted derivation for a similar copredication with an accidentally polysemous term.

(6.78) The bank has eroded because of the recent high water in the river and specializes in IPOs.

(6.79) a. Syntax and the coordination rule yield as before:
$\lambda x(\lambda v$ eroded ...$(v)[x] \wedge \lambda v'$ specializes ...$(v')[x])]$,
$\langle e\colon e, v\colon \textsc{p} \bullet \textsc{l}, v'\colon \textsc{financial-institution}\rangle$

b. If we try to apply the two predicates to the argument x, we'll get a type conflict. But we could try to apply this term to the subject DP *the bank*. After Merging and Type Accommodation we have:
$\lambda u\mathcal{P}[\lambda x(\lambda v\text{pick-up}(u, v)[x] \wedge \lambda v\text{master}(u, v)[x])](\lambda P\exists!w(\text{bank}(w) \wedge P[w])$,
$\langle x\colon e, v\colon \textsc{p}\bullet\textsc{l}, v'\colon \textsc{financial-institution}, w : ((\textsc{p}\bullet\textsc{l})\vee\textsc{financial-institution}), x\colon ((\textsc{p}\bullet \textsc{l}) \vee \textsc{financial-institution}))\rangle$

Now, however, we must apply Type Accommodation to x to resolve the type conflict between x and v and v'. Whichever we do, we will now specify x either to be of type $\textsc{financial-institution}$ or of type $\textsc{p} \bullet \textsc{l}$. And if we do one, we cannot use Accommodation to get the other type adjustment, since $(\textsc{p} \bullet \textsc{l}) \sqsubseteq \textsc{financial institution} = \bot$. Our assumption that these two types have no greatest lower bound makes sense in view that objects that are financial institutions have different individuation and identity conditions from things that are physical objects and locations. Thus, the attempted derivation crashes and no logical form can be constructed for sentences like (6.78.

Chapter 7

Restricted Predication

In this chapter I investigate how grammatical constructions like relative predication—the *qua* construction can introduce complex • types. Briefly the idea is that something of the form DP *as a* DP introduces a variable from the head DP whose complex • type is determined from the head DP and the DP in the comp of the adverbial clause. Thus, *John as a banker* introduces a variable for predication of the type HUMAN • BANKER. But first let's get some background.

Relative predication is sometimes invisible on the surface. Consider the apparently extensional constructions in (7.1), noticed by Jenny Saul (1997a,b, 1999) . Attitude contexts pose well-known problems for truth conditional semantics, in particular rendering problematic inferences involving the substitution of coreferential proper names. But the sentences in (7.1 exhibit a similar behavior.

(7.1) a. Superman always gets more dates than Clark Kent does.

b. ?? Superman always gets more dates than Superman does.

(7.2) a. Chris hit Clark Kent, but he never hit Superman.

b. ?? Chris hit Clark Kent, but he never hit Clark Kent.

The (a) versions of the above examples all seem satisfiable, perhaps even true, and to convey nontrivial information. They differ intuitively from their substitutional variants in (b), which seem necessarily false. These intuitions about the truth conditional status of (7.1a-7.2a)) versus (7.1b-7.2b) imply a semantic rather than pragmatic difference between the (a) and (b) variants; that is, there is intuitively a difference in the truth conditional status of the (a) and (b) variants, which should follow from the semantic interpretation of their logical forms.[1]

A Fregean, semantic approach according to which (7.1a-7.2a) are true and informative but their counterparts (7.1b-7.2b) are not appears promising. But it is challenging to apply the standard Fregean machinery to these simple sentences. On a Fregean semantic analysis, it is the presence of an attitude or modal operator that forces expressions to take their sense as their semantic value rather than their ordinary, ”extensional” semantic value (their reference). But what operator or construction triggers the Fregean machinery of non customary semantic values in (7.1-7.2)? We don’t want to stipulate that all apparently extensional constructions are in fact intensional!

To answer that question, consider the following, natural paraphrases of the (7.1a-7.2a) using *as* phrases:

(7.3) a. Superman as Superman always gets more dates than Superman as Clark Kent does.

b. Chris hit Superman as Clark Kent, but he never hit Superman as Superman.

The naturalness of the paraphrase strongly suggests that Saul’s examples involve some mechanism equivalent to *as* phrases. But what do *as* phrases mean? I’ll spend much of this essay answering that question. I’ll be concerned with sentences of the form

(7.4) ϕ*as* $\psi\chi$

where χ is some sort of predication on an argument introduced by ϕ. I’ll call the predication in χ the *main predication* and the predication derived from ψ the *restricting predication*.[2] I’ll return to the examples (7.1-7.2) in the concluding section of the paper.

A first thought is that *as* phrases function like attitude contexts to force the constituents within their scope to take on something like Fregean senses as semantic values. But attitude contexts and *as* phrases function quite differently with respect to inferences involving identity, suggesting that we cannot have a single account for both phenomena. Proper names in *as* phrases like those in (7.3)

[1] There may be in addition differences in implicatures or felicity and infelicity conditions that would naturally demand a pragmatic explanation, but I won’t consider these here.

[2] There’s considerable evidence that *as* constructions are clauses and involve predications, as Jaeger (2003) and Szabo (2003) argue. I take their view of the syntactic facts to be basically right, though I’ll add a few details in section 3 below.

have a different intensional behavior from names in other intensional contexts. (7.3a-d) remain ”informative” and even true in spite of the fact that the identity between CK and S is known or part of the background context, while the ”uninformative” counterparts are *false*. Thus, the following propositional attitudes seem completely consistent:

(7.5) a. John believes that Superman = Clark Kent

b. John believes that Superman as Superman always gets more dates than Superman as Clark Kent does.

The lack of substitutivity within *as* phrases is striking when we compare it to the epistemic transparency of other predications inside attitude contexts. Suppose that (7.6a-b) are both true.

(7.6) a. John believes that Hesperus is Phosphorus

b. John believes that Hesperus is dim and that Hesperus is a planet.

Then it seems to be a valid inference that

(7.6c) John believes that Phosphorus is dim and that Phosphorus is a planet.

If one quibbles with the inference from (7.6a-b) to (7.6c), it is enough for my purposes that one accept that John ought to believe that Phosphorus is dim and that Phosphorus is a planet if (7.6a-b) are true. The reason this argument is valid is that thoughts should support the customary logical inferences; so if you believe that $a = b$ and you believe $\phi(a)$ you believe, or at least ought to believe, that $\phi(b)$. But now what about *as* phrases? Suppose that in the spirit of traditional Fregean approaches to meaning, we adopt (1) a compositional theory of senses, according to which the sense of each meaningful, syntactic constituent of a sentence S becomes itself a constituent of the meaning of the proposition or thought that S expresses; so the meaning of a is F is the proposition in which the sense of F is applied to the sense of a. Suppose further that (2) we analyze propositional attitudes as a relation between a proposition and an agent, and finally that (3) the names ’Clark Kent’ and ’Superman’ are syntactic constituents of the *as* phrases in (7.5b) (something which seems undeniable). Then we land into the following difficulty. By (1) the embedded proposition in (7.5a) can be represented as the application of a complex concept to the senses of the proper names in the complements of the *as* phrases. Now since thoughts support the customary logical inferences, from (7.5a) and (7.5b) it follows by the laws of identity that

(7.7) John believes that Superman as Superman always gets more dates than Superman as Superman does.

So from a pair of intuitively consistent beliefs and some minimal assumptions, we infer that John should believe something necessarily false, as long as thoughts obey the ”laws of thought”, and this is clearly counter to intuitions.[3]

7.1 Landman's Puzzle

Landman (1989) provided a set of intuitive postulates which show that *as* phrases are closed under implication and other logical inferences and so have much in common with extensional constructions. They are also veridical constructions; a sentence like *John as a judge is corrupt* implies that John is a judge. These postulates suggest an extensional analysis. However, Landman's postulates, which I present informally, also reveal a deep problem with our intuitions about restricted predication.

Landman's Postulates:

1. John as a judge is John.
2. If John as a judge is corrupt and John as a judge is well-paid, then John as a judge is corrupt and well paid.
3. If taking bribes implies being corrupt then if John as a judge takes bribes then John as a judge is corrupt.
4. John as a judge is not both corrupt and not corrupt.
5. If John as a judge is corrupt, then John is a judge.
6. John as a judge either takes or doesn't take bribes.

These axioms are plausible but lead to the following problem:

[3] We could try resorting to a hierarchy of senses view: the complement of the *as* phrase in (7.5) would then be assigned a second level sense. Then, even while the first level sense of *Superman* might be identical to the first level sense of *Clark Kent*, they could differ in second level senses. But now suppose that John believes that Superman is Clark Kent (7.5a), and that he also believes that he believes that Superman is Clark Kent. We can run the above anti-Fregean argument above again.

- From (5): If John as a judge is John, John is a judge.
- From (1) John as a judge is John.
- So from (1, 5), John is a judge
- But similarly from (5): If John as a non-judge is John, John is a non-judge
- So by parallel reasoning from (1) John is a non-judge–i.e., John is not a judge.

Putting both arguments together then, we start from what appear to be uncontroversial premises to arrive at a contradiction.

7.2 More Puzzles

Adding to the puzzles about *as* phrases is the way the copula and proper names function within these restricting predications.[4] In standard predications, when we use proper names in predicative constructions and a definite NP in subject position, as in

(7.8) a. That man is Mark

b. The prettiest city in the world is Paris.

the predication is naturally construed as one of identity. However, *as* predications appear to be different. Consider, first,

(7.9) a. John as Sam was interesting.

b. John as Sam earns more than $50K.

These examples sound strange; whatever they are, the predications *as PN*, where *PN* is a proper name, do not function like their counterparts with the copula. Names with "descriptive content" or that designate roles like *Lear, Macbeth, Hamlet* or *Superman* and *Clark Kent* in complements of *as* phrases help us see what's going on.

(7.10) John as Lear was fantastic, but John as Hamlet was boring.

[4] Jaeger (2003) notices that *as* phrases are not synonymous with *is* predications but he does not discuss the case of proper names.

If the phrases *as Lear* and *as Hamlet* in (7.10) functioned the way the predications do in (7.8), then logic would dictate that Hamlet = Lear. But clearly this does not follow from (7.10), and so it would appear that proper names do not have a normal predicative role when they are complements of *as* phrases. Similar intuitions about the use of proper names obtain for the examples (7.3). In so far as (7.9a) sound OK, it's because we're saying that John is playing at being Sam—he dresses like him, talks like him, acts like him. This is of a piece with other predicative uses of names.

(7.11) He's an Einstein.

(7.12) You're no Jack Kennedy.

As phrases with proper names function similarly. They predicate properties associated with the bearer of the name to the head of the construction.

This discussion reveals yet another reason to reject a simple Fregean approach to *as* constructions. A Fregean approach to intensional contexts changes only the *reference* of the expressions within the context and *not* the nature of the predication. But we see that *as* clauses require precisely another account of predication. We must supplement the basic Fregean idea with some view of how *as* would link the sense of its complement with its subject.

To solve these puzzles, we have to do more than simply analyze the construction itself. We must rethink how predication works within and outside of *as* phrases, since *as* phrases affect how we understand the main predication. To say that John as a judge is corrupt is to say, roughly, that John is corrupt *insofar as* or *when* he is a judge. This temporal gloss invites an extensional semantics for *as* phrases. Jaeger (2003) and Szabo (2003) construct a semantics for *as* phrases that makes crucial use of situations or eventualities, both of which are temporally extended entities. Roughly both accounts take *as* phrases to restrict the main predication to some temporal span of the subject. On these views, predication depends on a hidden situation or eventuality argument. After reviewing and criticizing their accounts, I'll then propose my own account of *as* phrases, which also involves a reconsideration of how predication works.

7.3 Extensional Semantics for *As* Phrases

Jaeger (2003) stipulates that *as* phrases hold in a ”small” situation or part of the world of evaluation. He also observes that *John as a judge is corrupt presupposes* that John is a judge.[5] Jaeger rejects the formulation of the Landman axioms, because their formulation should filter out the presusuppositions, Jaeger's reformulation of all the axioms avoids the problem noted by Landman. To ensure the validity of his reformulated axioms, however, Jaeger has to make some further stipulations . For instance, to capture axiom 5, the predications in the *as* phrases are upwards persistent, in the sense that if John is a judge in a ”small” situation, John is also a judge in a larger situation.

To account for the restriction of the main predication by the *as* phrase Jaeger uses Van der Sandt's (1992) theory of presupposition binding, according to which the presuppositional content of *as* phrases will move to get ”bound” at an appropriate site in the logical form of the sentence. Jaeger appeals to a hidden argument in the main predication to which the presupposition will bind; for example, *John as a judge is corrupt*, becomes after presupposition binding: *John is corrupt when he is a judge*. As Szabo (2003) notes, however, this would imply that this hidden argument can be explicitly bound by another adverbial as in (7.13a). In that case we would predict that the presupposition given by the *as* phrase is simply accommodated and plays no role in the main predication. But this prediction turns out to be wrong. Consider (7.13b), which is due to Szabo (2003), and which has an explicit restriction on the putative, hidden argument of the main predication as well as an *as* phrase. It only has the interpretation that John as a judge makes money that would be considered good for a janitor. Thus, contrary to Jaeger's account, the *as* phrase plays a restricting role even in the presence of the adverbial *for a janitor*.

(7.13) a. John makes good money for a janitor.

b. John as a judge makes good money for a janitor.

Szabo bases his account of *as* phrases on a NeoDavidsonian analysis of predication according to which predications are relativized to ”states” or other eventualities and then adds a partial ordering $\sqsubseteq$ over states. On his analysis, *as* phrases are sentential (IP) adjuncts with the following truth conditions:

- *John as a judge is ϕ* is true just in case there are states s_0 and $s \sqsubseteq s_0$ such that

  1. John is a judge in s_0;

[5] The standard tests for presupposition in questions and with negation seem to suggest that Jaeger is right: (i) *It's not true that John as a judge is corrupt* still implies that John is a judge; (ii) any answer to the question, *Is John as a judge corrupt?*, implies that John is a judge.

2. ϕ holds of John in s.
3. ϕ holds of John in all states s_1 such that $s \sqsubseteq s_1$ or s_1 is a state in some contextually salient alternative to John's being a judge (such as being a janitor).

According to this semantics, *John as a judge makes $50K* says that only in states of being a judge does John make $50K. It also implies that John is a judge at the actual world or maximal state of John's and it avoids Landman's problem by relativizing the predication of John's being a judge and his being a non-judge to two distinct states. Szabo also avoids positing a hidden argument for the main predication and as such seems more satisfactory than Jaeger's in that respect. To account for the presuppositions noticed by Jaeger, we can simply take clause 1 as a presupposition. Another feature of Szabo's account is that it contains a sort of focus constraint, to capture the intuition that the *as* phrase's predication of a property P to its subject implies that the subject has other properties than P. In the jargon of the semantics of focus, *John as a judge* implies other alternatives, John's being P, to John's being a judge. This effectively captures the intuition that *as* phrases restrict the main predication to some particular, restricted aspect of the subject.

7.4 A New Puzzle

So far so good. Nevertheless, not all is as well as it seems. For one thing, it's not clear how to get the truth conditions that Szabo wants compositionally from the meanings of the sentence's constituent expressions.[6] The real difficulty with the extensional accounts comes the underlying Neo-Davidsonian view of predication, according to which all predications are relativized to some sort of eventuality. In Neo-Davidsonian semantics, if I say that I am happy now to be going home but unhappy now that I have such a long flight ahead of me, then you can infer:

[6] *As* phrases appear to be sentential or IP adjuncts. While syntactic techniques allow some variable binding across an IP and IP adjuncts (by for instance raising a noun phrase to some higher position like the specifier in CP), Szabo's semantics requires that the *as phrase* constrain the eventuality (state or event) argument of the main verb. To do this the *as* phrase would have to take scope over the asserted predication and somehow constrain the eventuality of the main predication even though in standard compositional theories of the syntax-semantics interface, the eventuality variable introduced in the main VP is existentially bound by the time we interpret IP adjuncts of the main clause.

(7.14) a. I am unhappy now.

b. I am happy now.

Can one be simultaneously in a happy and an unhappy state? Perhaps because we interpret these as two relative predications, but in general this strategy is unavailable and the contradiction inescapable for many pairs:

(7.15) a. John makes $100K a year.

b. John makes less than $70K a year.

In (7.15) we cannot recover from this pair a non contradictory meaning. The intuition is that one cannot be in two states with contradictory properties at the same time. In fact, one might go so far as to suppose:

- State Consistency Principle:
  $(\exists s \phi(s) \wedge \exists s' \phi(s') \wedge s O_t s') \rightarrow \exists s''(\phi \wedge \psi)(s'')$

Though intuitively correct, the State Consistency Principle is not part of standard Neo-Davidsonian semantics. And it cannot be added, as we will see, to Szabo's approach.

Instead, Szabo modifies the standard Neo-Davidsonian view of predication by requiring that such predications hold not only of *some* state at the present moment but in *all* states that include that one, up to and including my maximal present state, unless they are states that are alternatives of those singled out by the predication. This last bit isn't very clear, since we don't have any clear critierion as to when a state is an alternative to another or not. Let us call this principle the *Persistence* of predications.[7]

But now suppose that John works two jobs and thus that (7.16) is true:

(7.16) a. John is a judge

b. John is also a janitor.

According to Persistence, John will then be both a judge and a janitor in his maximal present state. So far so good: this seems to be true of people who hold two jobs at once.[8]

[7] Note that Jaeger also adopts something like Persistence.

[8] Although being a judge and being a janitor may be alternatives to each other for Szabo, they can both clearly apply to John in his maximal present state. Thus, I conclude that Szabo's constraint really does not affect persistence or else it gets the wrong results.

The other piece of the puzzle has to do with whether the main predications apply to all states that satisfy the restricted predication. Szabo's last constraint implies that this is so. The same intuition that underlies Szabo's fix of the standard Neo-Davidsonian view of predication leads us to accept the view that the main predication should hold in *all* those states in which the restricted predication holds. I call this constraint *universality*:

- Universality:
  If $\phi as\ \psi\ \chi$ holds, then in any state s in which ϕ has the property expressed by ψ, then the property expressed by χ applies to ϕ in s as well. That is, if *John as a banker is corrupt* is true then in all states in which John is a banker, he is corrupt.

Further support for Universality comes from the fact if it fails to obtain, then the truth conditions of (7.17ab) are jointly satisfiable in Szabo's account. (7.17a) would be true just in case there is some state s in which John is a judge and some substate s' of s in which John is honest. This semantic analysis of (7.17a) is compatible with John's being corrupt in some, many, even most situations when he is a judge.

(7.17) a. John as a judge is honest

b. John as a judge is corrupt.

Intuitions belie this prediction. If it's true that John as a judge is honest, then it's not true that in there are situations where John is a judge and he is corrupt.

Universality and Persistence together with Szabo's account yield a contradiction. It's not hard to see why. If *as* constructions allow us to predicate contradictory properties of some object under different restrictions, and the restricted predications are upwards persistent and universality holds, then we will predicate contradictory properties of the object in the same state. To make the problem concrete, suppose further that the following intuitively satisfiable claims are true.

(7.18) a. As a judge, John makes $50K.

b. But as a janitor, John only makes $20K.

Assuming (7.16a,b), Persistence implies that John is a judge and a janitor in the maximal state of John (at the present time). According to Universality, the main predications in (7.18a,b) should hold of every state that satisfies the predication in the *as* clauses, But then, the maximal present state s of John must be such that John makes \$50K in *s* and such that John makes only \$20K in *s*, which is impossible. Our argument has shown that there is no restriction on the main clause predication by the *as* clause: *John as a* ϕ ψ*'s* is just equivalent to *John* ψ*'s*. And this means that (7.18ab) are inconsistent given the perfectly innocuous assumption that John works both jobs. And this is clearly a terrible prediction for an analysis of *as* phrases to make. One might want to an inference from (7.18a) to (7.16a) and from (7.18b) to (7.16b). The New Puzzle, however, relies not on such inferences but just on the possibility that (7.18a,b) and (7.16a,b) can all be jointly true.

What can an extensionalist do? He could give up the possibility that (7.16ab) and (7.18ab) are jointly satisfiable, but that runs counter to quite robust intuitions. This would be to accept the conclusion of the puzzle—and to acknowledge, as we have seen, that restricted predication plays no essential role. He could reject Persistence; then the restricting predication need not hold in superstates (under $\sqsubseteq$) of those states in which it holds. But then, we must countenance in our present case the fact that there are some superstates of John's being a judge where he is *not* a judge. Since the logic is classical, it appears that we have no other option. But then it appears that we cannot say in this case that John is a judge, unless we are willing to say that he is not a judge as well. Either we say nothing or we say something contradictory. Neither of these options seems plausible. Another unpalatable way out of this puzzle is to give up Universality. But then one can't make sense of the intuitive incompatibility of (7.17ab), and the account of restricted predication suffers a similar difficulty as the standard Neo-Davidsonian view of predication.

The new puzzle infects any putative analysis of the Saul examples as containing hidden restricted predications. It seems impossible to distinguish situations or states in which Clark Kent is Superman from those in which Clark Kent is Clark Kent, if indeed Clark Kent *is identical* to Superman. If we cannot distinguish these states, then none of the sentences in (7.3) is satisfiable. Persistence and Universality imply that we must evaluate all claims about Superman as Superman and all claims about Superman as Clark Kent at the *same* maximal state of Superman. But then we cannot ascribe incompatible properties to Superman as Superman and to Superman as Clark Kent; the examples in (7.3) are predicted to be false. Finally, the New Puzzle affects the presuppositional part of the extensionalist accounts as well. It appears that the following sentence, though awkward, can be true:

(7.19) John as a judge is severe, but as a non-judge he is quite easy going.

Extensional accounts like Szabo's and Jaeger's predict that there must be a presupposition failure with (7.19) since the presupposition that John is a judge and the presupposition that John is not a judge must hold of John's present maximal state and they cannot both be accommodated or bound.

Though the extensionalists are right in rethinking the notion of predication in order to understand restricted predication, I think the New Puzzle shows that we need a fundamentally different approach. Restricted predication is fundamentally different from non-restricted predication, and the problem is that Neo-Davidsonians assume that relative and standard predication are in effect the same; relative predication only puts extra constraints on the stative element in the predication. The State Consistency Principle, which I take to be a fundamental principle guiding predication, cannot be consistently added to the extensionalists' view of relative predication. This shows that something else must be going on in relative predications. There is something irreducibly intensional about restricted predication, though the intensionality is of a different sort from that of attitude constructions. States are simply the wrong sort of object to analyse relative predication.

A sign that things have gone wrong is when we try to see whether relative predication has the marks of verbal predication—if events and states are involves then this constructionadmit adverbial adjuncts of the sort normally present in verbal predication. But clearly the *as* construction does not admit adverbial modification.

My idea is that restricted predications single out an aspect, or a ”guise” (to use an old term of Hector Neri Castaneda's), of an object to which the main predication applies, and that such aspects are intensional objects of a particular kind. *As* predications pick out individuals under the guise of a property contributed by the *as* clause's complement. An individual considered under the guise of a certain property or description is an aspect, and it is that aspect that figures in the predications of the main clause of a sentence containing an *as* phrase. Such aspects present the individual under some conceptualization, as we saw earlier. The approach using aspects analyzes relative predication as a phenomenon involving individual objects and clearly separates this from verbal predications involving eventualities. As such it is no surprise that adverbial adjunct modification is not a possibility in relative predication. The properties in the *as* phrase complement thus play a very special role in predication. They specify types that characterize the aspect of the object under discussion. The function of an *as* phrase is to introduce a type picking out a aspect of an object with several aspects. Many of the inferences that seem unproblematic for states need reexamination in this intensional setting.

7.5 *As* Constructions in TCL

TCL's technical implementation of informal talk about aspects is through the use of • types. Recall that • types can introduced by a special lexical item like *read* or by a grammatical construction—the *as* phrase construction; *as* phrases are • type introducers. *As* phrases coerce their subjects into something with several not necessarily spatiotemporal, metaphysical parts or aspects, one of which the complement or object of the *as* phrase serves to define. An object with several such aspects is modeled as an object of • type. In terms of the model of predication with complex types, an *as* clause requires the use of • introduction turning their subjects into objects of complex type, one constituent type of which is determined by the predication in the *as* phrase. In addition, *as phrases* force a restriction of the main clause predication to that metaphysical part of their subjects that is described in the *as phrase*. So for example, *John as a judge is corrupt* coerces *John* into having a complex type, one constituent of which is the type of being a judge, and it says that there is an aspect or metaphysical part of John—his being a judge—and that part is corrupt.

The types that *as* phrases introduce are sometimes contingent properties, rather than essential properties like that of being a physical object, an informational object or a place. John, for example, might cease to be a judge without ceasing to be John. But this is of a piece with the more flexible notion of type in current lexical semantics; the type FOOD is often used, but one could argue that something's being a food is a contingent property of the thing itself.

as phrases are a productive means for producing new dot types. We don't normally consider books as having an aspect of being paddles. Nevertheless, an *as* phrase can introduce such an aspect.

(7.20) This book as a paddle is useless. It's a fine piece of literature though.

What (7.20) says is that this book has an aspect or metaphysical part (among others) of being a paddle. Notice that the book isn't *just* a paddle though; the anaphoric pronoun refers back to the book in its normal nature, not its paddling aspect. This is an example of where we exploit the complex type in anaphora resolution.

As Jaeger and Szabo noted, the relative predication construction makes the book's being a paddle presupposed, and this puts limits on the acceptability of various relative predications. In (7.21) we see an attempt to presuppose information that is essentially incompatible with the basic type of the head noun:

(7.21) a. #The rock as an abstract object is interesting.

b. #The cigarette as an event lasted only 4 minutes.

Rocks essentially aren't abstract objects, and cigarettes though artifacts aren't events, and essentially so. It is thus impossible to accommodate such a presupposition, resulting in the infelicity of these examples.

At the level of logical form, the function of the complement of an *as* phrase is to specify a type for a variable *u* that serves as the argument in the main clause predication. But the *as* phrase also changes the type of the subject into a • type relating the term introduced by the subject in logical form to *u* via O-elab. So the logical form for *John as a judge is corrupt* should look like this:

(7.22) $\exists u(\textit{O-elab}(u, j) \wedge \textit{judge}(u) \wedge \textit{corrupt}(u)$ where u has the type JUDGE and j (John) has the type, ? • JUDGE)

John as a judge makes us think of John as having several aspects, one of which is that of being a judge. The others are left unspecified. This is reflected in the above logical form via the type constraints. So the *as* phrase coerces its subject into a complex type and specifies one of its aspects without specifying the complex type as a whole. By introducing such a type requirement on the aspect, the *as* phrase also introduces a presupposition—namely that the subject has an aspect of that type.

One might wonder why the *as* phrase should introduce a complex type at all. One intuition about *as* phrases is that the aspect they single out for predication in the main clause is, in some sense, one of several. When we speak of *John as a judge* as opposed to *John simpliciter*, the intuition is that we are speaking about a restricted aspect of John, and that John has other aspects or participates in other tropes besides that one singled out by the *as* phrase. As Szabo suggests, there is something comparative or contrastive about the *as* construction.So although an *as* only picks out one aspect of the subject, it implies that there are others. By having *as* phrases retype their subjects as having a complex, but not completely determined, type and by having the part specified by the *as* phrase be a proper part of the subject, we imply that there are other aspects to the subject.

This is close to the extensionalist view with one big difference. Aspects are not parts, as I've argued before and they're not physical parts either. Consider the categorization of a book as P • I. All of the book (in its material aspect) is a material object, *and* all of the book (in its informational aspect) is also an informational object. And so we can't conclude that the book *isn't* a material object (or an informational object) at some time, state or location. Nor is the book a mereological sum of its aspects, as I argued earlier. Nevertheless, those predications are predications restricted to different aspects. Restricted predications allow us to predicate incompatible properties of different aspects of objects; they also allow us to predicate properties of certain aspects that would be false of other aspects, indeed necessarily false, even if these predications hold at the same time. • types always give rise to a restricted predication when we ascribe to an object some property that is compatible only with one of constituent types. The big difference between this account and the extensionalist account is that predication to aspects is essentially different from normal predication. The extensionalist account tries to make do with states and is forced to say that if John is a banker (unrestrictedly) then the maximal present state of John is one where he is a banker. Universality then forces the extensionalist to make contradictory predications of that maximal state. On the present view Though John is a banker (untrestrictedly), this does not allow us to identify John with one of his (proper) aspects, the aspect of John as a banker.

While *as* phrases involve a predication semantically, the distributional facts suggest that they are not small clauses: as seen in (7.23), standard small clauses are restricted to certain argument positions and subject to strong restrictions on movement, whereas *as* clauses can modify many arguments and can move freely within the sentence. *as* phrases more closely resemble contrastive or comparative clauses:

(7.23) a. The doctor examined the patients naked.

b. ?The patients naked the doctor examined.

c. John squashed the book flat.

d. ? The book flat John squashed.

e. *John gave the book flat to Mary.

f. John gave the book as a present to Mary.

g. John gave the book to Mary as a present.

h. As a present John gave the book to Mary.

i. John gave the book rather than the stereo to Mary.

j. Rather than the stereo, John gave the book to Mary.

k. John gave the book to Mary rather than the stereo.

Comparatives (Heim, **?**) are usually analyzed as involving a degree function. *As* phrases don't have that but they have a similar structure, introducing an aspect that plays a role both in the main and secondary predication. I assume that *as* takes a trace of a noun phrase or DP that plays a role in the main clause—this is the *as* clauses's subject —-and another DP—in the complement of the *as* phrase, to form an IP that is adjoined to the main IP (much like a comparative). *as* takes the trace, its complement DP and the main clause IP as arguments—something which is necessary since *as* affects or places constraints on all of these elements semantically. The DP whose trace is in the subject of the *as* clause, I assume, undergoes quantifier raising so that it can bind a variable both in the small clause and in the main clause.[9]

as phrases also restrict the type of DPs that can serve as complements:[10]

(7.25) a. John as a doctor is competent.

b. ??John as every (any) doctor is competent.

c. ??John as many things is competent.

French and Spanish don't even have full DPs in complement position:

(7.26) a. Jean comme médecin (*un médecin, *le médecin, *tout médecin)

b. Maria como avogada

This suggests that the complement of the *as* phrases contribute a property, not a full DP, to interpretation. The property conveyed by the object NP of the *as* phrase's complement DP determines a type that picks out an aspect of the subject of the *as* phrase.

When we apply *as* to the content of the main clause, we coerce the type of the variable of the subject of its complement IP into that of a complex type, and the complement DP of the *as* phrase must specify a part or aspect of that complex type but that part is also the subject of the main predication. Thus, the meaning of *as* will embody a type constraint and also triggers the an instance of a type coercion, the introduction of a $\bullet$ type.

[9] The trace can either be in subject position, as is the case in most of our examples or in other argument positions of the main verb as in(7.24):

(7.24) John loves Mary as a janitor.

[10] They seem to be able to take definites and also possessive constructions as well as indefinites in English, but this is not true in other languages

Let's look at a derivation of the logical form of *John as a judge is corrupt*. The type assigments occur in angle brackets.[11]

(7.27) $\lambda Q \lambda w \lambda P \exists v(\text{O-Elab}(v, w) \wedge Q(v) \wedge P)$
with the type context $\langle v : \top,\ w{:}\ \top \bullet \top,\ P : \textsc{t};\ Q : \top {\Rightarrow} \textsc{t}\rangle$

We now combine this with the *as* phrase's complement using standard rules for type unification:

(7.28) $\lambda w \lambda P \exists v(\text{O-Elab}(v, w) \wedge\ judge(v) \wedge P)$,
with the type context $\langle v{:}\ \textsc{judge},\quad w : \top \bullet \top\rangle$

By considering the type of an aspect, we can reassign the type of the object whose aspect that is:
$\text{O-Elab}(v, w), \langle v{:}\ \textsc{judge},\quad w : \top \bullet \top\rangle \vdash \text{O-Elab}(v, w), \langle v{:}\ \textsc{judge},\quad w : \top \bullet \textsc{judge}\rangle$

We combine this with the subject of the *as* phrase, which is a variable x_i coindexed with the trace in the main clause and with the subject DP of type e:

(7.29) $\lambda P \exists v(\text{O-Elab}(v, x_i) \wedge\ judge(v) \wedge P)$,
with the type assignment: $\langle x_i : \top \bullet \underline{\text{judge}}\rangle$

This now combines with the translation of the main clause.

(7.30) $\exists v(\text{judge}(v) \wedge\ \text{corrupt}(v) \wedge \text{O-Elab}(v, x_i))$,
with the type assignment: $\langle v : \underline{\text{judge}}, x_i : \top \bullet \underline{\text{judge}}, \rangle$

We now finally combine with the subject DP:

(7.31) $\exists x_i(\text{john} = x_i \wedge \exists v(\text{judge}(v) \wedge\ \text{corrupt}(v) \wedge \text{O-Elab}(v, x_i)))$,
with the type assignment
$\langle j, x_i :? \bullet \underline{\text{judge}}, v : \underline{\text{judge}}\rangle$

On this analysis, *as* phrases serve principally to type the argument that they introduce. But they also of course are factive predications, as the logical form makes clear. Note that an intensional version of Univerality holds on this account too, and trivially so since the *as* phrase always picks a unique aspect of John of that type; thus, if John as a judge is corrupt, we cannot consistently say that John as a judge is also honest. However, Persistence does not hold when we move a predication from aspects to the whole object of complex type, and this is essential

[11] Once again, I'll ignore the presuppositional status of the *as* phrase's content here.For details as to how such an account might go, Van der Sandt (1992).

for solving the New Puzzle in this framework. John may be corrupt as a banker but not as a janitor, and we can't infer in this account that John himself is corrupt simpliciter, unless by that we simply mean that there is some aspect of John and John under that aspect is corrupt. Predications on aspects of an object do not automatically transfer into predications of the object of complex type. To say of a book that it weighs five pounds is only to say of its physical aspect that it has that property; its informational aspect does not have that property. Further, this notion of predication of aspects allows both presuppositions in (7.19) to be consistently accommodated because on this account the presupposition differs from those in the extensional account; the presuppositions here are that there is an aspect of John that is a judge and an aspect of John that is a non-judge.

One might argue that my account of predication is too weak to be plausible. But it's quite restrictive with regard to restricted predications: it is impossible for John as a banker to be both corrupt and not corrupt. Matters are otherwise with predications that are not restricted according to the grammar. Consider any pair of apparently contradictory sentences like *I am happy now* and *I am unhappy now*. Doesn't my account make these automatically compatible? On my account such predications may, or may not, be restricted predications. They are contradictory predications of me if they are unrestricted. But the preferred interpretation of a sentence like

(7.32) I am happy and unhappy now.

is one according to which we infer two restricted predications; i.e., there is an aspect of me that is happy and one that is unhappy; when we get incompatible predications on a single term, we read them as restricted predications. Nevertheless, 7.32) if not contradictory feels incomplete; the interpreter awaits a specification of the aspects in which I am happy and I am not happy. As I have noted, this is not always an available strategy—viz, for (7.15) repeated below.

(7.15) a. John makes $100K a year.

b. John makes less than $70K a year.

Neverthess, some inferences that fall under a persistence like principle do go through. Consider (7.33) and (7.34).

(7.33) a. John as a judge makes $50K a year.

b. John makes at least $50K a year.

(7.34) a. John as a judge in town A makes \$50K a year.

b. John as a judge in town B makes \$60K a year.

c. John as a judge makes at least \$110K a year.

Some predicates persist as predications from parts to larger parts or wholes. If an aspect or part of John makes \$50 K a year, then that part makes at least \$50K, and making at least X is upward persistent in the relevant sense. The partial ordering I've defined over aspects or tropes allows us to define for additive properties (like salary) a homomorphism from **P** to a domain of quantities so that we can sum up properties across aspects. But in general whether such persistence applies to other properties, like that of being a judge, will depend on the different sorts of restricted predications attributed to the object.

In addition consider the following sentence:[12]

(7.35) John as a banker is the same person as John and the same person as John as a salesman.

This example, though a bit unnatural, is perfectly intelligible. But I believe that we can make this statement true with a minimal number of stipulations. Let's assume that if one aspect a stands in the partial ordering to b, then at least defeasibly if $\phi(b)$, then $\phi(a)$. In other words this partial order over aspects is very similar to the subtyping relation itself, in that it supports defeasible inheritance. Now clearly *Same-person*(j, j) (John is the same person as John). We've identified the bare object John with the aspect $(j, \top)$, so we can say that the aspect of John as a banker stands in the requisite relation to John. Assume now, as is plausible that *Same-person*(x, y) is not only reflexive but also transitive and symmetric. It is easy to conclude given defeasible inheritance that *Same-person*(x, j) where x is the banker aspect of John and also that *Same-person*(y, j) where y is the salesman aspect of John. From these and the properties of the predicate *Same-person*, it quickly follows that *Same-person*(x, y), as desired.

More generally, it appears that the partial order on aspects functions with respect to predication in much the same way as the subtype relation does, which is not surprising as one is defined in terms of the other. Typically, if $\alpha \sqsubseteq \beta$, an inhabitant of α has all the properties that an inhabitant of β does, but not vice-versa. Sometimes the inheritance by an inhabitant of α of properties associated with β is blocked because of more specific properties associated with α. For instance HOUSECAT $\sqsubseteq$ CAT and Tasha, a house cat, inherits many properties associated with

[12] Thanks to Gabriel Uzquierro for this example.

CAT, the properties of having fur, being four legged, having retractable claws, being able to climb trees. An element of the general type won't inherit typically Tasha's properties—being gray and white, weighing 3 1/2 pounds and so on. CHEETAH is also a subtype of CAT but cheetahs don't inherit the property of having retractable claws. The same thing holds of aspects, and not surprisingly since the ordering on aspects derives from an ordering of their types under logical entailment, which naturally extends to an ordering with respect to $\sqsubseteq$. If John is a person, then John the banker is also a person, and John the banker is even the same person as John. Persistence of properties in general fails; more ”general” aspects, higher up in the partial ordering (recall that the object of complex type is at the top of this partial order), fail to inherit the properties of more specific aspects.

With this understanding of predication on aspects, we can make some headway into understanding the recursively embedded *as* phrases– one *as* phrase within the scope of aother.[13]

(7.36) John made good money as a chimneysweep as a youth.[14]

On the current proposal such embedded *as* clauses produce an aspect of an aspect; there is an aspect of John, John as chimneysweep and an aspect of that, which I'll call John as young chimneysweep. Given the partial ordering on aspects the we have:

- John as young c $\sqsubseteq$ John as c $\sqsubseteq$ John.

So the properties of John as a chimneysweep are defeasibly inherited by the aspect of John the chimneysweep in his youth. For instance John as a chimneysweep in his youth is a chimneysweep. Temporal aspects like that picked out by *as a youth* will pick out a temporal slice of John, and this affects predication roughly in the same way as a temporal adverb does but by a rather different mechanism. Thus, John as a chimneysweep in his youth will be temporally located to the time when he was a young man.

How do the Landman axioms fare on this view? Consider axiom (2):

2. If John as a judge is corrupt and John as a judge is well-paid, then John as a judge is well-paid and corrupt.

The logical form for the antecedent of the conditional will yield:

[13] Although *as* phrases are similar to comparatives in a number of respects, they differ in one important respect. Contrastives do not embed happily one within another.

[14] Thanks to John Hawthorne for this example.

- $\exists y(\text{judge}(y) \wedge \text{O-elab}(y, j) \wedge \text{corrupt}(y)) \wedge \exists z(\text{judge}(z) \wedge \text{O-elab}(z, j) \wedge \text{well-paid}(z)))$
  with the type assignment
  $(j :? \bullet \textsc{judge}, z : \textsc{judge}, y : \textsc{judge}$.

By our axioms on types, as they are both O-Elaborations of the same object and have logically equivalent types, $y = z$. And this yields the desired conclusion:

- $\exists y \wedge (\text{judge}(y) \wedge \text{O-elab}(y, j) \wedge \text{corrupt}(y))) \wedge \text{well-paid}(y)))$

Axioms (3,4, 6) just follow as simple inferences or instances of theorems of classical logic or by the persistence axiom for types.

The only problematic axiom is (1), at least if we assume that the *is* in (1) is the *is* of identity:

(1) John as a judge is John.

What we get for (1) is the following logical form:

- $\exists y(\text{judge}(y) \wedge \text{O-elab}(y, j) \wedge y = j)$.
  with the type context
  $\langle j :? \bullet \textsc{judge}, y : \textsc{judge}\rangle$.

This is not valid. The aspect involving John that the *as phrase* picks out isn't identical to the (the whole of) John, or John the thin individual. It can't be, for if it were, we would have demonstrated that a complex type and a simple type have a common inhabitant, which is something we demonstrated to be impossible. An object of • type can't be identical to an object of a simple type.

Rejecting such instances of axiom 1 is essential for addressing the behavior of proper names in this framework.

(1') Superman as Clark Kent is Superman.

(1") Superman as Superman is Superman.

Suppose that (1') yields as a logical form where $\phi_{CK}(y)$ stands for whatever formula involving y we see fit to analyze the contribution of the name *Clark Kent* when it occurs as a complement in an *as* phrase and R is any relation between y, which is the subject of the *as* predication, and the whole sentence's subject s, which is Superman.

(1"') $\exists y(\phi_{CK}(y) \wedge R(y, s) \wedge y = s)$.

Suppose that (1”) yields an analogous logical form. Now reconsider (7.3a), according to which Superman as Superman gets more dates than Superman as Clark Kent does. Using the logical forms for (1’) and (1”), we’ll be able to substitute within (7.3a) Superman everywhere for any guise of Superman—in particular Superman for the Clark-Kent part of Superman. But this now yields that Superman gets more dates than Superman, which is necessarily false.[15]

Finally, Landman’s axiom 5 must be understood in a certain way. It does follow that if John as a judge is corrupt (ignoring presuppositions), then John is a judge. But the predication in *John is a judge* is still a *restricted* predication; all that *really* follows in this account is that there is some aspect of John under which we can truly say that he is a judge.

7.6 Aristotelian *Qua* Phrases

Aristotle, who perhaps discovered restricted predication, holds a view according to which (7.37a) is false but (7.37b) is true.[16]

(7.37) a. An isosceles triangle as such (i.e. as an isosceles triangle) is such that the sum of the interior angles = 180 degrees.

b. An isosceles triangle as a triangle is such that the sum of the interior angles = 180 degrees.

The logical form for these sentences is not entirely clear. Many read the predication as giving necessary and sufficient conditions for something’s being an isosceles triangle qua isosceles triangle or at least a necessary condition— i.e. the predication is that of an ’iff’—or of an ”only if” statement. On my account of restricted predication, the definitional reading yields as logical forms for (7.37ab) (assuming some sort of unselective quantification over x and ignoring type assignments):

(7.37a’) $\forall x, y((\text{Isos-trian}(x) \wedge \text{O-Elab}(y, x)) \rightarrow (\text{Isos-trian}(y) \leftrightarrow$ $\text{the sum of the interior angles}(y) = 180^\circ))$

(7.37b’) $\forall x, y((\text{Isos-trian}(x) \wedge \text{O-Elab}(y, x)) \rightarrow (\text{Trian}(y)) \leftrightarrow$ $\text{the sum of the interior angles}(y) = 180^\circ))$

[15] We could interpret the *is* in , e.g., (1’) differently from the *is* of identity, perhaps as a predication of constitution. Thus, *John as a judge is John* would say that the part of John that is a judge is a part of John, which is completely uncontroversial.

[16] Szabo (2003) makes this point. See also *Sophistical Refutations* 167a7-9.

(7.37b’) is clearly provable and necessarily true, whereas (7.37a’) is not (at least not in the right to left direction). If these are the right logical forms, then we have an explanation of Aristotle’s intuitions about these sentences.

7.7 Proper Names in *As* Phrases Revisited

I return now to the role of proper names within *as* phrases. The present account predicts that names as complements of *as* constructions should be peculiar, because the complement of an *as* phrase are properties with the function of introducing a type that picks out a part of the subject and it’s very unclear how proper names define a type. If names simply refer to the individual, it would appear as though they should introduce the type e; further, understanding a name as a property of the sort $\lambda x x = n$ doesn’t pick out any type in the hierarchy at all, since this expression just picks out the object n! Some names, however, can denote types: names that are associated explicitly with roles in plays—*Hamlet, Lear, Faust, Aunt Augusta,* etc. *Superman* and *Clark Kent*, or to give a real life example, *Jesse Ventura* and *The Body*, which was Ventura’s moniker as a pro-wrestler, also seem to work this way. This suggests that there is a coercion of the meaning of a name when it is used as a type. Proper names denoting types *don’t* have their customary denotation: they pick out roles or concepts that objects can fall under, aspects that are proper parts of the objects they customarily denote.[17]

Thus, a predication involving proper names in *as* phrases changes from something of the form $\lambda P P(n)$ to predications of the form $\lambda P \exists x(\text{n-role}(x) \wedge P(x))$. This immediately blocks substitution of coreferential names within type designations, because coreferentiality of names doesn’t imply anything about the equivalence of the roles associated with them; the fact that $a = b$ doesn’t imply that the two roles associated with the names are the same. In this my analysis of *as* phrases is Fregean in spirit. But my analysis (with a standard, intensional analysis of propositional attitudes added) also implies that the thought that $a = b$ doesn’t imply that *x as a* is intensionally equivalent to *x as b*. In this my analysis differs importantly from a Fregean one of the sort discussed earlier. On the other hand, property denoting complements of *as* phrases aren’t predicted to be opaque in the same way, as no coercion to a role meaning is required. Logical relations between predicates are reflected in the partial order on types and our axioms for aspects will allow substitution of logically equivalent predicate expressions.

[17] This is a vindication at least in part of a Neo-Fregean approach: the predication involving a proper name is one of ascribing a role of some kind.

Following the outlines of the derivation in (7.27-7.31), we get (in the formula 's' stands for Superman):

(7.3a.1) $\exists y(s = y \wedge \exists z(\text{S-role}(z) \wedge \text{O-elab}(z, y) \wedge \exists x(s = x \wedge \exists w(\text{CK-role}(w) \wedge$ $\text{O-elab}(w, x) \wedge$ gets more dates than $(z, w))$
with the type context $x :? \bullet \text{S}, y :? \bullet \textsc{ck};\ z : \text{S};\ w : \textsc{ck}$.

We can further simplify this to:

(7.3a.2) $\exists z(\text{S-role}(z) \wedge \text{O-elab}(z, s) \wedge \exists w(\text{ck-role}(w) \wedge \text{O-elab}(w, s)$
$\wedge$ gets more dates than $(z, w)))$
with the type context $s :? \bullet \text{S} \bullet \textsc{ck},\ u : \text{S},\ v : \textsc{ck}$.

So Superman, in virtue of the roles that the names *Superman* and *Clark Kent* pick out, turns out to have several aspects and these aspects are what play a role in the main predication of (7.3a). Superman is forced, if this sentence is true, to have the types? $\bullet\ \alpha$ and ? $\bullet\ \beta$, which naturally yield the type ? $\bullet\ \alpha \bullet \beta$ for Superman.[18] The *as* phrases each pick out a relevant part of the individual, which serve as the arguments in the main predication.

Let's return briefly to Saul's original examples (7.1-7.2), I've suggested that they contain hidden *as* phrases. But how do they arise? Perhaps the comparisons and contrasts between Clark Kent and Superman license the comparative *as* constructions. Or perhaps it's because, these examples yield contradictory or incompatible predications. The threatened inconsistency of this situation triggers, as I have already suggested, a restricted predication of the sort given by the natural paraphrase of (7.1-7.2) using *as* phrases. The complex typing coercions of the sort of restrictive predications found in, *inter alia*, *as* phrases may happen for such pragmatic reasons.

This semantic analysis of *as* phrases is compositional, avoids the problems of the New Puzzle and explains why proper names behave strangely in otherwise apparently extensional contexts. It is a natural application of the conception of thick and thin individuals needed for copredication. In fact, the Saul examples show us that in fact, thick individuals are common subjects of predication. Relative predication is in fact a much more widespread phenomenon than one might think.

[18] For this inference, we need some additional axioms for underspecified types that I can't give here.

7.8 Plural Relative Predications

So far we have looked at *as* phrases only involving noun phrases that denote or quantify over single individuals. *As* phrases also modify predications in which one term denotes a collection or plurality as in:[19]

(7.38) a. John and Mary as a couple are a pain to have at a party, though John and Mary individually are fine.

b. The students as individuals are well behaved, but as a group they are not well behaved.

We've already seen at the beginning of this book that plurality affects predication. There are certain nouns like *couple*, *team* that require a non singular argument.

(7.39) # John makes a nice couple.

(7.40) John and Mary make a nice couple

(7.41) They are a good team.

Further certain predicates like *disperse* or verb phrases like *surrounded the building* require a collective sort of plural argument:

(7.42) a. Most students in the square dispersed (surrounded the building).

b. The students in the square dispersed (surrounded the building).

Other predicates like *smoke* take distributively read arguments or at least can. This points to a distinction in the type system not only between plural and singular types, but between different types of plural types, like COLLECTIVE and DISTRIBUTIVE.

Predicates can coerce plural terms into distributive or collective readings. When we get to examining coercions in detail, we will see that this means that there is a map or ”arrow” from collections to their members and back again. Such coercions will make use of dependent types, which I'll introduce in a couple of chapters. This construction turns out to be quite different from aspect selection for dual aspect nouns for rather subtle reasons, in particular with respect to the behavior of quantifiers in coercive contexts. In examining Nunberg's and classic GL's analyses of coercion, we saw that unlike the quantificational shifts evident with dual aspect nouns, coercions do not shift quantificational domains.

[19] Thanks to Friederike Moltmann for pointing these sort of examples out to me.

Despite these differences there is, as with many things in natural language metaphysics,[20] a certain flexibility in how we conceive of objects. *as* phrases appear to be able to make coercions involving plurality aspects of the denotation of the plural term. That is, relative predications can at least sometimes, *modulo* the proviso about presupposition accommodation, turn meanings associated with coercions into aspects of objects. For example, (7.43a) turns into (**??**

(7.43) a. The students as a group

b. $\exists x \exists y \text{O-elab}(x, y) \wedge \text{Students}(y) \wedge P(x)$, $\langle x\colon \textsc{collection}, y\colon \textsc{collection} \bullet \textsc{student}\rangle$

The approach I have argued for introduces two variables one for the collection of students and one for the students of complex type. We have seen also that the denotations of these entities must be distinct, but other than that the formalism is mute. These variables could in principle pick out different entities in the model. There are a number of accounts of plural predication according to which the individuals and the group will be distinct entities in the model—all those, for instance, that take as their starting point Link's (1984) lattice of individuals and plural sums. With these the aspects account is fully compatible. We only need to distinguish say for (7.38a) the set of the parts of the sum from the sum itself. For other approaches, e.g., those based on sets of assignments (Fernando 1993, Krifka 1996, Van den Berg 1996, Asher and Wang 2003), things are more complex. These approaches work with a simple domain of individuals, unlike Link's lattice construction that makes individuals and their sums distinct elements in the model. For these approaches our aspects must be understood as referring to the way the individuals are "packaged", either as individuals or as sums. This is not really that unintuitive. There are lots of devices that can act either on denotations or linguistic packaging—-negation, corrections, presupposition accommodation. The difference in types need not correspond to a distinction in entities in the model. The use of • types picks out a general metaphysical structure that can be specified in various ways (ways of looking at pluralities, aspects of individuals)

7.9 An Aside on Depictives

Depictive constructions are similar to relative predications in that they make two predications of an object. That is, in (7.44a) John has the property of being drunk

[20] See Asher (1993) for a discussion of how there are maps from propositions to facts to events, leading to a slippery slope along the spectrum of abstract entities.

and of driving home.

(7.44) a. John drove home drunk.
b. The doctor examined the patients naked.

These examples are often treated (cf Rothstein 2003, Kratzer forthcoming) using event semantics. The idea is that this construction introduces two events linked in a variety of ways. Within event semantics, it seems that we need to stipulate the relations between the events introduced in order to get the right temporal connections—namely, that John drove home while drunk. A bit of syntactic ingenuity is also needed to make *John* in (7.44a) the argument of the adjunct *drunk*.

Our type driven approach to predication provides a simpler approach to these constructions. One hypothesis is that depictives are in fact relative predications. That is, they pick out an aspect of the subject that is then the argument to the main predication. This would immediately license the desired temporal relations between the state of John's being drunk and his driving home. That is, I hypothesize that driving home is predicated of John as drunk. Hence, he must be drunk throughout the driving home. There is a difference, however, between depictives and *qua* predications. Relative predications with adjectives, especially stage level adjectives like *drunk*, attribute a temporary property of their argument; thus, John as drunk is a *temporal part* of John. And so in this depictive constructions are similar to coercions with eventualities, which we'll study in the next chapter—they pick out an aspect of their arguments that has a restricted temporal extent.

7.10 A Comparison of Frameworks

The analysis of restricted predication also teaches us something about thick individuals or tropes. One might think in fact that they are modes of presentation. But we've seen that thick individuals or tropes do not function as modes of presentation in attitude contexts. Predications of one thick individual *b* that is a trope of a thin individual *a* do not carry over to another trope within belief contexts, even when the believer knows that the tropes involve aspects of the same thin individual. This shows that predications are often restricted to aspects. Now the argument that I gave above doesn't, of course, establish that tropes couldn't *also* function as modes of presentation. It is a possible analysis of attitudes that a subject always entertain an attitude with respect to some object under a conceptualization—that is, relative to some aspect of that individual. That is, one would entertain the thought of an individual *as* having a certain property.

Something like the conception of a thick individual is already a staple of certain acounts of attitudes in dynamic semantics. At least with respect to attitudes about objects that are familiar to the agent, the analysis of these attitudes say in DRT requires a complex of attitudes, some of which contain information specifying in what respects the individual is familiar (Asher 1986, 1987, 1989, Kamp 1990). These parts of an agent's cognitive state are called *internal anchors*. A completely satisfactory attitude report according to these accounts would capture not only the content of the attitude but also at least the basic structure of the internal anchors. To take the well-known case of identity statements involving singular terms, suppose that $t = t'$ but that the agent does not believe this because, in the framework of DRT, his internal anchors are distinct for t_1 and t_2. Then an appropriate report would capture this by having distinct anchors for the variables or discourse referents introduced by t_1 and t_2.

From the present perspective, such internal anchors might be assimilated to aspects. Presumably the doxastic agent of the previous paragraph thinks of the referents of t_1 and of t_2 as having different properties and that is why he does not identify them. In other words he thinks of these objects under different aspects. That is, agents identify individuals in terms of one or more their aspects, even though their attitudes involve, at least for so called *de re* attitudes, predications of the thin individuals. That is why, if an agent who earlier had assumed that two aspects that figured in his attitudes belonged to different thin individuals later realizes that two aspects belong to the same individual, he is forced to adjust his *de re* attitudes. On this view, agents have attitudes towards thick individuals or aspects, even though the content of the attitude may involve normal, unrestricted predications to the thin individuals whose aspects are constitutive of the agent's attitudes. As the DRT accounts argued already in the 80s (Asher 1986, 1987, Kamp 1987, 1990), there is a matter of lattitude in how attitudes are reported; since the aspects that believers have in mind when they have an attitude about an object are often inaccessible to others, the best reporting strategy is often simply to give the content of an attitude in terms of unrestricted, simple predications. By retracing the DRT story about internal anchors with aspects and thin individuals, it seems that one might come up with a non representational version of that story. I will leave the matter here, however, as the attitudes are not a primary concern of this book.

The use of aspects, of thick and thin individuals, also offers an interesting perspective on event semantics. Event semantics is not just one theory but a whole range of theories on which eventualities of various sorts play many roles. Davidson (Davidson (1968/69)) proposed that action verbs have a hidden eventuality argument that could serve as an argument to adjuncts to the verb phrase. Since then, it has become standard practice to extend this usage to all verbs and to suppose that verbal inflection binds these event arguments and localizes them in time (Abney 1987). Eventualities play an undeniably useful role in the compositional semantics of a verb and its syntactic projections, but event semantics has tended to rely heavily on these entities for many other duties in semantics as well. They have been postulated as hidden arguments of adjectives and even nouns. We've seen in this chapter how extensional treatments of relative predication that appeal to some version of event semantics get into trouble by using eventualities in various roles for which they were not designed. At bottom there are a host of ontological and conceptual questions that practicing event semanticists rarely address: how are events individuated and how are they related to individuals? The mechanism of aspect construction from *as* phrases is general enough to pick out temporal parts of individuals—consider *as a youth*. It also subsumes aspects that involve how one conceives of a set of entities (the collective distributive distinction). The notion of aspect thus can do some of the work that eventualities are supposed to do in Neo Davidsonian theories. And aspects, unlike events, have clear identity criteria. They are also more general and so can subsume event like aspects singling out moments of time or portions of space time.

Asher (1993) argued there that there were maps, what I shall call dependent types, from facts and even propositions to eventualities. As we shall see in the next chapter, there are ample reasons for postulating maps from eventualities to individuals or their aspects. That is, an eventuality may single out an aspect of an individual. The generativity of the *as* construction points to a fluid transition between aspects and coercion, which is the subject to which I now turn.

Chapter 8

Rethinking Coercion

With our conceptual and formal analysis of • types and copredication involving them in place, I turn now to the more general phenomenon of coercion and logical metonymy. The classic cases of coercion involve some sort of ”shift” in a predication from the predication of a property to an object to the predication of a property to an eventuality of some kind. Here is one of motivating examples.

(8.1) Sheila enjoyed her new book.

(1.36a) has the standardly preferred reading that Sheila enjoyed reading the book. However, if the discourse context makes contextually salient that Sheila is an author, then the preferred reading is that Sheila enjoyed writing the book. If the discourse context makes contextually salient that *Sheila* is the name of a cat, then the event enjoyed is perhaps something like clawing, scratching and biting the book. Classic GL, which attempted to predict the different readings for the predication with the coercion from the qualia associated with the direct object of the verb, was unable to model these context dependent inferences.

The dependency of the coercion on several arguments of the coecer verb is a robust lexical phenomenon. Recall these examples from the first chapter.

(3.35) a. #Smith has begun the kitchen.

b. The janitor has begun (with) the kitchen.

c. The cleaners have started the suits

d. The exterminator has begun (with) the bedroom.

e. The painters have finished the windows.

It is clear that the information relevant to determining the eventuality that is the object of the aspectual verb cannot come from the argument that is undergoing the coercion in (3.35). Part of my task here will be to provide a formalism within which contextually available information can play a role in determining the type of the argument of a verb like *enjoy* or of aspectual verbs like *begin, start,* and *finish.*

In these classic coercion cases, we have a shift from one type to another in logical form—-often from an object to an eventuality associated with it. These sorts of type coercions do not in any obvious way involve a simplification of a complex type. That is, there are no intuitions that support the view that *kitchen* or any other noun referring to an artifact or natural kind denote an object that is both an eventuality and an artifact. Thus, *kitchen* is different in nature from, say, *lunch* or *interview*. The linguistic evidence for this intuition is that while *lunch* takes all sorts of predicates of eventualities straightforwardly, *kitchen* does not:

(8.2) a. The lunch starts/finishes at 12:30.

b. The kitchen starts/finishes at 12:30.

Eventuality readings of *the kitchen* in (8.2b) are possible, but they are much facilitated by invoking a particular discourse context, one, say, in which a visit to the kitchen is scheduled. The difference between (8.2a) and (8.2b) is clear though subtle. We have also seen a difference in the behavior in the selection of quantificational domains for dual aspect nouns versus traditional coercions.

If the nouns in (3.35) are not plausibly dual aspect nouns and do not involve a selection of an aspect, they nevertheless readily invite a ”transversal” shift in the type hierarchy from one type to another. When they are arguments to aspectual verbs we have a shift from some type of object to some type of eventuality. Sometimes these shifts are lexically determined but as we have seen they are often determined by the context. For instance, which eventualities end up being their arguments is not just a function of the direct objects themselves. (3.34), among other examples, show that sometimes the subject argument will end up determining the eventuality. Other examples show that it is in fact the larger discourse context that ends up determining the eventuality.

One question that crops up immediately is what is responsible for the coercion in these cases? It seems that aspectual verbs are responsible for the event coercion, though by themselves they don’t determine which sort of events end up being their object arguments. A test of this hypothesis is to consider a completely made up word like zibzab.[1]

(8.3) Sam enjoyed the zibzab.

When hearers are asked about this sentence, most of them report that he enjoyed doing something with the zibzab, but they have no idea what. The event interpretation can't derive from the lexical meaning of the noun, since it's made up and has no lexical meaning. The event interpretation, underspecified as it is, must come from the verb.

Another argument that the eventuality should come from the verb and *not* be part of the noun's meaning is this. Recall that event readings of nouns like *book* are not always available:

(8.4) a. The reading of the book started at 10 am.
b. #The book started at 10 am.

The predication in (8.4a) seems fine but weird in (8.4b). It seems difficult to deny phrases like *the book* an event reading if events are associated with *book* in the lexicon, as is the case in classic GL. Thus, it appears that only certain predicates should license the introduction of an associated eventuality with nouns like *book*.

Coercions from object types to eventuality types are widespread. Adjectives and other verbs that require noun phrases denoting sets of eventualities as their arguments can sometimes coerce their non eventuality arguments into denoting events, though some coercions are much better than others.

(8.5) a. That person is slow (in understanding things, in running, etc.)
b. The slow animal (the animal that is moving slowly)
c. ?The slow tree (the tree that grows slowly)
d. The event lasted an hour.
e. John lasted an hour (say in playing tennis)
f. The event started at 12 o'clock.
g. John started at 12 o'clock (started to run, to bicycle or to climb)

Even when there are no clearly associated qualia with a noun (even an artefactual noun which is supposed to come with qualia in classic GL), the aspectual verbs still coerce their arguments into eventualities. It is just that we don't know what these eventualities are, though of course appropriate discourse contexts can make precise the eventuality enjoyed in say (8.6):

[1] Thanks to Chris Kennedy for these sorts of examples.

(8.6) John enjoyed the bathroom.

To model coercion, I will resort to underspecified dependent types. Underspecification in the type system is familiar; I used it in the last chapter to analyse relative predication. On the other hand, dependent types are a new kind of complex type. Dependent types are a species of arrow objects in the abstract, category-theoretic model for typesBarr and Wells (1990), Asperti and Longo (1991). They are like the traditional functional types in that respect; when their parameters are specified they yield another type. Unlike functional types, however, their arguments are filled not by the process of application but by the process of coercion.[2] Functional types define higher order types, whereas dependent types do not. A dependent type encodes a dependency between the type α of some term and other types, one of which is designated a ”value” of the dependency relative to α and other types which I'll call *paramters*. If a term *t* is assigned a dependent type, then it will have the type determined by the function once the parameter types are determined, even though *t* does not *apply* as a lambda term to terms whose types are those parameters. Typically the parameters are types of terms that are arguments to the predicate in the coercing predication. Dependent types permit us to model how the sort of eventuality that is coerced from the meaning of the verb's direct object may depend on other arguments of the verb, as in the case of *enjoy, begin, start* or *finish*. Sometimes this function's value is not specified, for instance in (8.7) except in particular contexts (for instance, where it's known that Lizzie enjoys climbing rocks in general):

(8.7) Lizzie enjoyed the rock.

In this chapter, I will show how to make sense of such contextually dependent types as well.

Dependent types have many uses. Among them are the following:

- They enable us to define certain general types. For instance, we may define artifacts as any type on which the dependent type PURPOSE is specified and EFFICIENT CAUSE is typed as an agent.[3]

[2] Dependent types are also similar to the records used by Cooper in the framework of Type Theory (Cooper 2002, 2005). Records are functions from types to other types or even type assignments.

[3] This is something that Pustejovsky has recently explored in detail. These dependent types PURPOSE and EFFICIENT CAUSE are of course the qualia of classic GL. Note, however, that dependent types permit us to say that artifacts must have human or agentive causers; e.g., CAUSE(ARTIFACT) = ϵ(AGENT). ϵ is an underspecified dependent type whose value is some sort of eventuality type, to be specified by its other parameters or its context. I will make extensive use of ϵ shortly.

- Dependent types, as I intimated back in chapter 2, allow us to specify fine-grained subtypes of the type of propositions, as well as many of many other types. They will also allow us to state generalizations about the presuppositions of various word classes.

- Predications involving the aspectual verbs (*begin, start, continue, finish, end*) , verbs like *enjoy* but also purely temporal verbs like *last* will trigger the introduction of dependent types. Aspectual verbs will coerce their theme arguments to have an eventuality type. And this will license an application of a dependent type from entities to eventualities on the terms serving as the theme arguments. In some cases the type may be lexically specified, but in other cases the value of the dependent type is underspecified and may determined by the predicational (by which I include the other arguments of the coercing verb) or discourse context.

- We can model many other cases of coercion with dependent types: the coercions brought about by material adjectives, sound verbs and the Nunberg style examples. One can also consider, given my remarks on fictional objects, that fictional contexts may be an example of coercion on terms occurring within that context. Grinding, the conversion of an object into a portion of matter or mass is also amenable to this treatment, as is nominalization.[4]

- The genitive construction will also trigger the introduction of dependent types. The genitive construction requires us to relate the variable introduced by the head noun with the variable introduced by the DP in the genitive. My hypothesis is to use dependent types for this as well. These types can be specified either by a relation that helps define a complex type of the noun in the complement of the genitive construction or by the noun's head type itself if it is a relational noun.

- In other sorts of predications like predication in fictional contexts, it seems as though we allow the discourse context to reassign types to objects. Once again this points to a need for dependent types and a complex interaction between lexical and discourse theories.

The introduction of dependent types may be grammatically determined as in nominalisation or grinding above. But others may not be overtly marked in the gram-

[4] One can also think of Partee and Rooth style type shifting in this veinPartee and Rooth (1983)
.

mar and may be tied to the lexical semantics of particular terms, as is the case with the aspectual verbs.

In view of the phenomena discussed in connection with the review of classic GL in chapter three, we need to broaden the set of dependent types beyond the Arisotelian inspired qualia. The use of qualia led to a dilemma: the *qualia*, when given a real content, were far too restrictive a set of relations to be really useful in determining the meaning of sentences with coercions; in order to use qualia to model coercions generally, the relations associated with the *qualia* would have to be too vague to be useful. Dependent types are a flexible enough formalism to model any type of relation of any arity. What relations should we choose? Aristotelian relational categories are one sort that seem relevant as Moravcsik originally noticed. However, other categories might be relevant—we will see suitable candidates as we look at other constructions.

Dependent types are a powerful and flexible formalism, but they also introduce a danger. Type shifts involving them must be constrained and linguistically motivated if the whole system is to not collapse. If arbitrary type transformations are allowed, then compositionality becomes trivial and the type hierarchy becomes meaningless. In terms of the semantic framework that I've proposed, these coercions amount to an accommodation of the presuppositions given by typing requirements of these predicates. We will need to understand why an accommodation is possible with one predicate but not with another. At the very least, however, this makes coercion clearly a matter of the language rather than of general nonlinguistic knowledge!

8.1 Dependent Types in TCL

Dependent types within TCL are introduced during type adjustments that are present in coercion. The Head Typing Principle says that the verb's typing of its argument places should win out over the typing of its object argument if there is a coercion process. Thus if a verb requires an event as its object argument and the object argument that it requires is not an event but, say, a physical object, then either the composition process must crash or some coercion process must occur that converts the type of the object argument into some associated eventuality.

Since the dependent type often gives us an unspecified value, I will use *underspecified* types like ϵ, which is a dependent type that yields an unspecified eventuality type when applied to its type arguments or parameters.[5] That is, $\epsilon(a_1, \dots a_n)$

[5] As such ϵ conveys a bit more information than ? which we used in the analysis of relative

is an unspecified eventuality type. In cases where the conventional types of arguments do not specify the coerced underspecified types, discourse context may do so. It is up to a theory of discourse structure and interpretation to specify how this is to be accomplished. If the eventuality type remains unspecified, then the predication is not entirely well-formed. It is similar to the effect of an unresolvable anaphor such as the pronoun in (8.8) when uttered in an out of the blue context without any salient antecedent:

(8.8) He's silly.

In TCL dependent types are formed according to the following recursive definition

(8.9) DEPENDENT TYPES: If $\mu_1, \ldots \mu_n$ are types or parameters and θ is a dependent type, $\theta(\mu_1, \ldots, \mu_n)$ is a type.

Extending our notion of extended greatest lower bound to dependent types, we have:

- $\theta_1(\beta_1, \ldots \beta_n) \sqcap^* \ \theta_2(\delta_1 \ldots \delta_n) \ = \ (\theta_1 \sqcap^* \theta_2)((\beta_1 \sqcap^* \delta_1, \ldots \beta_n \sqcap^* \delta_n)$

This leads to some straightforward observations about subtypes with dependent types:

Lemma 6 *Subtypes with dependent types:*

a

$$\frac{\theta_1 \sqsubseteq \theta_2}{\theta_1(\beta_1, \ldots \beta_n) \sqsubseteq \theta_2(\beta_1, \ldots \beta_n)}$$

b

$$\frac{\beta_i \sqsubseteq \delta_i}{\theta_1(\beta_1, \ldots \beta_n) \sqsubseteq \theta_2(\delta_1 \ldots \delta_n)}$$

A dependent type is a structured object whose inhabitants are some type of entity that involves other objects as "actants"—typically either a type of eventuality or proposition. But in principle a dependent type can be any function from entities to other entities.

Dependent types greatly expand the expressive power of our type system, enabling us to define specific types of eventualities or structured entities like propositions. For instance, they allow us to pick out not only the event type of smoking,

predication and which is completely unspecified.

say, but also the event type of an agent smoking a cigarette. As another example, consider the class of transitive verbs. We may specify a general recipe for computing the thought that an intransitive verb gives rise to. Such a recipe is given in (8.10a); the type of an intransitive verb is a function from a type α (which stands for the subject of the intransitive verb) and an eventuality type, which I will abbreviate as EVTY, to a dependent type IV with two parameters. The parameters of IV are just the arguments of the verb. This is a subtype of how we have considered intransitive verbs up to now $\alpha \Rightarrow (\textsc{evty} \Rightarrow \textsc{t})$. It corresponds to the standard lambda term in (8.10b):

(8.10) a. $\alpha \Rightarrow (\textsc{evty} \Rightarrow \textsc{iv}(\textsc{evty}, \alpha)))$

b. $\lambda x \lambda e IV'(e, x)$

Note that $IV(\textsc{evty}, \alpha)$ is a subtype of t, the type of propositions.

As a first application of dependent types, let us look at how they account for the data that motivate qualia. The data motivating qualia show that there are various activities conventionally associated with different types of objects. We model these as dependent eventuality types one of whose parameters involves the type of the object with which the activity is associated. Dependent types β of α in principle can be any sort of parameterized type one of whose arguments involves α. That is, dependent types β of α always involve, in our technical sense, α. The notion of a dependent type is thus much more general than that of qualia. We will be able to model the context sensitivity of coercion using dependent types—something we cannot do with qualia—because in principle all of the parameters of dependent type can affect its value.

In investigating coercion informally, we have already seen that predications involving coercion are different from predications involving aspect selection, which we have investigated in the previous two chapters. Rather than the $\bullet$ exploitation rule, the rules for coercion resemble the lexical $\bullet$ introduction rule, a rule restricted to particular lexical items with a limited effect on the nature of the entities involved in the predication—i.e. coercion rules involve a ”local” type shift. Type shifts induced by aspectual verbs and the like from objects to eventualities, like those induced by lexical $\bullet$ introduction, do not give rise to versions of the quantificational puzzle. They do not affect the reading of the syntactic argument that is not the head, something we observed in chapters three and four by looking at pairs like the following.

(8.11) George enjoyed many books last weekend.

(8.12) Fred stole every book from the library.

(8.11) means that George enjoyed doing something with many books last weekend, presumably reading them. The coercion does not affect the quantification in the DP but only the way the variable introduced by the DP is related to the argument of the verb *enjoy*. In (8.12) the predication that forces the type shift on *book* from P • I to P affects the interpretation of the entire DP. Intuitively, the type shift in (8.11) is much less intimately connected to the meaning of the DP. Accordingly, the transfer principle for event coercion must differ from those for • exploitation. It should not shift the quantificational domain or the change the objects the DP is in some sense intuitively about.

Another indication that coercions are similar to the • introduction rule comes from the sensitivity to this principle to actual lexical items and not to general ontological categories and principles. Aspectual verbs like *stop* and *finish* have a clearly related meaning (though there are aspectual differences), but their behavior in coercion contexts is quite different.

(8.13) a. Mary finished eating the apple.

b. Mary finished the apple.

c. Mary stopped eating the apple.

d. Mary stopped the apple.

e. Jules finished smoking the cigarette

f. Jules finished the cigarette.

g. Jules stopped smoking the cigarette

h. Jules stopped the cigarette.

Whereas (8.13a,b) have interpretations in common (in (8.13b) Mary could have finished doing other things to the apple), (8.13c,d) do not. (8.13d) only has a reading that (8.13c) lacks, in which Mary stops some physical motion of the apple, like its rolling off the table. (8.13a,c) have similar meanings, although (8.13c) does not entail that she ate the whole thing whereas (8.13a) does. (8.13b,d) have quite dissimilar meanings: *finish the apple* cannot mean stopping some physical motion of the apple. Similar observations hold of the pairs (8.13e,g) and (8.13f,h). The aspectual differences between these verbs can account for the differences between the similar meanings of (8.13a) and (8.13c), but no explanation using event aspect for the differences between (8.13b,d) on the one hand the between the pairs (8.13a,c) and (8.13b,d) themselves on the other seems forthcoming. [6]

There are similar though less striking differences between *start* and *begin*. Further, the coercion effects of these verbs are less easily interpretable with many consummable direct objects than *finish*.

(8.14) a. Mary started eating her apple.
b. Mary started her apple.
c. Mary started eating her sandwich.
d. Mary started her sandwich.
e. Mary began eating the apple.
f. ?Mary began the apple.

These examples sound more or less good depending on their arguments. I have a harder time getting the eating event associated with fruits with these verbs than with constructed foods like sandwiches or candy.[7] When *start* or *begin* take natural foodstuffs as their direct objects, I do not find that the coercions pick up an eating event, even when the example is primed in the context.[8] While *begin, start* and *finish* all coerce some arguments —e.g., *start a cigarette, begin a cigarette, finish a cigarette*, they are not all equally happy in their coercive capacity.

Finally yet another aspectual verb *end* hardly seems to induce event coercion at all.

[6] As far as I know no one has looked at these minimal pairs.

[7] I didn't find any citations on *Google* for "starting apples," starting peaches or the like. And *begin the fruit* sounds even weirder to me. I did find one citation involving starting a peach candy. With vegetables like *carrot*, there are many citations involving products derived from carrots like *carrot juice*, but no citations for *started a carrot* by itself.

[8] This generalization would seem to explain Vespoor (1996)'s observations below.

(8.15) a. Yesterday Last night, my goat went crazy and ate everything in the house.
b. At 10 pm, he started in on your book.
b". At 10 pm, he began to eat your book.
b". #At 10 pm he began your book.

(3.50b") is plainly bad even though we are primed in the context to understand that the eventuality to be coerced to is an eating event. Vespoor (1996) postulates that certain identities of the sort I discuss below get to be part of the semantic content, even though they are defeasible at the outset. Technically this would involve a certain prioritization to the defeasible type specifications I introduce below. If the generalization observed above is correct, we would not need to resort to this complication.

(8.16) a. ?Mary ended the cigarette.

b. ??Julie ended the apple.[9]

c. Alexis ended the sonata with a flourish.

d. Lizzy ended the sonnet with a wonderful image.

e. Mary ended the meeting (the discussion, the negotiations).

When we use the verb *finish* with a consummable, we can specify its eventuality to one of consumption, whereas we cannot be assured of doing so when the coercing predicate involved is *end*. The coercive capacities of *end* are quite restricted, in fact to things that are informational artifacts—i.e. things of type $\text{P} \bullet \text{I}$ or entities that are or have eventualities of some sort as aspects (e.g., things like films). Event coercion is very much a matter of the verb's fine grained meaning, in particular its presuppositions. The qualia of GL associated with a food like apple have little to do with the coercions involved except at a quite superficial level.

In investigating transfer rules for lexical $\bullet$ introduction, I proposed that in keeping with the Head Typing Principle, we shift the meaning of the entire argument of the predicate responsible for the $\bullet$ typing.[10] Event coercions introduce a complication, however, because they support copredications.

(8.17) John bought and then enjoyed an apple

It seems to me that 8.17) can mean that John bought a physical apple and then enjoyed eating it. As we shall see, we can use the method of transfer that we had with $\bullet$ introduction and still get the appropriate copredications to work out. Crucially, what allows the copredication to work with coercions is that an additional variable is introduced for the eventuality of eating the apple that is linked in the logical form after Transfer to the variable introduced by the DP *an apple*. My analysis of coercion thus makes use of the Separate Term Axiom and shares this feature with my analysis of aspect selection. In general it is the use of separate terms related together that distinguishes logical polysemy from accidental polysemy. This in fact predicts that copredications should work with all cases of logical polysemy.

[9] There seem to be no sensible uses of this on Google, though someone might attempt to convey the proposition that Mary killed the apple by using this sentence. There seems to be an alternative to this aspectual use of *end* for all sorts of objects that can be destroyed, which is *put an end to*. This might be responsible for blocking the coerced reading of (8.16a,b).

[10] The other was in effect to shift the meaning of the predicate responsible for the $\bullet$ typing.

Event Coercion, like the $\bullet$ type rules, adds a variable with the dependent type that we will take account of in Transfer. Unlike the $\bullet$ rules, however, dependent types $\epsilon(\alpha_1, \ldots, \alpha_n)$ introduce parameters and we will have to introduce variables associated with those parameters. These variables will be linked in Transfer to other arguments of the term t.

(8.18) **Event Coercion (EC) for a particular term t that has the same presuppositions as *finish* or *enjoy* :**

$$\frac{\Delta, C \vdash t[t']\colon \bot,\ C \vdash v\colon \alpha,\ \Delta \vdash v \in t',\ \Delta \vdash \alpha \sqcap \textsc{eventuality} = \bot}{\Delta, C + \langle z\colon \epsilon(\alpha_1, \ldots \alpha_n, \alpha),\ z_1\colon \alpha_1, \ldots z_n\colon \alpha_n\rangle \vdash t[t']}$$

Dependent types are modelled as morphisms from entities to eventualities involving them and other entities. The type ϵ is underspecified by EC and Transfer . It links the type α of an object together with a type of eventuality that is conventionally or contextually associated with objects of type α and the other parameters. These parameters are either determined by the main verb giving the coercion or by other elements in context. EC is very general and allows in principle many other parameters to be involved in a type shift. For the purpose of modelling coercion with the aspectual verbs, howeveronly two parameters need to be introduced; these coercion cases all involve predicates of one or two arguments. One must be fixed to the type of the syntactically specified argument, because in a dependent parameterized type, one of the parameters must be the type that is ”coerced”. The other one, α_1 is the agent or syntactically determined subject of the underspecified eventuality ϵ. These types affect the transfer of information from the type shift to logical form.

From EC we now pass to an appropriate form of Transfer. As with $\bullet$ exploitation and $\bullet$ introduction, we must integrate these new variables into logical form. Correlated with a completely specified dependent type $\delta(\alpha_1, \alpha_2)$ is a formula with three variables $\phi_{\delta(\alpha_1,\alpha_2)}(x_0, x_1, x_2)$ which describes the inhabitants of δ in the object language. In such a formula $x_0\colon \delta(\alpha_1, \alpha_2)$, $x_1\colon \alpha_1$, and $x_2\colon \alpha_2$). Transfer for EC will introduce such a formula into logical form, which will provide the appropriate relation between the variable introduced introduced by EC and the variable contributed by the original term whose type forced the type coercion via EC. For EC the corresponding formula is $\phi_{\epsilon(\alpha_1,\alpha_2)}$. Now since ϵ as a dependent type has in our case two parameters, the formula ϕ_ϵ has arguments corresponding to those two parameters, plus one for the eventuality which carries the dependent type. I will suppose that the variable introduced for α_1, the type that is not the type of the coerced term, is existentially bound. As to the other variables, it will depend on the type of the coerced term whether they are lambda or existentially bound; the functor introduced by Transfer, here as with predication involving aspect selection, must combine with the logical form of the terms involved in the coercion so as to make the predication succeed.

Given our earlier observations, the material introduced in transfer has to go within the context of the local predication which forces the coercion, as in our rule of • introduction. So, for example, if we have a transitive verb *enjoy* applying to a DP like *many books* as in (8.11) repeated below,

(8.19) 8.11] George enjoyed many books

we need to apply the functor in (8.20) to the DP's contribution to logical form given in (8.21) in order to transfer the effects of the type coercion:

(8.20) $\lambda\mathcal{P}\lambda Q[\mathcal{P}[\lambda v[\exists \mathrm{z}\exists \mathrm{z}_1(Q(\mathrm{z}) \wedge \phi_{\epsilon(\mathrm{AGENT},\alpha)}(\mathrm{z}, \mathrm{z}_1, v) \wedge \mathrm{z}_1 = \mathrm{Ag}(\mathrm{z}))]]]$
$\langle \mathrm{z}\colon \epsilon(\textsc{agent}, \alpha),\ \mathrm{z}_1\colon \textsc{agent}, v\colon \alpha\rangle$

(8.21) $\lambda P\mathrm{Many}(x)\ (\mathrm{book}(x),\ P(x)),\ \langle x\colon \textsc{p}\rangle$[11]

Combining this functor with the DP's contribution to logical form yields the following:

(8.22) $\lambda Q(\mathrm{Many}(x)\ (\mathrm{book}(x),\ \exists \mathrm{z}\exists \mathrm{z}_1(Q(\mathrm{z}) \wedge \phi_{\epsilon(\alpha_1,\alpha)}(\mathrm{z}, \mathrm{z}_1, x) \wedge \mathrm{z}_1 = \mathrm{Ag}(\mathrm{z}))),$
$\langle \mathrm{z}\colon \epsilon(\alpha_1, \textsc{p} \bullet \textsc{i}),\ \mathrm{z}_1\colon \textsc{agent},\ x\colon \textsc{p}\rangle$

Using Accommodation, we can apply (8.11) to the representation of *enjoy*'s meaning in (8.23) to get (8.22) as a representation of the logical form for the entire sentence in (8.11). Notice that I've specified *enjoy* as a control verb. The aspectual verbs also function in this way.

(8.23) $\lambda\mathcal{P}\lambda e\lambda u[\mathcal{P}(\lambda v(\ \mathrm{enjoy}(e, u, v) \wedge \mathrm{Ag}(v) = u))],\ \langle e, v\colon \textsc{evty}, u\colon \textsc{agent}\rangle$

(8.24) $\mathrm{Many}(x)(\ \mathrm{book}(x),\ \exists e'\exists \mathrm{z}\exists \mathrm{z}_1(\ \mathrm{enjoy}(e', g, z) \wedge \mathrm{Ag}(z) = g \wedge \phi_\epsilon(\mathrm{z}, \mathrm{z}_1, x) \wedge$
$Ag(\mathrm{z}) = \mathrm{z}_1)]$

The functor in (8.20 suffices to handle all of the event coercion verbs. We can use it as the main case of our transfer rule:

Transfer for Event Coercion triggered by a term with the same presuppositions as *enjoy* or *finish* over a DP argument:

[11] I will ignore for the time being the complexities of plurality.

- Suppose $\Delta, C \vdash t[t'] \mapsto \Delta, C + \langle ; z \colon \epsilon(\alpha_1, \ldots \alpha_n, \alpha)\ z_1 \colon \alpha_1, \ldots z_n \colon \alpha_n \rangle \vdash t[t']$

- Then: $\Delta \vdash \lambda\mathcal{P}\lambda Q[\mathcal{P}[\lambda v[\exists z \exists z_1 (Q(z) \wedge \phi_{\epsilon(\mathrm{AGENT},\alpha)}(z, z_1, v) \wedge z_1 = \mathrm{Ag}(z))]]][t'],$
$\langle z \colon \epsilon(\alpha_1, \alpha),\ z_1 \colon \textsc{agent}, v \colon \alpha \rangle$

The use of EC and the transfer rule predicts that both the original object and the term for associated eventuality are both available as anaphoric antecedents, something which was difficult within the feature structure approaches associated with classic GL. Like our other transfer functions, Transfer for Event Coercion reflects the relations between types that underlie the semantics of the type shifting rules. The dependent types in event coercion describe a morphism from types of objects to dependent types of eventualities involving those objects, and Transfer for Event Coercion reflects that morphism from objects to eventualities in logical form. Why should this transfer principle and type shift from objects to eventualities be sound? The answer has to do with the presuppositions of the aspectual verbs and *enjoy*. Enjoying a thing, for instance, presupposes having interacted in some way with the object, and that interaction is an event. Now *enjoy* doesn't specify what that event is. The event could be just looking at the object as in *enjoy the garden* or perhaps some other activity. Similarly, one can't finish an object unless one is involved in some activity with that object, whether it be creating it or engaging in some other activity towards it. That is why such transformations are lexically based; it is the lexical semantics of the words that license the coercion and that makes the rules sound.

In many of these examples, the dependent type ϵ introduced by EC is underspecified but sometimes it is not. How should we think of the specification process of ϵ? GL is right that the types of the objects involved in ϵ are important in specifying the eventuality involved in the coerced predication. The qualia of classic GL are, from the perspective of TCL, dependent types of one argument. They are partial functions from types to types. They classify various events as agentive, telic and so on with respect to a certain parameter type—the type with which they were associated in GL.[12] To capture the intuitive aspects of GL's story about coercion, we just need to stipulate the following components to our type hierarchy:

[12] Dependent types are a natural generalization of the qualia features of the AVMs of classic GL. AVMs themselves correspond to records, which are functions from a type to a set of records or dependent types. The full recursive power of dependent types and records is used in modelling AVMs.

(8.25) a. write$\sqsubseteq$ agentive(p $\bullet$ i)
b. roll $\sqsubseteq$ agentive(cigarette)
c. smoke$\sqsubset$ telic(cigarette)
d. read $\sqsubseteq$ telic(p $\bullet$ i)
e. drink $\sqsubset$ telic(beverage)
f. play, listen-to $\sqsubseteq$ telic(music)

In reviewing criticisms of GL, however, we saw that the use of qualia, even when restricted to artifacts, didn't cut the coercion data at the joints.[13] The intuitive examples in which telic and agentive seem to function as classic GL predicts could be captured by a different classification—say, one that used modal types like consummable together with a type like artifact, where that is understood as using. To enjoy or to begin/finish a consummable artifact (e.g., a prepared food or beverage but not limited to such) almost always has the reading that one enjoyed, began or finished consuming the consummable. One could then specify the particular type of event of consumption based on the particular type of consummable. We can imagine other general types like container, containable, and dependent types like contain, where we could specify things like bottle, can $\sqsubseteq$ contain(liquid) to capture the "figure/ground" alternations that have been noticed in the linguistic literature by many. It is quite unclear, however, how one should pick one's dependent types and whether there is linguistic data to motivate a single set of dependent types.

More importantly, as noted in Lascarides and Copestake, (1995) world knowledge or discourse information can help specify underspecified types like ϵ. We have also seen that such information, as well as information about other parameters in ϵ, may defeat default lexical specifications for ϵ like those postulated for the qualia in classic GL. How should defeasibility enter into our type inferences that derive from this sort of coercion? We could assume a notion of defeasible inheritance in which subtypes inherit associating types from their supertypes but in which more specific product types will override more general associating types inherited from more general types. As in Lascarides et al. (1996), we could then account for blocking by having a default of a more specific type takes priority over a default of a less specific type.[14]

[13] For a similar criticism of classic GL, see Vespoor (1996).

[14] in earlier versions of TCL Asher and Pustejovsky (2004), defeasibility was part of the type coercion rules themselves.

Rather than make our type shifting rules defeasible, however, we can defeasibly specify in a background logic that would be part of the lexicon values for the partial function ϵ. Separating the reasoning about the values of dependent types from the type composition process allows the type coercion rules to be monotonic. We do not have to incorporate defeasibility into any of the type coercion rules. The TCL coercion rules themselves don't completely specify what type $\epsilon(a1, a2)$ is. This is done rather by the defeasible type specification rules that give appropriate defeasible values for ϵ in certain discourse contexts and when the parameters of the function are specified. This allows us to use information from the discourse already present in a limited way to infer information about the type of the associated eventuality.

Our logic here for specifying types is a universal fragment of a modal logic with a weak conditional operator > which contains variables over types, constants for atomic types, function symbols for dependent types and $\bullet$ types, and non logical predicate symbols like $\sqsubseteq$, $\sqcap$. Our axioms can all be written out in a universal fragment of first order logic, and which is thus amenable to quantifier elimination and a proof that its validity predicate is decidable.[15] The semantics, proof theory, and licensing of defeasible inferences of such a fragment is well understood (Asher and Morreau 1991, Morreau 1992, Morreau 1997, Asher and Mao 2001). $A > B$ is true roughly just in case if A then normally B. While Modus Ponens is not a valid rule for >, the semantics for > allows us to defeasibly infer B from $A > B$ and A. The logic also supports a form of specificity reasoning. I.e., conditionals with logically more specific antecedents "win" over those with less specific antecedents. So the following inference pattern is defeasibly valid, $(A \wedge B) > C, A > \neg C, A \wedge B \mathrel{|\!\!\sim} C$, where $\mathrel{|\!\!\sim}$ represents the defeasible inference relation. This is essentially the same logic used by Asher and Lascarides (2003) in reasoning about discourse structure, though applied to a different language. It is easily extended to incorporate information about discourse structure into it.

Here are some sample default identities that we might want to adopt. Many more exist that depend on suitable discourse information.

- $(a1 \sqsubseteq \textsc{human} \wedge a2 \sqsubseteq \textsc{p} \bullet \textsc{i}) > \epsilon(a1, a2) = \textsc{read}(a1, a2)$
- $(a1 \sqsubseteq \textsc{author} \wedge a2 \sqsubseteq \textsc{p} \bullet \textsc{i}) > \epsilon(a1, a2) = \textsc{write}(a1, a2)$
- $(\textsc{a}1 \sqsubseteq \textsc{goat} \wedge a2 \sqsubseteq \textsc{p} \bullet \textsc{i}) > \epsilon(a1, a2) = \textsc{eat}(a1, a2)$
- $(a1 \sqsubseteq \textsc{janitor} \wedge a2 \sqsubseteq p) > \epsilon(a1, a2) = \textsc{clean}(a1, a2)$

[15] See Lascarides and Asher 1993 for a discussion.

We can also simply deny certain equalities as in

- $(a1 \sqsubseteq \textsc{goat} \wedge a2 \sqsubseteq \textsc{p} \bullet \textsc{i})) > \neg\epsilon(a1, a2) := \textsc{eat}(a1, a2)$

Given the argument of chapter 1, the default rules are part of the lexicon, part of word meaning and the theory of predication.[16] It will undoubtedly be a tricky business to write these default specifications. But these could be gleaned perhaps from data about word collocations across corpora.

We can use the the > logic also to define properties of types in the type system. In particular, telic and agentive qualia are not defined for natural kinds, but they are for artifacts, and necessarily so. In view of the nature of artifacts, we can say that the arrow from artifacts to eventualities involving them (like the event of their being made) is guaranteed to exist. Note, however, that being defined does not necessarily imply that they are specified! Qualia are thus instances of dependent types that are partial functions. If we add to our logic predicates ↑ and ↓ to the > logic, where ↑ means that it's argument is undefined and ↓ means that its argument is defined, then we can specify formally the properties of artifacts and natural kinds as follows:

- $\top >\uparrow (\textsc{telic}(\textsc{nat kind}))$
- $\top >\uparrow (\textsc{agentive}(\textsc{nat kind}))$
- $\downarrow (\textsc{telic}(\textsc{artifact}))$
- $\downarrow (\textsc{agentive}(\textsc{artifact}))$

As the data suggests, there are many possible values for dependent types like ϵ. This suggests that ϵ may return a set of values for some second arguments that can further be specified in certain discourse situations; ϵ might even involve a non-deterministic choice element, choosing an eventuality out of a set of potential candidates with a certain frequency. The main advantage of the dependent type approach, however, is that it is allows us to specify associated eventualities by taking other arguments of the dependent type into account. This will allow for us to specify many cases of coercion that we have already examined and that fall outside the scope of traditional qualia.

To see how we get a derivation of one of the classic qualia cases, let's now consider (8.26).

[16] This makes for a much richer notion of lexical meaning than, e.g., Fodor and Lepore (1998) or Capellen and Lepore would want to countenance.

(8.26) Sheila enjoyed the book.

Let's now proceed with an analysis of (8.26).

(8.27) *enjoy*:
$\lambda\mathcal{P}\lambda u\mathcal{P}(\lambda v \text{enjoy}(u,v) \wedge Ag(v) = u), \langle \mathcal{P}\colon (\textsc{event} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, v\colon \textsc{event}, u\colon \textsc{agent}\rangle$

Putting together (8.27) and the term for *the book*, we get:

(8.27a) $\lambda\mathcal{P}\lambda u\mathcal{P}(\lambda v \text{enjoy}(u,v) \wedge Ag(v) = u)[\lambda\exists x(\text{Book}(x) \wedge P(x))]$,
$\langle \mathcal{P}\colon (\textsc{event} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, v\colon \textsc{event}, u\colon \textsc{agent}, P\colon (\textsc{p} \bullet \textsc{i} \Rightarrow \textsc{t}) \Rightarrow \textsc{t}, x\colon (\textsc{p} \bullet \textsc{i})\rangle$

There is here a type clash, between the argument for *enjoy* and the type requirement it places on that argument. *enjoy*'s presupposition induces a coercion and thus introduces new variables into the context via EC. Because of the entry for *enjoy*, we can specify α_1 to the type of the first argument of *enjoy*. The second argument of ϵ is specified to be the type of the variable introduced by the direct object, in this case *the book*. Thus, we can derive, letting C stand for the typing context in (8.27c):

(8.27b) $\lambda\mathcal{P}\lambda e\lambda u\mathcal{P}(\lambda v \text{enjoy}(e,u,v) \wedge Ag(v) = u)[\lambda P\exists x(\text{Book}(x) \wedge P(x))]$,
$C + v'\colon (\textsc{p} \bullet \textsc{i}),\ z\colon \epsilon(\textsc{agent}, \textsc{p} \bullet \textsc{i}), z_1\colon \textsc{agent}\rangle$

With the first rule of default identities, we can now specify the description of the eventuality using the information on types in (8.27c), calling the typing context in (8.27b) C':

(8.27c) $\Delta, C' + z\colon \textsc{read}(\textsc{agent}, \textsc{p} \bullet \textsc{i})\rangle \vdash \lambda\mathcal{P}\lambda e\lambda u\mathcal{P}(\lambda v \text{enjoy}(u,v) \wedge Ag(v) = u)[\lambda P\exists x(\text{Book}(x) \wedge P(x))]$,

Now we use the functor provided by the Transfer rule on *the book* and redo the applications to the point where we were before:

(8.27d) $\lambda\mathcal{P}\lambda e\lambda u\mathcal{P}(\lambda y(\text{enjoy}(e,u,y) \wedge Ag(y) = u)[\lambda Q\exists x(\text{book}(x) \wedge \exists z, z_1(Q(z) \wedge \text{read}(z, z_1, x) \wedge Ag(z) = z_1)], \langle \mathcal{P}\colon (\textsc{p} \Rightarrow \textsc{t}) \Rightarrow \textsc{t},\ v'\colon \textsc{p}, z\colon \textsc{read}(\textsc{agent}, \textsc{p} \bullet \textsc{i}, z_1, u\colon \textsc{agent}, e\colon \textsc{evty}\rangle$

Using Application, we now get

(8.27e) $\lambda u\lambda e\exists x\exists w\exists y(\text{book}(x) \wedge \text{read}(y, w, x) \wedge Ag(y) = w \wedge \text{enjoy}(e,u,y) \wedge Ag(y) = u), C'$

Simplifying by substituting equals for equals and substitution of logical equivalents, we get:

(8.27f) $\lambda u\lambda e\exists x\exists y(\text{book}(x) \wedge (\text{read}(y,u,x) \wedge \text{enjoy}(e,u,y)), C'$

Filling in the rest of the arguments in the standard fashion we we derive for *Sheila enjoyed the book*

(8.27g) $\exists e\exists x\exists y(\text{book}(x) \wedge (\text{read}(y,s,x) \wedge \text{enjoy}(e,s,y)), C'$

That is, we derive the reading that Sheila enjoyed an event that is associated with *the book*, and that event is by default an event of her reading the book. Similarly, we predict *Elizabeth enjoyed the sonata* to mean that Elizabeth enjoyed playing the sonata. This default can be overridden if we know that Elizabeth doesn't play sonatas.

The TCL account of coercing verbs predicts that copredication succeeds when using a coercive verb. Recall our earlier example,

(8.17) John bought and then enjoyed an apple

If we suppose that the two verbs coordinate together to yield with the appropriate typing contest,

(8.28) $\lambda\mathcal{P}\lambda x\mathcal{P}[\lambda y[\lambda u, v\ \text{buy}(u,v)[x][y] \wedge \lambda u', v'\ \text{enjoy}(u',v')[x][y]]]$, where $\langle x, y\colon e\rangle$

then we'll want to use Accommodation and then Application on x and then combine the DP meaning with the coordinated VP meaning first to avoid a type clash, and this will allow us to specify the type of y to P. We get then

(8.29) $\lambda x\exists w(\ \text{apple}(w) \wedge [\lambda v\ \text{buy}(x,v)[w] \wedge \lambda v'\ \text{enjoy}(x,v')[w]]]$

We can now proceed to use EC and Transfer on the second conjunct to get the appropriate reading.[17]

[17] Note that classic GL cannot handle such examples, if one is to select the qualia value as the contribution to logical form of *apple*. It will run into exactly the same difficulty that GL's account of copredication with • types did.

8.1.1 Discussion of TCL's Analysis of Coercion

According my proposal, it is the presuppositions of *enjoy* and *finish* that provide the justification for the event coercion and for the particular form of transfer. TCL predicts that event coercion is fully general in the following sense: any predicate with presuppositions those of *finish* or *enjoy* will give rise to EC. So for example, verbs like *last* that have similar presuppositions[18] will also give rise to a version of EC. EC does not determine a specific eventuality type as the value of the morphism for a given object, and thus Transfer yields an underspecified formula characterizing the eventuality it introduces in logical form. When such eventualities cannot be defined by the context or by other predicational parameters, TCL predicts that it's not clear what's being said; there is no fully formed logical form for such a sentence, much like a case of an unresolved anaphor. For example in (3.29), the predication is predicted to be odd or incomplete, much in the same way as *he's silly* uttered in a context where an antecedent for the pronoun isn't readily available expresses an incomplete proposition:

(3.29) John enjoyed the door.

The lexically restricted EC rule makes just the right predictions in this case. It does not overgenerate as classic GL did. Recall further that such cases of coercion, which do not sound natural, need only a bit of context to become good again. In the next section, we will see how discourse context can specify the underspecified types introduced by EC.

TCL also predicts that verbs with different presuppositions—among which, given our discussion above, we must include the aspectual verbs *begin, stop, start* and *end*— will not have exactly the same forms of EC or of Transfer. They will not give rise to the same readings as *enjoy* and *finish*. Consider this minimal pair involving *finish* and *stop*.[19]

(8.30) a. John has finished the garden (the kitchen).

b. #John has stopped the garden (the kitchen).

[18] *Last* presupposes that their subjects have temporally bounded states or activities—e.g., of existence or some more contextually defined state or activity.

[19] Note that even *finish* and *enjoy* don't have exactly the same EC rule at the level of types and they certainly behave differently at the level of the default identities; to enjoy a garden suggests that one enjoyed looking at a garden, whereas to have finished a garden conveys that one has finished constructing the garden. *Finish* in its transitive usage must take an accomplishment type eventuality, whereas *enjoy* need not. *Finish* also requires that its accomplishment argument be a action by the subject that directly affects the object. This is not true of *enjoy* either.

An agent's finishing *X* presupposes that the agent was involved in some sort of accomplishment involving *X*. *Finish* in its transitive use has a presupposition that its subject is performing and that directly affects the direct object and it is that eventuality that the agent finishes, whereas *stop* lacks this presupposition. When an agent stops *X* where *X* is a DP, the presupposition is just that *X* was involved in some sort of event or process that the agent puts a stop to, not that the agent puts a stop to some event that he and *X* were involved in. Moreover, the action in the case of *finish* must be of accomplishment type, whereas this is not true of *stop*.[20] Consider for instance,

(8.31) a. I've stopped smoking cigarettes.
b. I've finished smoking cigarettes.
c. #I've stopped smoking 5 cigarettes.
d. I've finished smoking 5 cigarettes.

(8.31a) implies that I've stopped doing an activity, the activity of smoking cigarettes. (8.31b) does not have this reading. Instead (8.31b) seems almost to force an accomplishment structure on the event of smoking cigarettes. The pair (8.31c,d) is also interesting: *finish* naturally combines with a gerund denoting an accomplishment, whereas *stop* does not.

In comparison to *finish* or *enjoy*, *stop* has different control properties. While *stop* does control syntactically explicit agent argument positions of complements like in *John stopped smoking* but it does not control agent argument positions of eventualities that are not syntactically expressed. We can put this point more clearly in the type theoretic framework: the presuppositions of *stop* do *not* license a dependent type of eventuality involving *X and* the subject of *stop*; they license a dependent type of eventuality only on *X*. Hence the particular transfer functor for this coercion will differ from that for *finish* or *enjoy*. For *John stopped the apple*, the form of EC appropriate to *stop* will introduce a term *z* with the type $\epsilon(\textsc{apple})$. We can say a bit more about the type of presupposition that *stop* gives rise to given the data we've surveyed. It appears that if the object argument *X* is typed PHYSICAL, then the process involving *X* must be a physical motion. This is something that we can specify using the language for type specifications that served to specify default values for the dependent type ϵ. For instance,

(8.32) $\textsc{stop}(\textsc{agent}, \epsilon(\textsc{p})) \rightarrow \epsilon(\textsc{p}) \sqsubseteq \textsc{phys-process}$

[20] Agents can of course stop doing something to *X*, in which case there is a presupposition that the agent was doing something to *X*.

Thus, we predict that it's hard to accommodate the relevant presupposition for (8.30b), because it's just so unclear what physical motion the garden or kitchen could be undergoing that John could put a stop to.

The aspectual verbs *begin* and *start* do not have the same presuppositions as *finish* or *enjoy* either. For example,

(8.33) a. John has not started (begun) making dinner.

b. Has John started (begun) making dinner?

do not entail that dinner is being made or that John is doing anything about dinner, whereas

(8.34) a. John has not enjoyed (finished, stopped) making dinner.

b. Has John finished (stopped, enjoyed) making dinner?'

all entail that there is, or was, a making of dinner. In the jargon of the presupposition literature, there is no projection of an eventuality presupposition outside the scope of the negation or question operator in (8.33, whereas there is for (8.34). Since such projections are a standard test for the presence of presuppositions, it would appear that *begin* and *start* do not have presuppositions of the sort that *finish, stop* and *enjoy* do. They do have some sort of presupposition, though it is quite a weak one. (8.33) presuppose at least that its author believes that John should or will make dinner. And this presupposition holds up for nonsense words as well. (8.35) implies John will do or should be doing or have done something to the zibzab.

(8.35) John has not started (begun) a zibzab.

begin and *start* have strong presuppositions about the type of arguments they permit. They take some physical objects, typically, constructed or fabricated consummables—-sauces, pastas, cigarettes, and for *start* anything with an engine it (motocycle, bus, airplane, car, etc.) *Begin* and *start* are quite restrictive with respect to non event and non physical arguments. They share this behavior concerning arguments of abstract object type with other eventive verbs like *last*, but not, curiously enough, with *enjoy*.

(8.36) #John has started/begun the fact (the thought, the proposition) that he is hungry/the number 2, the cumulative hierarchy of sets.

(8.37) John has started/begun thinking about the fact that he is hungry/the number two/the cumulative hierarchy.

(8.38) #The proposition/fact that 2=2 will last for all eternity.

(8.39) John enjoys the cumulative hierarchy/the number π.

(8.40) John enjoys the fact/ the thought that you are unhappy.

It's of course weird to say that John enjoys the proposition that you are unhappy, though presumably not that John enjoys entertaining the proposition that you are unhappy. This points to a really finegrained use of the defaults in the TCL system that goes way beyond any of the generalizations given by the qualia. It would appear that it's just a linguistic fact that for certain abstract object type arguments, the dependent type $\epsilon(\textsc{a}_1, \textsc{prop})$ introduced by coercion is not defined. On the other hand, there are many more undefined operations for the true aspectual verbs. More formally, we can add to our axioms about types:

- $\uparrow \epsilon(\textsc{a}_1, \textsc{prop})$
- $\uparrow \textsc{begin}(\epsilon(\textsc{a}_1, \textsc{i})$[21]
- $\top >\uparrow \textsc{start}(\epsilon(\textsc{i})$, but $\downarrow \textsc{start}(\epsilon(\textsc{rumor})$

An important fact about such weak presuppositions is that they can often "disappear" (technically they don't disappear but are rather accommodated within the scope of the relevant operators). It seems that the presupposition disappears when we embed the aspectual verb under a future progressive or under a question or negation operator:

(8.43) Are they going to start a new album?[22]

Such weak presuppositions would lead us to expect the coercion effects which depend on them to be spottier as well.

[21] A potential counterexample is *story*. Some have claimed this is an abstract object. However, there are many examples attested on *Google* of stories lasting. *Last* typically holds of events or states:

(8.41) Maybe that wasn't the rustle of pages you heard while this story lasted, but Peter Pan himself, listening in.

(8.42) Although the story lasted until the very last issue of *Monster Fun*, it did not make ... The story lasted for just 19 issues...

These examples would suggest that *story* is perhaps of a complex type—$\textsc{evty} \bullet \textsc{i}$ or in other uses $\textsc{p} \bullet \textsc{i}$

[22] Thanks to Julie Hunter for this example.

Recall in the discussion of classic GL, we saw that that the eventuality introduced by coercion is sometimes not available for anaphoric reference.

(8.44) Paul was starting a new book. That lasted for 3 days.

The TCL approach to coercion using dependent types makes these eventualities available for coreference, when eventualities are the appropriate type of object selected for by the coercing predicate.[23] The reason why some examples sound bad has to do with the intensional use of certain aspectual verbs like *start* and with the fragile nature of their presuppositions. *start* is like the imperfective mood; one can start something without finishing it. If these intensional uses involve event types and not events as arguments, this suffices to explain the awkwardness of anaphoric references to events with such coercion predicates. The aspectual verbs with the stronger presuppositions do support event anaphora.

(8.45) a. John has finished a new book. That took three years out of his life.

b. John has finished a new book. The writing took three years, and then the efforts to find a publisher three more.

I find both of these examples fine, when the pronoun in (8.45a) refers to the creation or writing of the book.

The anaphoric data with *enjoy* remains more problematic. It appears difficult to pick up an event with an anaphoric pronoun even when we put in a lot of discourse context to help with the anaphor, though individual judgements are quite variable on this score. Most native speakers, however, find (8.46b) better than (8.46a).

(8.46) a. ?James is enjoying a cigarette. It started just now.

b. James is enjoying the smoking of a cigarette. It started just now.

c. ? James enjoyed the book. But it only lasted a couple of days. So he needed to get some other mysteries to entertain himself after work during the Summer School.

d. James enjoyed the book. His reading of it lasted a week.

Sensitivity to types of eventualities may affect coercion in other ways too. Consider the discussion of *want* by Fodor and Lepore (1998). They argue that *want* as in

[23] I noted earlier that the problems with the anaphora test seem limited to the class of verbs that take event types as arguments.

(8.47) want a beer

in effect coerces its arguments into something of the type denoted by inifinitival phrases. But they don't make use of underspecification; instead they make use of the light verb *have*. Thus (8.47 should mean *want to have a beer*. The problem with their proposal as pointed out by Harley (2006)? is that even the light verb *have* has some restrictions that make this proposal not work. She considers,

(8.48) a. John wants a compliment.

b. John wants a pat on the back.

which are perfectly fine but their synonyms according to the Fodor and Lepore strategy

(8.49) a. #John wants to have a compliment.

b. # John wants to have a pat on the back.

aren't particularly good. One should say *John wants to get a complement*, *receive a compliment*, *get a pat on the back*, and so on. On the present proposal *want* might induce like the aspectual verbs or *enjoy* a dependent type shift on its object argument that is underspecified by the verb. Like the simple coercion cases used to motivate qualia, the direct object of *want* with the *want DP* construction can specify what sort of eventuality the dependent type should give.[24]

If coercion is relatively straightforward in object position for the TCL approach, there remains a puzzle about the lack of coercion in subject position. Why are (8.50b,c) so bad?

(8.50) a. The reading of the book /book reading started (began, ended) at 10 am.

b. #The book started (began, ended) at 10 am.

c. The eating of the pasta started (began, ended) at 6 pm.

d. ?The pasta started (began, ended) at 6 pm.

e. The cooking/preparation of the pasta started (began, ended) at 6 pm.

[24] Harley argues that lexical decomposition is necessary to treat these examples. But this does not seem to be the case, if we use underspecification and dependent types.

Currently, EC does not extend to these uses of the aspectual verbs. However, these constructions require some comment in virtue of the following, perfectly acceptable examples:

(8.51) a. The pasta started to boil

b. The pasta started to sizzle/sizzling.

These point to a complex compositional problem; what these examples mean is that the pasta's boiling or sizzling started. The TCL system is not designed to handle such constructions, though they are an interesting object of study. Given this observation concerning (8.51), we see an important property of coercions. A term can trigger a coercion with respect to some argument, i.e. a modification of logical form in TCL, only if the argument term is within its scope. So verbs can't introduce coercions in virtue of a type clash with their subjects, as they are external arguments. On the other hand, as we've seen, subjects can at least affect the content of a verbal predicate coercion; since subjects take verbs and verb phrases in their scope in TCL, this is more positive evidence that coercion obeys a scope like principle. This is what we should expect if it is indeed a ”local” type of operation.

8.2 Discourse and Typing

With the basic coercion story in place, it's time to now integrate context dependence at the discourse level. We have seen how to integrate information relevant to the coercion of an argument from other arguments of a functor using the > logic. Now we turn to the problem of integrating factors from the larger discourse.

There is a lot of evidence that typing is dynamic and depends at least in part on context. The eventuality involved in enjoying a book will depend on discourse context. It is not wholly lexically governed. Within a fairy book context, goats can enjoy reading books as opposed to eating them. Stones can talk. Standard predicational restrictions can be relaxed or wholly rewritten. How do we do this? Lascarides and Copestake offer one approach with the ”discourse wins” rule. But we need to place this kind of operation squarely within the lexicon. In order to do this we need to take a detour through a theory of discourse structure. That is what I propose to do now.

I will choose SDRT, a theory that offers a formal account of the hypothesis that discourse has a hierarchical structure upon which interpretation depends. For our purposes I will need the following features of SDRT.[25]

- SDRT's semantic representations or logical forms for discourse, SDRSs, are recursive structures. A basic SDRS is a labelled logical form for a clause, and a complex SDRS will involve one or more discourse relation predications on labels, where each label is associated with a *constituent*, i.e., a perhaps complex SDRS.

- An SDRS for a discourse is constructed incrementally within a logic of information packaging that uses several information sources and that is responsible for the final form of the SDRS. The logic of information packaging, which reasons about the structure of SDRSs, is distinct from the logic of information content, in which we formulate the semantic consequences of an SDRS.

- The rules for inferring discourse relations are typically rules that exploit a weak conditional $>$. They form part of the *Glue Logic* in SDRT, which allows us to ”glue” new discourse segments together with discourse relations to elements in the given discourse context. This logic has exactly the same rules as the logic for specifying values for dependent types, though the language of types and the language for describing discourse logical forms are distinct.

- The discourse relations used in SDRT, which have definite semantic (e.g. spatio-temporal, causal, etc.) effects, are binary and either *coordinating* (Coord) or *subordinating* (Subord). Examples of subordinating relations are *Elaboration*, where the second constituent describes in more detail some aspect of some eventuality or some fact described in the first constituent. Some coordinating relations like *Narration* (where constituents describe a sequence of events) and *Continuation* (where linked constituents elaborate simply on some topic) require a topic; i.e., there must be a simple, constituent, a common “topic”, that summarizes the two related constituents and that is linked to them via the subordinating *Elaboration* relation. If this third constituent has not been explicitly given in the previous discourse, it must be “constructed”.

Discourse structure affects the way underpsecified elements of semantics can be resolved Sometimes the temporal structure of a discourse is more elaborate

[25] For details,see, e.g., Asher (1993), Asher and Lascarides (2003).

than what is suggested by a semantic analysis of tenses (such as that found in DRT (Kamp and Reyle 1993). There are clearly temporal shifts that show that the treatment of tenses cannot simply rely on the superficial order of the sentences in the text. Consider the following discourse (from Lascarides and Asher 1993).

(8.52) a. (π_1) John had a great evening last night.

b. (π_2) He had a great meal.

c. (π_3) He ate salmon.

d. (π_4) He devoured lots of cheese.

e. (π_5) He then won a dancing competition.

(8.52c-d) provides ‘more detail’ about the event in (8.52b), which itself elaborates on (8.52a). (8.52e) continues the elaboration of John’s evening that (8.52b) started, forming a *narrative* with it (temporal progression). Clearly, the ordering of events does not follow the order of sentences, but rather obeys the constraints imposed by discourse structure, as shown graphically below. Thus the eventualities that are understood as elaborating on others are temporally subordinate to them, and those events that represent narrative continuity are understood as following each other. The relevant parameter for interpreting tenses is *discourse adjacency in the discourse structure*, not superficial adjacency. A theory like SDRT (Asher (1993), Asher and Lascarides (2003) provides the following discourse structure for (8.52) and this allows us to get a proper treatment of the tenses therein. Here π_6 and π_7 are discourse constituents created by the process of inferring the discourse structure.[26] Note that π_1 and π_2 serve as topics for the Narrations holding between π_2 and π_5 and π_3 and π_4.

Temporal relations between events introduced by verbs with certain tenses are underspecified in a language like English, and discourse structure is an important clue to resolving this underspecification. SDRT predicts that discourse structure affects many types of semantic underspecification. Nearly two decades of work on ellipsis, pronominal anaphora, and presupposition has provided evidence that this prediction is correct (Asher 1993, Hardt, Busquets and Asher 2001, Asher and Lascarides (1998, 2003)). My hypothesis here is that discourse structure also helps resolves underspecification at the level of types and hence contributes to content in predication.

To see how this comes about, we need to think a little harder about discourse coherence and its relation to discourse structure. In SDRT, as in most theories

[26] See Asher and Lascarides (2003) for details.

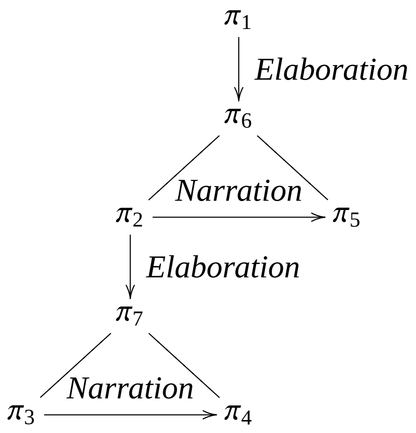


Figure 8.1: SDRT graph for (8.52)

of discourse interpretation, to say that a discourse is (minimally) coherent is to be able to derive a discourse structure for it. Discourse coherence is a scalar phenomenon, however. It can vary in quality. Following Asher and Lascarides (2003), I say that an SDRS τ_1 is more coherent than an SDRS τ_2 if τ_1 is like τ_2, save that τ_1 features strictly more rhetorical connections. Similarly, τ_1 is more coherent than τ_2 if $\tau_2 \rightsquigarrow \tau_1$ (i.e., τ_1 is just like τ_2 save that some underspecified conditions in τ_2 are resolved in τ_1). But for now, let's focus on the perhaps simplistic position that discourse coherence is maximised by 'maximising' the rhetorical connections and minimising the number of underspecified conditions. We can define a principle that will govern decisions about where one should attach new information when there's a choice. It will also govern decisions about how other forms of underspecification get resolved. And the principle is: the preferred updated SDRS always *maximises discourse coherence* or *MDC* (Asher and Lascarides 2003).

The degree-of-coherence relation $\leq$ thus specified is a partial ordering on discourse structures: other things being equal, the discourse structures which are maximal on $\leq$ are the ones with the greatest number of rhetorical connections with the most compelling types of relation, and the fewest number of underspecifications.

MDC plays a role distinct from the axioms in the glue logic for inferring discourse relations in SDRT. MDC is a way of choosing the best among the discourse structures allowed by the axioms in the glue logic. It's something like an optimality constraint over discourse structures that are built via the glue logic axioms. Asher and Lascarides (2003) examine in detail how MDC works in picking out the intuitively correct discourse structure for (8.52), as well as many other examples. We won't be much concerned here with exactly how discourse relations are inferred, but we will need from time to time to refer back to this background logic.

To get a feel for how MDC works in tandem with underspecification, consider the example from Asher and Lascarides (2003), (8.53):

(8.53) a. I met an interesting couple yesterday.

b. He works as a lawyer for Common Cause and she is a member of Clinton's cabinet.

The pronouns *he* and *she* introduce underspecified formulas into the logical form for this discourse. They could be bound deictically to salient individuals in the context, but that would not allow us to infer a tight connection between (8.53a) and (8.53b). The discourse would lack coherence. On the other hand, if *he* and *she* are linked via a "bridging" relation to the DP *an interesting couple*, then we can infer a strong discourse connection between (8.53a) and (8.53b). MDC predicts that this anaphoric interpretation of the two pronouns is preferred because it leads to the preferred discourse structure.

8.3 Back to the Lexicon

Armed with SDRT's notion of discourse structure, we can return to the examples with the aspectual verbs. I will use the speech act discourse referents $\pi_0, \pi_1, \pi_2, \ldots$ to isolate the minimal discourse units in these examples.

(1.37) a. ??Yesterday, Sabrina began with the kitchen (π_1). She then proceeded to the living room and bedroom (π_2) and finished up with the bathroom (π_3).

b. Yesterday Sheila cleaned her house (π_0). She began with the kitchen (π_1). She then proceeded to the living room (π_2) and bedroom and finished up with the bathroom (π_3).

c. Last week Julie painted her house (π_0). She started with the kitchen (π_1). She then proceeded to the living room and bedroom (π_2) and finished up with the bathroom (π_3).

Aspectual verbs like *begin (with)*, *proceed* and *finish up* incorporate a certain bit of discourse information.[27] They are, from the perspective of SDRT, *Narration*

[27] For more on "discourse verbs", see Danlos (2006).

introducers. *Begin with*, *continue with* or *proceed with*, and *finish up with* form a natural sequence in a narrative story. The point of these verbs is to introduce narrative discourse structure into logical form, while the object arguments of these verbs should provide the events that make up the narration. Here is a possible lexical entry for *proceed* that formalizes thesepoints:

(8.54) $\textit{proceed} \mapsto \lambda\mathcal{P}\lambda\mathcal{Q}\exists\beta\exists\alpha(\alpha{:}\,\mathcal{Q}[\lambda x\mathcal{P}(\lambda y\phi_{\epsilon}(x,y))]\wedge \text{Narration}(\beta,\alpha)\wedge\beta = ?)$

The adverbial *then* can also function as a clue for Narration, something that we can easily formalize in SDRT's glue logic (Asher and Lascarides (2003)).

Using these lexical entries together with an axiom for *then*, the logical form for (1.37a-c) will each specify a discourse structure containing the information Narration(π_1, π_2) and Narration(π_2, π_3). In SDRT Narration(π_1, π_2) and Narration(π_2, π_3) imply that there is a topic discourse constituent that π_1, π_2 and π_3 elaborate on. This topic discourse constituent is explicit in (1.37bc) and it is intuitively this topic that specifies in (1.37) what the object arguments of the aspectual verbs are.

The eventuality type of the object argument of *Begin with* can sometimes be specified by the object in the PP via our default type specification rules. Consider,

(8.55) Jane began with a cigarette and then proceeded to a glass of wine.

(8.56) Jane began with a fried oyster appetizer (π_1) and then proceeded to the osso buco (π_2).

In these examples, the discourse verbs together with the discourse connector *and then* both specify that the two verb phrases must be linked via the Narration discourse relation in the glue logic axioms for SDRT. However, in (8.55) and (8.56), the resolution of the underspecified dependent types proceeds independently from the construction of the discourse structure thanks to the type specification rules in the lexicon, and in fact it is these resolutions that serve in (8.56) to construct a topic for the discourse, as required by the fact that Narration holds between (π_1) and (π_2): the topic is something like Jane's meal.

In (1.37a) we must construct a topic as in (8.55) or (8.56) that elaborates the constituent containing π_1, π_2 and π_3 linked by Narration. In order to do this, we need to look at the eventualities involved in the Narration and try to generalize from them to construct a topic. However, the relevant eventualities are those provided by EC and Transfer, and their type is underspecified—-they are of type ϵ(AGENT, KITCHEN). The problem is that ϵ doesn't return a determinate value when applied to KITCHEN and the type of AGENT, at least as far as our lexically given type specification rules are concerned. In this respect *kitchen* is different from *cigarette, novel* or words associated with • types like *lunch* (MEAL • EVENT). If this eventuality cannot be specified, then we cannot specify the topic for the Narration. The discourse is thus lacking in coherence, not only because of the underspecified eventualities involved in the sequence but also because of the lack of topic.

On the other hand in (1.37bc), by linking the underspecified conditions describing the eventualities in the logical forms for the clauses π_1, π_2, and π_3 with the condition describing the eventuality in π_0, we can easily infer Elaboration between π_0 and the constituent containing π_1, π_2, and π_3, thus satisfying the requirements of Narration. Where there is an underspecification like this and resolving it with the eventuality in topic would allow us to link the two constituents via Elaboration, then MDC favors this resolution. As with other resolutions of under specifications in SDRT, the resolution of the under specification and the construction of the discourse structure are codependent tasks. By filling in the under specified types in a particular way, we can build the discourse structure—we can have a fully specified topic that we elaborate on with the second and third constituents, which are linked by Narration. And by building such a discourse structure we resolve the under specifications. MDC will pick the discourse structure with the underspecifications resolved as the preferred one; it has a well formed discourse structure and no under specifications.

Let's now reconsider the principle ”discourse wins” proposed by Lascarides and Copestake (1996). Lascarides and Copestake as well as Asher and Lascarides point out that the qualia based predictions of classic GL are easily overturned in suitable discourse contexts. I repeat that moral here again with an example very similar to the discourse examples above.

(8.57) a. Last week Suzie worked on a number of paintings of consummables.

b. On Monday she began a cigarette; on Tuesday she began and finished a glass of wine.

c. On Wednesday she finished the cigarette, and started on a plate of tacos.

d. Thursday she finished the plate of tacos.

e. I really enjoyed the plate of tacos.

The event readings posited by the qualia of classic GL simply aren't available in this discourse context. MDC and the process of discourse construction can overrule any of the defeasible conclusions that issue from the specification logic for underspecified dependent types. These examples are further evidence that most likely qualia are special examples of a much more general phenomenon of dependent type coercion.

8.3.1 Other Cases of Event Coercion

Noise Verbs

The aspectual verbs and *enjoy* provide one class of event coercing constructions that allow the introduction of dependent eventuality types. But there are other forms of coercion. Other verbs, for instance, can coerce eventuality readings. For instance*hear*, *deafen, drown out* and so on.

(8.58) a. We hear the piano two floors down.

b. The orchestra drowned out the piano.

c. The guitar was deafening.

d. The airplane was deafening.

We can assimilate verbs requiring ”noise” arguments to the case of *stop*. They too license a form of event coercion, but the dependent type involved in the examples in (8.58) is more specific— it must be an eventuality which makes a noise. As with the other examples of eventuality coercion, it is the verb that does the coercion. Though we may not understand exactly what's going on in (8.59) and we can't specify what eventuality is involved, we infer that some eventuality involving the stone that makes noise was deafening.

(8.59) The stone was deafening.

Once again, we have to address the implications of the TCL view on which noise making events are introduced by the transfer rules. With regard to Kleiber's (3.11), repeated below

(3.11) Nous entendimes le piano, qui nous parvenait flottant par-dessus du lac (We heard the piano which came to us floating over the waters of the lake).

we can say that while it is certainly bad, this fact is predicted by the TCL approach on which the meaning of the noun is not itself shifted. The noun, even the noun phrase *le piano* retains its normal type, and so the relative pronoun must agree in type with it. This predicts that the only possible reading of (3.11) is one on which the piano is itself floating over the waters of the lake.

On the other hand true anaphoric uses of pronouns should be good according to my analysis.

(8.60) a. We hear the piano two floors down. It starts every evening around 10 and goes on for a couple of hours.

b. The guitar was deafening. It hurt my ears.

The pronoun in both of these examples has a reading on which it picks out the noise made by the object that is the argument of the noise coercion predicate in the previous sentence. Thus, TCL seems to make the right predictions about these examples, unlike Nunberg or classic GL.

8.4 More Types and Transformations

So far we've seen examples of coercion involving events and matters related to events. But there are many other examples of coercion that our dependent types can model.

Coercions from objects to their representations

Depiction verbs also introduce dependent types and function as coercion verbs.

(8.61) John is drawing a fish.

(8.62) Suzie is painting a landscape.

(8.63) Pat sculpted a lion.

(8.64) This program models buildings in 3D.

(8.65) Chris sketched his hand.

All of these verbs coerce their object arguments into some sort of representation of the object. The Head Typing Principle predicts in all these cases that the object type must undergo coercion if possible. That is, they introduce a depedent type REPRESENTION-OF$(a1, a2)$ and the coercion rule will specify $a2$ to the original type of the object argument. This sort of depedent type introduced is exactly analogous to the coercion cases we have seen so far. But in this case we are not introducing an eventuality but another kind of entity, a representation, which is a special kind of physical object.[28]

Coercion of the depiction type also works with prepositional phrases. Consider the following pair (due to Marliese Kluck):

(8.66) a. The garden with the flowers was especially beautiful.

b. The dress with the flowers was especially beautiful.

The head noun provides a preferred interpretation of the prepositional phrase. This example is interesting because there is no type conflict between the head noun and the prepositional phrase. There is a reading on which the dress could have been grouped with real flowers, though it's not the preferred reading. There's even a representational reading of the whole noun phrase—that is the picture of the garden with the flowers of the picture of the dress with the flowers. These examples are not captured within TCL because TCL type shifts are guided by type conflicts. We could adapt the TCL system, however, by making the type shifts dependent on plausibilities. That is, if one reading with a type shift makes the discourse more coherent, then the type shift will be licensed. With this we can then use the machinery of TCL I leave the matter here, however.

Resultatives

Another example of a coercion verb is *freeze*. *freeze* requires that its theme argument be a liquid. It denotes a process whereby its theme argument, which starts out in a liquid state ends up in a no longer wholly liquid state. However, *freeze* can apply to things that are not liquids, as in (8.67c).

(8.67) a. The water froze.

b. The river froze.

c. The bottle froze.

(8.67c) is clearly a case of object coercion. Bottles are not liquids. However, what they contain is often a liquid. We typically understand (8.67c) as a coercion. The contents of the bottle froze, contents which we force to be liquid or at least partially liquid (bodies and wet laundry can also freeze solid).

[28] Classic GL, however, would have great difficulty in making these dependent types one of the qualia. So once again we see the gain in expressive power of dependent types.

(8.68) a.. The bottle froze.
[The liquid in the bottle] froze.

b.. The bottle of soda froze.
[The soda] in the bottle of soda froze.

A verb like *freeze* also introduces a dependent type—call it CONTENTS that is invoked when the direct object of *freeze* is not a liquid or is not a material that stiffens under cold and is rather a container. *drink* is another verb like freeze that invokes the dependent type CONTENTS, when its object is not a liquid:

(8.69) Nicholas drank the bottle (two bottles).

Do these coercions really introduce new objects into the discourse, as our mechanism would predict? We must see how such coercions fare with the anaphora test.

(8.70) Nicholas drank the bottle. It was a delicious Corbières red, a mixture of grenache, cabernet and morvèdre.

(8.71) Nicholas drank the bottle, and he spilled none of it on himself for once.

The anaphora test shows that the contents of containers are entities that are available for coreference if the containers themselves are. This is at least partial confirmation then of an approach like the one in TCL, in which a separate discourse entity for the contents is introduced as the result of using transfer to reflect the effects of the type shift of the coercion at the level of logical form.

There is an intriguing connection between these coercions and the resultative construction. Here are some examples of this construction.

(8.72) a. The bottle froze solid.

b. Nicholas drank the bottle dry.

c. Julie wiped the table clean.

d. James hammered the metal flat.

In such examples, the freezing results in the contents of the bottle being entirely solid. That is, the property of being solid is predicated of the contents of the bottle after the freezing process is over; the table has the property of being clean after James has wiped it, and so on. Wechsler (20??) and others have made important contributions to the analysis of the resultative construction giving us reasons why only certain adjectives can go into the construction, but this construction still requires several stipulations when implemented in an event semantics, the standard approach to this construction.

Our type system allows us to do something else. We must only stipulate that the adjective itself in this configuration introduces a dependent type RESULT that has two parameters, the type of the event introduced by the main verb and the type of the direct object. This dependent type yields an aspect of the direct object, the aspect in which the object has the properties that result from its participation in the event. Wechsler observed that these properties must be of a certain kind. Thus, resultatives, depictives and *qua* predications all involve predications to aspects of objects. Resultatives, however, involve a dependent type in the predication; they introduce a particular sort of aspect and a particular kind of object elaboration—this result-state relation that I will just call *Result*. Some resultatives like (8.72a) involve two depedent types. The final logical form for (8.72a) after we have made the type shifts due to the coercions and transfer of the type shifts to logical form is this:

(8.72′) $\exists e, x(\text{bottle}(x) \wedge\ \text{freeze}(x, e) \wedge\ \text{Result}(u, e, x) \wedge\ \text{solid}(u))$

8.4.1 Cars and Drivers, Books and Authors

The famous Nunberg cases that we examined in chapter 3 also are examples of coercion involving dependent types.

(8.73) a. I'm parked out back.

b. The ham sandwich is getting impatient.

c. Plato is on the top shelf.

d. Thomas is on the top shelf.

Some of these are more difficult to interpret than others. (8.73a) for instance has a very easy interpretation with the coercion, whereas most speakers need a bit of help with (8.73b).

With the ”parking” examples, the TCL approach will assume that the verb licenses the introduction of a vehicle type associated with the argument. If that's the case, then we have just another lexical coercion introducing a dependent type, this time mapping the head type to the type of an associated vehicle. Is this argument available for subsequent reference? It would appear to be so.

(8.74) I'm parked out back. It's a Volvo.

(8.75) I'm parked over by the Esso FBO.[29] It's a Comanche.

On the other hand, it is not just the particular verb *park* in the passive form that that can coerce the relevant type shift in such examples.

(8.76) a. I'm out back. It's a Volvo

b. He's out back. It's a Volvo.

c. John's in front of the house. It's a Volvo.

The interpretation of the (8.76b) really depends on features of the discourse context. Consider the following two extensions of (8.76b)

(8.77) a. I just saw John. He's out back. It's a Volvo.

b. He's in Paris. #It's a Volvo.

c. Hi. Can you get our cars for us? He's out back. It's a Volvo. I'm over on the left. It's a Subaru.

The car interpretation of (8.77a) feels more awkward than (8.76a) or (8.77c). And the *it* (8.77b) is really very difficult to interpret when we take *it* to be an anaphoric pronoun, even when interpreters are primed to associate vehicles with agents. Such coercions seem to be triggered by the predication of a nearby location to agents unless the agent is not capable of owning a car. It would appear that the dependent type is licensed by the presence of a lexical element—e.g., a verb like *park*– or a relevant context in which it's easy to associate vehicles with agents in some way in nearby locations that the agent clearly does not occupy. (8.77b) fails to meet the discourse conditions that would trigger the coercion, and (8.77a) does so only with difficulty. This last condition on coercion is much more dependent on extra lexical and predicational resources and may be why this coercion has a more pragmatic feel to it. *Mutatis mutandis* for the coercion from authors to their books.

8.4.2 Verbs of Consumption

Sometimes the type constraints of certain verbs on their object or internal arguments are so specidfic that we omit the internal argument. This is true of certain verbs of consumption.

[29] An FBO is a flight base operation where planes can be parked.

(8.78) a. John smokes after dinner.
b. John drinks a lot at parties.
c. I've eaten.
d. Nancy drove to work
e. Lizzy climbed well.

These are not exactly coercions, but they are related. These verbs assign defeasible type requirements to their internal arguments that are sufficiently precise that they contribute to truth conditional content. One naturally understands (8.78a) as meaning that John smokes cigarettes after dinner. This type assignment is defeasible; John could smoke cigars, a pipe or even non-tobacco products in the right context—marijuana, opium, crack. But the latter are much less preferred. Similarly, (8.78b) is naturally understood as meaning that John drinks a lot of alcohol at parties, though in certain contexts we can change the type of what John drinks. TCL can model these implications within the default type specification logic. That is, we may take *smoke* using a dependent type that is a subtype of the type TRANSITIVE VERB and then stipulate:

- $\text{SMOKE}(\text{AGENT}, \sigma) > \sigma = \text{CIGARETTE}$

There remains the question of what information one should add to logical form for such intransitive uses of transitive verbs. Is there for example a variable at the level of logical form standing for the internal argument? The anaphora test tells us that such terms, if they are there at all, are not available for anaphoric coreference.

(8.79) Nancy drove to work. It used a lot of gas.

How to interpret the anaphora test in these cases is a delicate matter. Similar sorts of problems crop up in languages with incorporation, where incorporated nouns though contributing to truth conditional content serve only with difficulty as antecedents to anaphors. Though interesting, this is a subject that no longer concerns TCL and the theory of predication.

8.4.3 Aspectual Coercion

One of the truisms about the progressive is that stative constructions aren't supposed to support a progressive aspect—for instance (8.80a) is an example that is evidence for this generalization. Nevertheless, we do say and understand things like (1.16b-d) perfectly straightforwardly.

(8.80) a. # John is knowing French
b. John is being silly.
c. John is just being John.
d. John's being French.
e. John's being an asshole.

Contrast (8.80b-d) with their non progressive counterparts *John is silly, John is John, John is French* and it becomes obvious that the propressive form of *be* produces a coercion. Following the received wisdom concerning the analysis of the progressive, I will assume that when the progressive applies to a verb or verb phrase, the eventuality argument is required to be an eventuality that is not a state. However, when the progressive operator applies to a copular phrase, the constuction licenses the introduction of a dependent type mapping a state of type σ to a type of activity that when performed by an agent yields a result in which the agent is silly. This coercion is licensed by a particular predicational construction like the resultative construction.

This coercion is relatively robust. For example, it passes the anaphora test:

(8.81) a. John's being silly, and he's doing it to annoy you.
b. #John is silly, and he's doing it to annoy you.

The pronoun *it* as an argument to the verb *do* must pick up an event or event type in (8.81a), not a state; attempting to use the same construction to refer back in the state as in (8.81b) is impossible.

Let us suppose that there is a dependent type of the form $\textsc{activity}(\sigma, \alpha)$, whose input parameters are a state and the bearer of that state. The output is an eventuality that is an activity or process involving an object of type α. This dependent types is sort of the converse of the resultative dependent type. The associated transfer principle for this dependent type is:

(8.82) $\lambda P \lambda e \lambda x \exists \phi \exists s \exists \mathrm{z}\, {}^{\vee}\phi(x, e) \wedge \ \mathrm{Result}(e, s) \wedge P(s, x)$

In order to provide some derivations, I need to make one other assumption, about the copula. I'll assume that the copula introduces a state on which one can hang temporal information— the explicit copula can carry tense information in many languages. The state serves to localize temporally the predication. With the framework of aspects that I have introduced, such temporal localization requires the introduction of an aspect of the copula's subject argument; and the temporal span of the aspect is included in the state.[30] Formally speaking, this means that we have the following entry for the copula:

(8.83) $\lambda\mathcal{P}\lambda s\lambda x\exists z(\mathcal{P}(\lambda u(\text{O-elab}(u, x) \wedge \tau(u) \subseteq_t s))(z))$

If we now put the semantics of the copula together with the semantics of an intersective adjective like *silly*, then we get the following semantics for the VP:

(8.84) $\lambda s\lambda x\exists z(\text{O-elab}(z, x) \wedge \tau(z) \subseteq_t s \wedge \text{Silly}(z))$

When we now attend to the type shift engendered by the progressive operators applying to (8.84) and use transfer, we get the following meaning for (1.16b):

(8.85) $\exists e(e \circ \text{now} \wedge \exists\phi\exists s({}^{\vee}\phi(j, e) \wedge \text{ Result}(e, s) \wedge \exists z(\text{ O-elab}(z, x) \wedge \tau(z) \subseteq_t s \wedge \text{Silly}(z))))$

In words this says that John is doing some activity whose result state is *s* and *s* includes the temporal span of some aspect of John in which he is silly. These seem to me to be the right truth conditions for such a sentence, and this discussion shows how aspectual coercion falls within the framework of coercion using dependent types as I've set it up.

The analysis of the copula given here enables us to revisit a problem that concerned Szabo's account of relative predication. Recall the pair:

(7.14) a. I am unhappy now.

b. I am happy now.

My analysis of the copula given here predicts two aspects, one in which I'm happy, one in which I'm not happy.

(7.14′) a. $\exists s \circ \text{ now}\exists z(\text{O-elab}(z, i) \wedge \tau(z) \subseteq_t s \wedge \neg\text{happy}(z))$

b. $\exists s \circ \text{ now}\exists z_1(\text{O-elab}(z_1, i) \wedge \tau(z_1) \subseteq_t s \wedge \text{happy}(z_1))$

However, these aspects are defined only in terms of some temporal span that overlaps the moment of utterance. That is, they pick out the entire thin individual relative to two temporal spans. However, the intersection of those two temporal

[30] Some analyses of adjectives like, e.g., Higginbotham's **?** postulate a state argument in the adjective already. On such a view of the argument structure of adjectives, the copula could then be a pure identity predicate. I do not subscribe to such a view. I do subscribe to the analysis by **?**, inter alia, on which adjectives have an inherent degree argument as well, but that need not enter into the picture here.

spans is non-empty; it includes at least the moment of utterance. Thus, (7.14) end up implying that I am both happy and unhappy at a particular time (the ”now of the utterance) and so implies that (7.14) is inconsistent, unless some other way of specifying the two aspects shows that they are not merely temporal segments of the individual. Specifying aspects by means of other properties besides temporal span is precisely what relative predication is designed to do, and so while the pair in (7.14’) are predicted to be contradictory, pairs of sentences like *as a professor he is unhappy, as a married man he is unhappy* are predicted to be perfectly compatible.

8.4.4 Grinding

Grinding occurs when a bare count noun is required to have a mass interpretation. Grinding is triggered in part by the use of a bare noun in the singular and partly by the demands of a predicate, and so they qualify as a type of coercion.[31] There appear instances of copredication where we understand a bare noun both as a kind and as a mass.

(8.86) Snow is frozen water and all over my yard right now.[32]

Grinding is another dependent type whose introduction certain predicates will license. That is, the dependent type GRIND takes any physical object type or kind type as in (8.86) and converts it to a subtype of MASS. But it appears that this function is also lexically introduced. The mechanism of dependent type coercion approaches this copredication in the same way that it approaches copredication involving event coercion.[33]

8.5 Nominalisation

The grammatical construction known as nominalisation also involves a type shift. Nominalisation is a means for referring to entities that are not ordinarily under-

[31] As David Nicolas pointed out to me, grinding is not freely available, as bare singular nouns in English can have a kind or a mass interpretation.

[32] Thanks to Jeff Pelletier for this example.

[33] Note that we can get finegrained differences in grinding using types:

(8.87) Rabbit is usually yummy but not when all over your windshield.

We have a coercion from the meat sense where we generically quantify over that to rabbit muck or a portion of rabbit matter.

stood to be elements of the domain of discourse– inhabitants that is of the type e. Nominalisation can be understood as a function on types—from types that are not subtypes of e into e. Consider the ”er” nominalisation which takes a verb and picks out an agent who does the sort of action denoted by the verb. Here are some verbs that undergo this shift:

(8.88) clean, carry, lie, bake, sell, report, buy, create, write, invent, perform, dance, cater, fight, box, fence, ski, climb, sail, port, auction, preach, demonize, photograph, paint, choreograph, teach, swim, wrestle,...

We could formalize the ”er” transformation at the level of types as a function from the type of a transitive verb to the type of its subject argument, except that sometimes this nominalisation doesn’t work that way—in particular when there already is another word in the lexicon for an agent that does the activity denoted by the verb. For example consider the verbs *cook*. The ”er” nominalisation of *cook* doesn’t denote someone who cooks, because there is already the noun *cook* that does this. A cooker is instead a pot or something one cooks in. So a proper formulation of the ”er” transformation for a transitive verb V where γ is the subtype of e out of which the generalized quantifier type α is formed (i.e., $\alpha = (\gamma \Rightarrow \textsc{t}) \Rightarrow \textsc{t}$) should go as follows

- The ”er” Arrow
  er: $\alpha \Rightarrow (\textsc{evty} \Rightarrow (\beta \Rightarrow \mathrm{V}(\beta, \textsc{evty}, \alpha))) \longrightarrow \rho(\mathrm{V}, \alpha)$, where $\rho(\mathrm{V}, \alpha) \sqsubseteq \beta \sqsubseteq e$

$\rho(\mathrm{V}, \alpha)$ like ϵ is an underspecified type, but it picks out individuals who V. Accompanying this type shift is a transfer rule giving the contribution of ”er” at the level of logical form. We can proceed as in Asher (1993), for instance.

However, the realm of nominalisations is much larger than what was canvassed in Asher (1993). Asher (1993) only looked at saturated abstract entities—propositions, facts, eventualities and so on. There are also elements of e corresponding to unsaturated entities like properties and concepts:

(8.89) a. the property of being triangular

b. the concept of being red

c. being triangular

d. the property of properties such for any property everything that is a man also has that property.

Natural languages have sufficient resources to map any type into a subtype of e. That is, we have another very general coercion mechanism that is licensed by a variety of nominal constructions.

- The Nominalisation ”Arrow”:
  $\nu\colon \alpha \longrightarrow e$, for any type α

We should distinguish the nominalisations of two types if the types are distinct. That is, ν should be $1-1$. There is also an effect at the level of logical form; corresponding to ν is a function symbol that picks out the entity that corresponds to the object of higher type. At the cost of a little ambiguity, I'll use ν to represent that function symbol as well.

The first question to ask is whether the system of types with a nominalisation arrow remains consistent? The set of all types that can be produced with finite applications of $\Rightarrow$ and the definable dependent types is at most countably infinite, given that we have a finite number of type constructors. So there is no problem with the assumption that there is a 1-1 function from all types into e, if we assume that e has a countably infinite set of subtypes. We should like to say the same thing for each of these types inhabitants as well. But as we are not in a set theoretic interpretation, there is no need, for instance, to suppose that all possible functions from e to T are inhabitants of the type $e \Rightarrow \mathrm{T}$. Indeed, we had better not! To avoid cardinality problems we must restrict the inhabitants of each type to the linguistically definable properties or at least some countable set.

Our nominalisation operator allows us to make sense of predications to properties. We can now make sense of predications like that in (8.90):

(8.90) The property of being nice is nice.

And we can compare these predications with predications to saturated entities:[34]

(8.91) a. Being nice is nice.
b. It's good to be good.
c. To be always cute is not cute.

Note that the nominalisations of the saturated entities seem to go much better with these quasi self-referential predications than the nominalizations of unsaturated entities.

[34] For an analysis of these constructions, see Asher (1993) and Hegarty (2003).

(8.92) a. The property of being a horse is a horse.

b. The property of being triangular is triangular.

c. The property of being red is red.

The general observation is that properties of individuals do not apply to their own nominalisations, analytically. That is, the sentences in (8.92) are *analytically* false. There are some exceptions:

(8.93) a. the property of being a property:

b. The property of being a property is a property.

(8.93b), in contrast to the examples in (8.92) seems *analytically true*.

These data are in part explained by TCL's typing conventions. By distinguishing between objects of type P and abstract or informational objects of type i, we know that predicates that are typed as physical properties ($\text{P} \Rightarrow \text{T}$) cannot apply to purely informational objects. Nominalisations of properties, however, must be informational objects; they are clearly abstract. So we can explain the analytical falsehood of (8.92) simply by appealing to this type distinction. Once again we see an empirical vindication of our typing system.

We must now consider what the nominalisation arrow does to the underlying logic. Does our system go inconsistent, now that it is possible to predicate properties of their nominalisations? In a typed system, we cannot construct the Russell property or other directly self-referential objects. However, we can come close to expressing self-application as well as non-self-application once we have nominalisation. Below is the lambda term for non-self-application as applied to properties of type $e \Rightarrow \text{T}$. Recall that ν not only names a type shift but also gives the nominal correlate of the property in logical forms.

(8.94) $\lambda P \neg P(\nu(P)),\ \langle P \colon e \Rightarrow \text{T},\ \nu(P) \colon e \rangle$

(8.94) is a term itself of type $(e \Rightarrow \text{T}) \Rightarrow \text{T}$, the type of a generalized quantifer or second order property. It is the property of those first order properties that do not apply to their own nominalization.

It turns out that many (first order) properties have the property (8.94). If all types have a nominalized counterpart, then (8.94) has a correlate in e, namely,

(8.95) $\nu(\lambda P \neg P(\nu(P)))$.

In the type free lambda calculus, the Russell paradox can be reconstructed, when (8.94) applies to itself. However, in the typed system even with a nominalisation type shifter, things are not so simple. Applying (8.94) to its own nominalisation isn't even well formed, since (8.95) is of type *e* and (8.94) is of type $(e \Rightarrow \textsc{t}) \Rightarrow \textsc{t}$. It's important to notice that the nominalisation shift does not make *e* a universal type (as in Martin Löf's original system, proved inconsistent by Girard (for a nice exposition see Coquand (1986)). So there is no Burali-Forti paradox for the present system, although there are plenty of incompletenessesCoquand (1986).

Let's now see if we can derive (8.93) as an analytical truth. Indeed we can. Once again, let's go through the steps.

(8.96) is a property:
$\lambda x \exists P\ \nu(P) = x$

(8.97) The property of being a property:
$\nu(\lambda x \exists P'\ \nu(P') = x)$

(8.98) The property of being a property is a property:
We need to show:
$\lambda x[\exists P\ \nu(P) = x]\{\nu(\lambda x \exists P'\ \nu(P') = x)\}$

1. $\nu(\lambda x \exists P\ \nu(P) = x) = \nu(\lambda x \exists P\ \nu(P) = x)$ Identity
2. $\exists Q\ \nu(Q) = \nu(\lambda x \exists P\ \nu(P) = x)$ 1, E^2G
3. $\lambda x \exists Q\ \nu(Q) = x[\nu(\lambda x \exists P\ \nu(P) = x)]$ 2, Abstraction

Similarly we can derive that *the property of not being a property is a property* is also an analytical truth.

(8.99) the property of not being a property
$\nu(\lambda x \neg \exists P\ \nu(P) = x)$

(8.100) Show that this is also a property:

- $\nu(\lambda x \neg \exists P \nu(P) = x) = \nu(\lambda x \neg \exists P \nu(P) = x)$
- $\exists Q(\nu(Q) = \nu(\lambda x \neg \exists P\ \nu(P) = x)$
- $\lambda y \exists Q \nu(Q) = y[\nu(\lambda x \neg \exists P \nu(P) = x)]$

Cases that we can't (shouldn't?) handle (unless we invoke polymorphism as in ML)[35]

- there is only one relation of identity
- the property of self application applies to itself.

In fact, we can prove a quick general consistency theorem by giving the interpretation of TCL within the category of set. Remember, the set conception is **NOT** the right *ontological* conception, but we are interested in consistency here, not truth. So as long as we can prove all the rules and type constraints within this model, we are OK. I will not take account of the dependent type of fictional objects and loose talk in this model, however.

- atomic types are sets of urelements. $\sqsubseteq \mapsto \subseteq$.
- • types have a categorial interpretation as pullbacks (Asher 2007a, 2007b)—in the category of sets as a set of equivalence classes over pairs.
- Dependent, functional, dynamic types all have evident interpretations in the category of set in general.
- Nominalization is the function from the collection of TCL definable functions (a countable collection) into sets (Wiener Kuratowski identity)

Therefore:

Theorem 1 *The type theory of TCL is consistent if ZFU is consistent.*

Our theory shows that there are certain predicates that necessarily do not apply to any objects. Consider the predicate that expresses the property of not having any properties. This seems sensible enough. But it turns out to look like this

(8.101) $\lambda x \forall P \neg P x,\ \langle P\colon e \Rightarrow \mathrm{T}\rangle$

Suppose we now instantiate $P = \lambda x \neg P_0 x$ and let a be any object such that $\lambda x \forall P \neg P x[a]$. By intantiating P to P_0 and application, we get

(8.102) $\lambda x \neg P_0 x(a) = \neg P_0(a))$

But on the tother hand, instantiating $P = \lambda x \neg P_0 x$, we get

[35] Thanks to Renaud Marlet for the suggestion about ML.

(8.103) $\neg\lambda x \neg P_0(a) = \neg\neg P_0(a) = P_0(a)$

If the system is consistent then there is no object (of any type) that has the property $\lambda x \forall P \neg P(x)$. If we think about this in set theoretic terms, however, this makes a certain a certain amount of sense: there is no object of ZFU such that no set contains it in ZFU. However, we can't say this about the property of non self-application. Many properties have the property of not applying to their nominalisations. But not all properties have this property—the relation of identity, for example, or, to use an example from the medieval philosopher Nicholas of Cusa, the relation of being not other than.[36]

8.5.1 Pluralities

There is an important semantic distinction that involves the semantics of In a theory of predication then we have to be able to mark those predicates that type for a collective predication of a plural type vs. a distributive one. That is, a predicate like *disperse* or *gather* takes a collectively understood plural argument. On the other hand, other predicates go either way, but there is a big difference as to whether the predication is understood collectively or distributively.

(8.104) Three students lifted the piano.

When (8.104) is understood as a collective predication, then all three students are involved in lifting the piano together, whereas understood distributively it says that each of the three students lifted the piano individually.

There does seem to be then a map from groups to their members and back again that affects predication. For some predicates (by which I also include quantifiers), the plural argument can be understood either collectively or distributively, whereas other predicates force their plural arguments be understood in a particular way. Like almost all of maps that underly coercion, the map from collective to distributive and back again also supports copredication (Asher and Wang, 2003):

(8.105) The students worked very hard (distributive) and mowed the whole meadow (collective).

Making sure that an argument is collective ought to be a matter of type checking in a typed system, but it is a subtle matter how the argument is transformed by the

[36] Nicholas of Cusa proposed the following analogy for the Trinity: *non aliud non aliud quam non aliud est.*

composition process and by subsequent discourse (see Wang 2005 for details). For instance, a predicate like *mow* may be understood either collectively or distributively; its final interpretation depends on the nature of its argument and the discourse context.[37]

(8.106) The students mowed the meadow on the left and the professor the meadow on the right.

Such examples suggest that coercions work like anaphors in ellipsis contexts allowing for ”sloppy” interpretations. Accordingly, I suggest that the recovery of the target in ellipsis proceeds prior to the coercion steps, just as the recovery of a verb phrase with an anaphor must proceed prior to the assignment of an antecedent to the anaphor as argued for in Asher (1993).

8.5.2 Metonymic Predications

Kleiber (1999) and others have proposed a metonymic model of predication, on which predications that apply properly to parts of objects are sometimes predicated of the whole. I argued earlier that this cannot be a general model of what is going in the logical metonymy cases unless we empty the notion of part and whole of content. However, there are many examples of predication where metonymy is operative. Recall the following examples from chapter 3.

(3.16) a. Paul est bronzé. (Paul is tanned).

b. Les américains ont débarqué sur la lune en 1969. (The Americans landed on the moon in 1969).

c. Le pantalon est sale. (The trousers are dirty)

d. Le stylo est rouge. (The pen is red)

As we saw earlier as well, part whole relations may be involved in copredications:

(3.17) a. Paul est bronzé et très athlétique. (Paul is tanned and very athletic)

b. Les américains ont débarqué sur la lune en 1969 et ont mené une sale guerre en indochine. (The Americans landed on the moon in 1969 and waged a dirty war in Indochina.)

c. Le pantalon est sale et troué. (The trousers are dirty and torn)

d. Le stylo est rouge et très cher. (The pen is red and very expensive)

[37] Thanks to Ofra Magidor for the following example.

Metonymy proves to be a simple coercion operation using the map from an objects to their parts. Clearly the existence of such a map is guaranteed: if the whole exists so must its parts (though the reverse map is not guaranteed to exist). Once again, however, we can see that such examples don't really shift the type of the expression that denotes the whole. There are no quantificational puzzles here. We don't individuate *the Americans* any differently in the first or second clauses of (3.17b), and we don't get any counting differences either the way we did for books. Thus metonymy is of a piece with other coercions we have examined in this chapter.

Chapter 9

Adjectival Modification: Coercion and Loose Talk

The previous chapter investigated different kinds of verbal or VP coercion. There are also interesting coercions that result from applying an adjective to a noun. For instance, adjectives that constrain the denotations of the nouns they modify to be made out of a certain kind of matter, what I'll call here *material adjectives* have a coercive function. There have been discussed in the literature, but there are other adjectives that are also coercers but less studied. I will give an overview of these here. My analysis of the coercive function of adjectives will lead us to study what might be in fact another form of predication—-the predication the occurs in so called ”loose talk.” Loose talk is a very common phenomenon but also very puzzling; it occurs when we predicate properties of objects that they only have in some approximate or ”loose” sense. What that sense is, I'll explain presently.

9.1 Material Adjectives

Material adjectives provide more evidence for TCL's context sensitive and dynamic notion of typing. These adjectives supply the material constitution of objects that satisfy the nouns the adjectives modify, as in (9.1):

(9.1) glass (wooden, stone, metal, tin, steel, copper) bowl[1]

[1] The explicit adjective *wooden* versus *wood* indicates that it makes sense to think of these modifiers as adjectives.

According to the theory of qualia in classic GL, the material or constitution quale, the material out of which objects that satisfy the noun are made, is specified by the lexical entry of the noun. But in (9.1) it is not *bowl* that specifies the material constitution, but rather the modifier. This situation is in fact very common for artifacts, as they can be made out of many many different materials.

Further, many objects can be constituted from materials that can affect the typing of the head noun.

(9.2) a. stone lion (vs. actual lion)

b. paper tiger (vs. actual tiger)

c. paper airplane

d. sand castle

When the constitution of the object is given by an adjective whose denotation is not a possible type of constitution for the type of object denoted by the head noun, we get a shift in the type of the head noun. This shift is important because it supports different sorts of inferences.

(9.3) a. A stone lion is not a lion (a real lion), but it looks like one.

b. A stone jar is a jar

c. ?A paper airplane is an airplane.

As lions are living animals, they cannot be made out of stone; so a stone lion is not a real lion. *Stone* is no longer an intersective nor even a subsective modifier. In contrast a stone jar remains a jar, as jars can be made out of stone. In this case *stone* is intersective. I am not sure whether a paper airplane is an airplane. If one thinks of airplanes as having certain necessary parts like an engine or on board means of locomotion, then most paper airplanes aren't airplanes. On the other hand, many people tell me that their intuitions go the other way.

Material modification and discourse structure reveal curious interactions with world knowledge. Consider for instance (9.3c). It is part of world knowledge that paper airplanes don't have engines. Nevertheless, it appears that *the engine* gets interpreted in the same bridging way in the two examples below.

(9.4) a. John closed the door to the airplane. The engine started smoothly.

b. John made a paper airplane in class. The engine started smoothly.

The bridging inference in (9.4b), in which the engine is inferred to be part of the airplane, would seem to go against world knowledge; the requirement to maximize discourse coherence and to resolve underspecified conditions trumps world knowledge. This seems to indicate that in fact *paper* doesn't behave like *stone* in *stone lion* but rather remains something closer to intersective. *Paper* isn't *exactly* intersective, because paper airplanes are in some half way land —in principle they could have engines, as (9.4b) demonstrates and even transport people and goods but typically they do not. Recalling that types are concepts, it would appear that the type of *airplane* has shifted but not so much as with the shift from LION to STONE LION.

This presents us with a puzzle. Can such predications actually change the type of the modifier from intersective to non intersective or from subsective to non-subsective? Or is it rather that the modification actually changes the type of the head noun? If the type of the head noun is what changes, that would explain not only the cases like *stone lion* and *stone jar* but also the puzzling in between cases like *paper airplane*, or perhaps also *sand castle*.

To figure out our puzzle about material modification, let's first try to write down how a material adjective specifies the matter of the objects satisfying the common noun phrase. We might specify the following for a material adjective:

(9.5) material-adj: $\lambda P \lambda x(P(x) \wedge \exists u(\mathrm{MAT}(u) \wedge \text{made-of}(u, x)), \langle u : \textsc{type}(\mathrm{MAT}), x : \textsc{phys-obj}\rangle$

But this gets things wrong with *stone lion* or *sand castle*, because it we can still infer monotonically that a stone lion is a lion. This clashes with the inferences encoded in the type hierarchy according to which LION is a subtype of ANIMATE. Our rules for lexical inference should preserve subclass/superclass relations; so we should be able to infer from the fact that something is a stone lion the fact that it is animate. Similarly, we will infer from the fact that something is a sand castle to the fact that it is a habitation (place where one lives like apartments, houses, villas, cottages, etc.) Both of these inferences seem dubious at best.

It is from similar examples that Kamp and Montague concluded that adjectival modification takes place at the level of intensions. In the TCL framework we would have,

(9.6) material-adj: $\lambda P \lambda x({}^{\vee}\mathrm{MAT}(P(x))), \langle x : e, P : s \Rightarrow (e \Rightarrow t)\rangle$

But this logical form prevents us from making the right inferences for many other cases of material modification; e.g., we want to be able to infer from the fact that something is a stone jar that it is a jar. And consider the case of the paper airplane as an intermediate one; it appears we can accommodate the presence of an engine at least enough to process examples like (9.4b), but it also feels wrong to accept the inference:

- x is a paper airplane ⟶ x has an engine.

A better account is a typed approach that can split the difference between these two. In addition (9.5), we need to stipulate that the predication of the material adjective coerces its argument's type into a dependent one. For instance applying the adjective *paper* to *airplane* converts the type from simply AIRPLANE to an object of the type MADE-OF(PAPER, AIRPLANE). Formally, we change the typing context of (9.5) exploiting the type of the argument of P. P must be a type of a common noun phrase and thus of the form $\alpha \Rightarrow \textsc{t}$. I will pick out the type α with a function arg(P).

(9.7) $\lambda P \lambda x(P(x) \wedge \exists u(\mathrm{MAT}(u) \wedge \text{made-of}(u, x))$
$\langle u : \textsc{type}(\mathrm{MAT}), x\text{: } \textsc{made of}(\textsc{type}(\mathrm{MAT}), \textsc{type}(\arg(P)))\rangle$

paper airplane would thus yield the following logical form plus typing context:

(9.8) $\lambda x(\text{airplane}(x) \wedge \exists u(\text{paper}(u) \wedge \text{made-of}(u, x)),$
$\langle u : \textsc{paper}, x\text{: } \textsc{made of}(\textsc{paper}, \textsc{airplane})\rangle$

Unlike the standard story with qualia in classic GL where these come as part of the fixed meaning of a noun, predication involving a material adjective in TCL specifies the matter out of which the satisfiers of the noun are made. Adjectival modification *changes* the the information that would typically to be given by the *qualia* associated wtih the noun meaning in classic GL. This shows yet again how types are sensitive to the predicational context; information determined by *qualia* in classic GL isn't fixed by single lexical entries but dynamically evolves within the predicational context. And it's yet another reason for dispensing with *qualia* and moving to TCL's underspecified types and more dynamic notion of typing.

The modification of the noun type by the predication of the material adjective shifts the type of the variable that is the lambda abstracted argument in the common noun phrase's logical form. This furnishes a counterexample to the Head Principle. But here this makes sense; determining the matter of something may change its underlying nature. In this our type theoretic analysis echoes the Montegovian analysis of adjectival modification. And it inherits a similar worry. What exactly can we infer about objects of such types and how do such types affect the predication of the head noun predicate to the variable with the modified type?

The dependent type MADE OF(PAPER, AIRPLANE) suggests that the object is essentially made of paper. Why might this be a reasonable hypothesis? Recall the discussion of Kripke's *Naming and Necessity* concerning the wooden lecturn. It turns out, if he's right and I think he is, that wooden lecturns have different essential properties and hence individuation conditions from say plastic lecturns, in the sense that a wooden lecturn is necessarily made of wood and no wooden lecturn could remain the same lecturn if it were magically transformed into some other material. Similarly, paper airplanes are necessarily made of paper, which distinguishes them from airplanes generally. This also corrects that conflation in classic GL between the modal flavors of the different qualia. Such constitutive ”qualia” contribute to individuation conditions in a very different way from say telic and agentive qualia that are as Pustejovsky rightly points out, also involved in the individuation of artifacts. The purposes to which an object is typically put are modal properties themselves. That is, using the reflection of the type hierarchy in logical form:

- $x : \textsc{made of}(\textsc{paper}) \to \Box\exists y(\text{made of}(x,y) \wedge \text{paper}(y))$
- $(x : \alpha \wedge y\colon \epsilon(a1,\alpha) \wedge \epsilon(a1,a2) \sqsubseteq \textsc{telic}(a1,a2)) \to \Box(\phi > \exists y(\phi_{\epsilon(a1,a2)}(x,y))$

But within our > logic we can say more about objects of the type MADE-OF(α,β). Let us suppose that we associate with types another dependent type MAT that is quite related to MADE OF. But it returns the matter out of which objects of a type may be composed. So for a type like JAR we have the following subtype formulas

- $\textsc{stone} \sqsubseteq \textsc{mat}(\textsc{jar})$
- $\textsc{earthenware} \sqsubseteq \textsc{mat}(\textsc{jar})$
- $\textsc{glass} \sqsubseteq \textsc{mat}(\textsc{jar})$
- $\textsc{metal} \sqsubseteq \textsc{mat}(\textsc{jar})$
- $\textsc{wood} \sqsubseteq \textsc{mat}(\textsc{jar})$

Then, we can write down the following type constraint that helps us discover the types of objects that MADE-OF gives us.

(9.9) $(\textsc{made-of}(\alpha,\beta) \rightarrow (\alpha \sqsubseteq \textsc{mat}(\beta) \leftrightarrow \textsc{made-of}(\alpha,\beta) \sqsubseteq \beta).$

We can properly predicate ’jar’ of an object if in fact its type is compatible with that of being a jar. And our constraints above show this to be true. To be precise, for *stone jar*, we have the following logical form:

(9.10) $\lambda x \lambda v \mathrm{jar}(v) \wedge \exists u(\mathrm{stone}(u) \wedge \mathrm{made\text{-}of}(u,x)),\ \langle v\!: \textsc{jar}, u : \textsc{stone}, x : \textsc{made-of}(\textsc{stone}, \textsc{jar})\rangle$

Accommodation will lead to a proper predication and reduction of the lambda terms.

(9.11) $\lambda x \mathrm{jar}(x) \wedge \exists u(\mathrm{stone}(u) \wedge \mathrm{made\text{-}of}(u,x)),\ \langle u\!: \textsc{stone}, x\!: \textsc{made-of}(\textsc{stone}, \textsc{jar})\rangle$

Now there is no obstacle in this case to infer that x is really a jar. Since stone is something jars can be made out of, a stone jar is a, litterally, jar. Similarly for wooden airplanes, metal airplanes, even plastic airplanes.

On the other hand, stone lions can’t be real lions, because the matter out of which real lions are made can’t be stone. So (9.9) doesn’t apply, and we can’t infer that stone lions are lions. Furthermore, this implies that there is a conflict between the type assignment to the variable and the demands of the predicate. And furthermore, no rule allows us to resolve the conflict. None of the • exploitation rules apply and neither does in this case Accommodation.[2]

Nevertheless from the logical form of *stone lion*, the predicate *lion* still applies to a variable which has now a type that is incompatible with being a lion, since the variable picks out an object that is made out of a material that lions can’t be made of. In this case, we understand the predication of *lion* to the variable typed MADE-OF(STONE, LION) ”loosely.” In what follows I’m not giving an analysis of *when* we are to interpret a predicate loosely. I’m only saying that examples like *stone lion* or perhaps *paper airplane* are examples of loose talk. Type mismatches, I hypothesize, are neessary. But isn’t right to say that every type mismatch can be understood as an example of loose talk. Sometimes, there is no recovery from a type mismatch, as we saw with the examples that motivate a type driven account. It is, I suspect, a complex business to give sufficient conditions for loose talk. Intentions and a certain amount of convention, I suspect, are involved. In keeping with this, it appears that there are no good tests for loose talk that I have been able to find in the literature.

[2] Paper airplanes are arguably not airplanes really either, because airplanes can’t be made out of paper. Paper, or at least normal paper, lacks the required strength to weight ratio needed for an airplane to fulfill its functions. But we can also see why paper airplanes constitute an intermediate case, as it’s not completely ruled out that airplanes are made of paper.

9.1.1 Loose Talk

Loose talk, metaphor, poetic license are all ways that predications that don't literally work can be reinterpreted even without the benefit of context. I'll take loose talk to be generated by type clashes that occur within a certain kind of discourse context to be elaborated on shortly.

Let's first consider a simple geometric case.

(9.12) (Pointing to a shape that a child, Pat, has drawn) You've drawn a circle Pat.

(9.13) I need a circular table to put in that corner.

We call things circles or circular when they only approximately resemble mathematical circles. When we do this we are speaking loosely. To interpret such sentences loosely, we make use of a set of background alternatives. The loose interpretation of such sentences is the object that Pat has drawn or the table that I need has a shape that is closer to that of a mathematical circle than any of the relevant alternatives—in this case, simple geometric shapes. There's probably no way to accurately describe the geometric shape drawn by Pat and saying ”You've drawn something that resembles a circle” might well not meet the pragmatic requirements of the conversation.

There are two notions involved in this intuitive picture that require analysis—the notion of alternatives, and the notion of closeness. Let's first turn to the notion of alternatives. Which set of alternatives is at issue depends on the predicate that is to be interpreted loosely. In general this seems to be a matter of the internal semantics of the predicate, not its external denotation. We look to (the lowest proper?) supertype in the type hierarchy to find the relevant alternatives. Thus, when the predicate 'circular' is to be interpreted loosely, then we look to the supertype of 2D-shape. The alternatives are given by the other types just under this supertype—-square, rectangle, triangle and so on.

Now we have to understand the notion of ”closeness” involved in loose talk. On the picture that I am advocating, loose talk involves a notion of closeness between some object that is paradigmatically, by definition, or prototypically a P and the object that we are calling loosely a P. We then compare that degree of closeness and the closeness of the object in question and a paradigm of P', where P' is some relevant alternative to P.

It is perhaps helpful here to think about the parallel case of vagueness in which a notion of closeness is often thought to be involved. Vague predicates support what is called Sorites reasoning. The premises of this argument look remarkably unobjectionable and yet we arrive at a contradiction. To give a time honored example, consider the following theses:

(9.14) a. 0 grains of sand do not make a heap.

b. One cannot create a heap from something that is not a heap by adding one grain of sand.

c. 50,000 grains of sand grouped together constitute a heap.

The first two theses entail something that seems necessarily false and contradicts the third premise (9.14c). Nevertheless, they are separately plausible. The thesis that concerns a measure of closeness is the second one. All vague predicates support a similar thesis that we can generalize as the principle of the equivalence of observationally indistinguishable objects (EOI) with respect to a property P.

- An object x is observationally indistinguishable from an object x' with respect to an observational property P iff x has P iff x' has P.

The second thesis concerning heaps claims that adding one grain of sand to a collection of grains of sand produces an observationally indistinguishable collection with respect to the property of being a heap.

EOI certainly provides a measure of closeness that is relevant to the application of a predicate. But notice that it is quite different from the standard for loose talk. The standard for loose talk is given by a paradigm or a prototype instance of the predicate in question. There is no ”local” comparison between indistiguishable near circles in the case of loose talk.

Could we construct a Sorites-like series for the predicate *is a circle*? Let f be a topological transformation of a closed region c that moves some but not all points on the boundary of c to or away from the interior of c by at most $\frac{1}{n}$ for some natural number n such that $f(c)$ is perceptually indistinguishable from c.

- c_0 (a mathematical circle), $f(c_0)$, $f(f(c_0))$, ..., $f^\alpha(c_0)$ (definitely not a circle but rather, say, an ellipse).

It appears that we could fall prey to a Sorites reasoning and a paradoxical conclusion in the case of shape as well as for color or for heaps. Nevertheless, the reasoning here is quite different with respect to the case of loose talk. When figuring out whether each $f^n(c_0)$ is a circle or not in loose talk, we make reference in thought to a paradigm circle. In the Sorites argument, the comparisons are local.

9.1.2 Metrics

In our analysis of loose talk,we have now invoked a relation of closeness, which requires us to consider the notion of a measure or metric. Where does this metric come from? Is the metric to be located in extensions? No, the metric depends on features associated with the predicate that make up its internal semantics. Words for geometric shapes are one of a handful that have explicit definitions; in such cases, we can look to the definition to devise a metric that might be a discrete measure suggested by the deformation function above or perhaps a more continuous measure. In general, however, words don't have explicit definitions. So, should we try to get a measure by looking to a predicate's extension? This seems not very promising, because of externalist considerations. The extension of many predicates is not given by "what is (solely) in the head" of a competent speaker. The denotation of *lion* for example is determined by the criteria for belonging to the same species, which is presumably some sort of DNA code. Now I can judge whether something's a stone lion, even though I have no idea really what the species identifying criterion for lions are. It's certainly not what I use to judge whether something's a stone lion. That's based on looks. Or take the following case of what seems to be loose talk:

(9.15) (Pointing to a very weak alcoholic drink, I exclaim) This is water!

My predication to be understood loosely is based on a superficial feature of water—that it's tasteless. Such superficial features flesh out the internal semantics of predicates whose core is the TCL predication rules. Thus, competent speakers do make judgments as to whether a loose predication holds based on features that are part of the internal semantics of a predicate.

From the perspective of TCL, this internal semantics is based on the type system, but the type system of TCL as presently conceived does not provide sufficient information. First of all, we have seen that the effects of coercion and copredication obtain almost exclusively at the upper end of the type system. That is, the types at the upper end of the type hierarchy, which include the basic ontological differences between substances of entities as well as the higher order functional types, are those that are clearly relevant to predication, while the more finegrained types seem largely not to play a role in the linguistic system at least with respect to . But for loose talk, we would need to resort, it would seem, to rather low level types like LION, CASTLE, CIRCLE and so on. The type system contains a subtype relation which reflects universal generalizations; thus, if $\alpha \sqsubseteq \beta$, then in the object language $\forall \vec{x}(\phi_\alpha(\vec{x}) \rightarrow \phi_\beta(\vec{x})$, where ϕ_α is a λ term whose type is α. An additional relation in the type system would be a defeasible subtyping link, which corresponds to the truth of a generic statement in the object language. Many type hierarchies incorporate both sorts of relations on the type structure. If we were to do this, TCL could encode the requisite information for defining the metric pertinent to many loose predications. Generic statements express the right sort of properties for figuring out whether something is closer to a lion than a giraffe—-for instance *male lions have manes, adult lions have big strong paws, lions have whiskers, giraffes have long necks* are all statements that provide properties relevant to determining whether something loosely speaking is a lion rather than a giraffe.

As with the case of circles, there is a certain amount of perceptual information relevant to determining whether a statue is, loosely speaking, a lion or not. Thus, in order to make sense of loose talk we would need, in many cases, links between types and perceptual information that lies outside the TCL system properly speaking. For example, using the deformation function for shapes together with the notion of alternatives, we can capture the notion of closeness. This metric is defined largely with respect to perceptual information.

(9.16) That object *qua circle is closer (count the number of iterations of the deformation function) to being a protoypical circle than to a prototype of any other geometric shape.*

It's also clear that not all perceptual information and not all generic statements involving the predicate are relevant to predications of loose talk. To return to (9.15), it's certainly a generic and even an analytic truth that WATER is a subtype of MASS. Nevertheless, this is not relevant to the metric needed for evaluating the loose talk in (9.15).

The idea that predications can involve loose talk invites us to take a look again at relative predications or standard predications involving proper names. While there is no comparison of alternatives that goes on in examples like (9.17), they also invoke superficial features associated with the name.

(9.17) a. You're no Jack Kennedy.

b. He's an Einstein.

c. Nicholas as David was not very convincing.

The predicative use of names indicates that the names are conveying information–Nicholas qua David has certain properties typically associated with the bearer of the name. To be a Jack Kennedy, you have to have certain properties we associate with John Kennedy—charismatic, being a leader, being a force for good. Similarly, to be an Einstein you have to possess properties associated with Einstein—being really smart. As for David, well, if you know who I'm talking about, then you'd accept generic statements like:

(9.18) David wears fancy glasses.

(9.19) David wears Capra and Cavelli shirts and slacks.

The properties or traits associated with the predicative use of the name can be gleaned from such generics, but once again not all properties of generics are relevant to the predication, for example those that follow from essential properties of David.

(9.20) David is a human being (entails $\forall x(x = d > \text{human}(x))$)

The predicative use of a name picks out a set of distinctive characteristics that pick out David. Distinctive characteristics are associated with a type. It is these distinctive characteristics that form the internal semantics of the language, and it is these that are encoded in the TCL type hierarchy using the defeasible subtype relation.

By counting characteristic traits or constructing a metric from perceptual information as in the case of shapes, we can define the ”standard” bearers or ”prototypes” that are used to anchor the metrics for loose talk.

- A prototype of some type α is an object x such that there is no object qua α that fits more of the associated traits of α than x does.

We are finally now able also to give the truth conditions for loose talk.

- $P(x)$ is satisfied loosely with respect to an assignment g at a world w in a structure $\mathfrak{A}$ iff $g(x)$ is closer to elements of $P^{\mathfrak{A}}$ than $g(x)$ is to elements of $P^{\mathfrak{A}}$, where Q is some relevant alternative to P.

An important difference between this approach to loose talk and other possible approaches like (Sperber and Wilson (1986) or Carston (2002) is that the actual extension of the loosely interpreted predicate does not change. If we change the actual extension of the loosely interpreted predicate then we render *false* statements involving the predicate that are strictly and literally true. Here is one example:

(9.21) All points on a circle are equidistant from a single point.

This proposition remains true on the present account.

In my account of loose talk, we end up making sense of stone lions as really a loosening of the predication relation involved; stone lions are lions only in that they resemble lions with respect to shape more than other alternatives —other common animals. Similarly, stone lions aren't strictly speaking lions, but they are closer to lions on the relevant metric (here shape) than any of the alternatives suggested by *lion* We can also make sense of the intermediate cases. When we say that paper airplanes are airplanes and sand castles are castles, this is also a stretch or a loosening of the predication involving the predicate *airplane* or *castle*. Sand castles aren't really castles, but they look more like castles than anything else.

My analysis of loose talk applies to other troublesome adjectival modifications. The troublesome adjectives I have in mind are the well known non subsective adjectives like *fake, ersatz, alleged*. These seem also counterexamples to the Head Typing Principle because the adjective really erases or generalizes the head noun's argument. In effect we have a type *downdate*. That is *a fake lion* is no longer a lion but something fake. *An alleged criminal* is not a criminal but someone that is alleged to be a criminal and so on. For at least some of these it seems that we can use the analysis of loose talk developed here. Fake lions aren't lions but they are more like lions with respect to some set of contextually salient features than any other alternative.

9.2 Evaluative Adjectives

Adjectival modification can also affect whether things have a function. Consider again (1.33a):

(1.33a) good lunch, good rock

It would be silly to assume that *rock* has some function on its own. As a natural kind TELIC(ROCK) is undefined in TCL, as I have set things up. But *good* can coerce the type of the natural kind into some sort of an artifact, for which we have seen the telic dependent type is well defined. The exact value of the dependent type is underspecified; the discourse context or what we typically think of as an adjunct modifier of the noun phrase can make clear what the purpose is.

(9.22) This is a good rock for skipping (throwing, carving, chiseling, etc.)

Sometimes the dependent type is specified by a common noun.

(9.23) This is a good skipping rock.

Or it can be specified via a familiar form of relative predication:

(9.24) This rock is good as a skipping stone.

All of these seem roughly equivalent in meaning. So it would seem that we should give roughly the same analysis to them. Using the relative predication as a guide, we see that in the case of (9.24) *good* is predicated of an aspect, as the analysis of relative predication would suggest. But what type is the aspect? It would seem that evaluative adjectives select for some sort of event associated with the aspect or the head noun. For example, contrast these two predications:

(9.25) a. Samantha is a mathematician.

b. Samantha is a really good mathematician.

There is a subtle shift in meaning between these two occurrences of *mathematician*.[3] To say that Samantha is a mathematician typically means that she has a certain profession; being a mathematician is like being a banker or a lawyer. On the other hand, to say that Samantha is a really good mathematician is to say that she is good at what mathematicians do—proving theorems, solving math problems and so on. While one can get the activity reading for (9.25a), I find it very difficult to get the ”profession” reading, the reading that Samantha has a particular profession, with (9.25b), and it's the same for *good lawyer*, *great writer*, *good salesman*. It's the same goes for evaluative adjectives at the negative end of the scale: to say of Fred that he's a lousy mathematician or a bad mathematician is just to say that he's bad at doing math, or at the limit that he can't do math. Similarly, to say that a rock is a good skipping stone is to say that it is good for the activity of skipping.

These readings have a lot in common with the event coercion readings for the aspectual verbs and *enjoy*. But they are also somewhat different. To be a good skipping stone, to be a good mathematician is to have a certain disposition. And dispositions are different than events. They are modal or quantificational properties; an object has a disposition to engage in an activity ϕ just in case in the appropriate circumstances the object normally does ϕ. That is, Samantha's being good at math means that in the relevant circumstances she normally does what mathematicians do and she does it well. Thus, the evaluative adjective modifies an event but that event is under the scope of a generic operator, which Asher and

[3] Thanks to Julie Hunter for discussions on this point.

Morreau (1991), Pelletier and Asher (1997) define using a weak conditional > and first order quantification. The dispositional reading comes from the requirements of the adjective and from the coercive capacity of the noun: there is a natural map from *mathematician* to the disposition to do math, and for every good mathematician x at time t, such a disposition must also exist for x at t (i.e. the map is everywhere defined). On the other hand, the map from good mathematicians to the activity of doing math need not be everywhere defined and so the coercion here is not guaranteed to be sound—some good mathematicians may only rarely engage in the activity. This observation predicts that event coercions over *mathematician* should not give rise to the activity of doing math, unless a lot of context is provided. A partial confirmation of this intuition comes from when we put *mathematician* as a direct object of an aspectual verb or *enjoy*, the event of doing mathematics is not salient, to say the very least.

(9.26) a. # Sam began that mathematician.

b. Sam enjoyed that mathematician.

c. Sam enjoyed mathematics at university.

It's very unclear what it is to begin a mathematician—unless it's a strange way of saying that Sam conceived a child who would grow up to be a mathematician. Similarly, the event of enjoying a mathematician entails the existence of some event (performances, cannibalism or sexual encounters come to mind) involving a mathematician, but only in a very particular context can it be an event of doing math (imagine that Sam is judging the performance of several mathematicians doing math).

On the other hand,

(9.27) Those were good oysters.

(9.28) That was a good cigarette.

have an event reading: there was an event of eating those oysters, and that event was good, or there was an event of smoking that cigarette and that event was good. Even the present tense

(9.29) These are good oysters.

has both a dispositional reading and an event reading. And this contrasts rather starkly with *that was/is a good mathematician*, which doesn't have the event reading

To sum up, evaluative adjectives select for an event or disposition type—whether an eventuality or a disposition is introduced within predications involving EC depends upon the type of the noun that is in the scope of the evaluative adjective or upon the content of an accompanying *as* or *for* adverbial phrase.[4] I will continue to use the dependent type ϵ, except that ϵ may now return event types or disposition types.[5] As we would predict when the dependent type is not specified, we get an anomalous predication, just as we do in the classic cases of coercion:

(9.30) a. # This rock is good as granite. (but contrast 'this rock is a good example of granite')

b. # This rock is good.

Before giving lexical entries and derivations, we need to address the question of subsectivity: does EVAL($\phi(x)$) entail $\phi(x)$? Given the difference in the readings noted in (9.25), we shouldn't in fact expect this inference from the activity reading to the profession reading. But what about from the evaluation of the activity to the activity reading itself? The positive evaluative adjectives certainly have that entailment: if you're a good mathematician, you're a mathematician in that you can do math. But what about the negative, evaluative adjectives like *bad, terrible* and *worthless*?

(9.31) That rock is a terrible skipping stone. In fact it's not a skipping stone at all.

If *terrible skipping stone* entailed that the rock was a skipping stone, then this discourse would be contradictory, or we would have to understand the second sentence as a correction by the speaker of his first claim. It looks like negative evaluative adjectives imply that the object may not have the capacity for the activity associated with its description.

Following **?** and Heim 19??, I will treat evaluative adjectives as functional relations over entities x and degrees with respect to a certain property of eventualities or dispositions involving x. So their lexical entry is something like this:

(9.32) $\lambda P \lambda x \lambda d \text{Eval-Adj}(P, x, d))\langle P : (\textsc{event} \sqcup \textsc{disposition}) \Rightarrow \textsc{t}, x : e, d : \textsc{degree}\rangle$

[4] The latter produces in TCL a relative predication involving the subject and makes the eventuality or disposition an aspect of the individual. But as I've noted, almost anything can be an aspect!

[5] Notice that the telic role doesn't really fit with these examples. It's really not the telos of a mathematician to do math. We don't think that people have ends other than themselves. Further, professions don't have telic roles, at least if we interpret the notion of telos as involving a function.

For evaluative adjectives, we can conceive of these degrees as positive or negative. The generalization seems to be that if the degree $d > 0$ then $\mathrm{Eval}(d, \phi)$ implies ϕ. When $d < 0$, then $\mathrm{Eval}(d, \phi)$ does not.

When P must combine with a predicate that is not a property of eventualities or dispositions, a type clash occurs and a version of EC introduces a variable with an appropriate dependent type so that the predication can take place. Transfer will introduce a term that can combine with P integrating the newly introduced variable. With this in mind, we can now proceed to a derivation for a common noun phrase like *good mathematician*. predicate

(9.33) $\lambda P \lambda x \lambda d \mathrm{Good}(P(x), x, d)[\lambda u \mathrm{Mathematician}(u)]\langle P : (\textsc{evt}\sqcup\textsc{disposition}) \Rightarrow \textsc{t}, x\colon e, d\colon \textsc{degree}, u\colon \textsc{agent}\rangle$

EC yields

(9.33a) $\lambda P \lambda x \lambda d \mathrm{Good}(P(x), x, d)[\lambda u\ \mathrm{Mathematician}(u)]\langle P : \textsc{disposition} \Rightarrow \textsc{t}, x\colon e, v\colon \epsilon(a1, \textsc{mathemat}$

Now with Transfer, we get:

(9.33b) $\lambda P \lambda x \lambda d \mathrm{Good}(P(x), x, d))[\lambda Q \lambda u \lambda z \phi_{\epsilon(\textsc{AGENT},\textsc{TYPE}(Q))}(z, u)[\lambda u \mathrm{Mathematician}(u)]]\langle P : \textsc{disposition} \Rightarrow \textsc{t}, x : \textsc{mathematician}, v\colon \epsilon(a1, \textsc{mathematician}), u\colon \textsc{mathematician}\rangle$

Several uses of Application and Accommodation yield:

(9.33c) $\lambda x \mathrm{Good}(\lambda z \phi_{\epsilon(\textsc{AGENT},\textsc{MATHEMATICIAN})}(z, x), x, d))\langle x : \textsc{agent}, v\colon \epsilon(\textsc{agent}, \textsc{mathematician})\rangle$

We now need to specify what $\phi_{\epsilon(\textsc{AGENT},\textsc{MATHEMATICIAN})}(z, x)$ is. It's a disposition, which I define in terms of a generic quantification over events. Thus, we need an extra component of Transfer for dispositions shifts the disposition typed variable over to one of event type and adds the requisite quantificational structure. The end result will be:

(9.24"') $\lambda x \forall v(\phi_{\epsilon(a1,\textsc{MATHEMATICIAN})}, x, v) > \mathrm{Good}(v, d)\langle x : \textsc{agent}, v\colon \textsc{event}\rangle$

Let us briefly evaluate the present proposal with what classic GL proposed. There was a problem with the classic GL qualia in giving the semantics of evaluative adjectives. In the verbal coercion cases, what was wanted was a dependent type that is an event type that is then instantiated in the predication. But here there should be no instantiation. A rock might be good for skipping even if it is never skipped. Thus, evaluative adjectives give us something like an intensional context with a generic quantification over events with degrees as an outcome of some sort (degrees of goodness). The present proposal solves this problem and provides an account that accords with intuitions.

Chapter 10

The Genitive Construction

10.1 The Genitive

Asher and Denis (2004) show that the genitive construction offers empirical evidence in favor of the more flexible typing system of TCL in contrast to the "classic" GL. They look in particular at the account proposed by (Vikner and Jensen,2002) that explicitly appeals to GL.

10.1.1 Previous Analyses of the Genitive

Let's start with some simple English examples of the genitive constructions:

(10.1) a. Bill's mother

b. Mary's ear

c. The girl's car

Interpreting phrases like (10.1) requires one to compute a *relation* between the two nominal referents that are introduced, respectively, by the specifier NP (aka *genitive NP* or *possessum*) and the head noun (aka the *possessor*) (see e.g. **?**). The problem is that this relation is often not directly specified by the grammar, and as a result these constructions give rise to many interpretations (i.e., many different relations can be inferred). For instance, depending on the context, *the girl's car* can be the car *owned*/*driven*/*dreamt about*/*designed*. . . by the girl.

? argue that GL's lexical semantics of the head noun is exploited during the interpretation of the genitive. According to them, the default relations found in the genitive constructions are provided by the different *qualia roles* associated with the head noun. More precisely, the idea is that the genitive NP, acting as the functor, is able to type-coerce the (monadic) denotation of head noun into a relation: crucially, this relation corresponds to one of *qualia* roles in the lexical entry of the noun. This approach has *prima facie* some empirical appeal, for one can indeed find examples corresponding to most of the *qualia* roles:

(10.2) a. FORMAL: The car's design

b. TELIC: Mary's book (i.e., the book *read by* Mary)

c. AGENTIVE: Bill's cake (i.e., the cake *cooked by* Bill)

However, this approach has a number of problems, as Asher and Denis (2004) argue. They claim that the telic is not part of the semantically licensed readings but rather a pragmatic one. If one goes along with their view, then this already indicates that the qualia aren't a homogeneous set of types semantically associated with a word. Asher and Denis's (2004) main criticism is that the *qualia* are simply not a reliable guide for inferring the genitive relation. We can't find any plausible example of a true CONSITUTIVE relation between a head and the genitive NP; there are lots of genitives that involve material constitution—e.g., *the car's metal, the sail's fabric, the computer's motherboard*, but in these examples the quale constitution is introduced by the genitive NP not the head. Trying to deal with this problem in the framework of GL has *de facto* an undesirable consequence, since it supposes either a relaxation of the Head Principle or an *ad hoc* redefinition of the CONST role; J&V opt for that latter alternative proposing that the CONST *quale* be interpreted as the *part-whole* relation.

Because of this, J&V are often led to adopt an extremely liberal view of the *qualia*. For instance, they assume that the same *qualia* role (namely, the CONST role) is responsible for explaning both the interpretation of genitive like *The girl's nose* (clearly a case of *part-whole* relation) and that of *The girl's team* (clearly, a *set-membership*). This makes the CONST role very unclear. For one thing, there is first a directionality problem: the *part-whole* relation is between the head N and the genitive NP, while the *set-membership* relation goes the other way around,in potential violation of the head principle. That is the girl is constituted in part by her nose in one case, but it is the team that is constituted in part by the girl. This principle is used by Asher and Pustejovsky (2003) to avoid spurious ambiguities when exploiting or introducing complex types. It's also part of TCL.

There is a more crucial problem: these two relations *part-whole* and *set-membership* are rather different in formal terms (e.g., the former is clearly transitive, while the latter is not). Under the analysis of Asher and Denis (2004), nouns like *nose* and *team* having different intrinsic types (a *body-part* and a *group*, respectively) and therefore dependent types; and so, they will naturally give rise to different types of relations.

Another argument against *qualia* comes from morphologically rich languages that have grammaticalized certain genitive relations. Crucially, none of these languages have grammaticalized *qualia* roles therein. In Basque for instance, there are two genetive postpositions, namely an unmmarked *-(a)ren* suffix and a marked *-ko* suffix that specifically encodes (spatial) *localization* and which thus requires that the type of the object denoted by the nominal in this genitive case be a location (see **?**). Below is a minimal pair from Basque that gives an illustration of the two genitives:

(10.3) a. liburu**ko** argazkia
book-ko photo-def
'the photo from/in the book'

b. liburu**aren** argazkia
book-aren photo-def
'the photo of the book'

Crucially, the relation of localization whether temporal or spatial is not part of the *qualia*. However, if we broaden the set of categories which might govern dependent types, say to Kant's substance accident dichotomy (also Aristotle's), then many accidents can play this role. Or more specifically, we may consider the forms of intuition, space and time, to play important roles that allow us to locate objects. Another important function here of location is that of Backgrounding, which leads us to links between the lexicon and discourse structure. It might very well be that the cognitive semanticists like Talmy have isolated at the lexical level discourse factors, perhaps inaptly named figure and ground. These categories seem to isolate something important in the selectional restrictions of certain relational prepositions, for example (see Asic 2005).

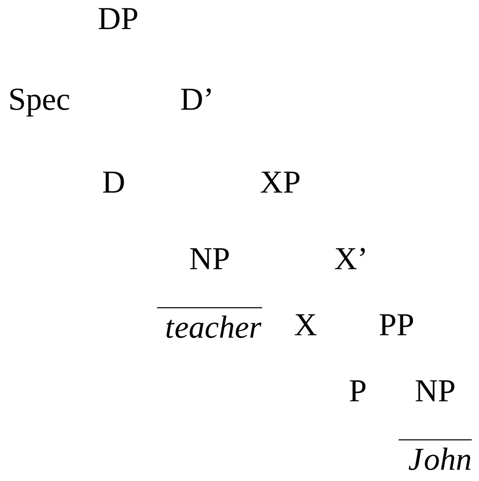
There are several other problems with qualia as suitable categories for the analysis of the genitive. For instance, the *qualia* by themselves fail to provide the relation *own* (i.e., material possession) that one finds (preferred) with most physical objects (e.g., *John's car*) in many different languages.

These different points seem to make it clear that the *qualia* roles are in a way arbitrary for something like the genitive construction; they don't seem to have any privileged epistemic status. The more general notion of dependent type coded in the $\otimes$ formalism seems the right way to go at least for adjectives.

10.2 Genitives within TCL

Asher and Denis (2004) assumed an Abney like syntax for the genitive construction with a DP analysis (cf. Abney (1987); see also Asher (1993)) in which the construction is headed by an empty functional D head, which assigns (genitive) case to the Spec, DP. Asher and Denis (2004) borrow den Dikken's (1998) syntactic analysis of the genitive, which assumes a small clause structure in which the possessor is the complement of the predicate that is predicated of the possessum.

(10.4)



This predicative structure is then selected by a determiner. At this point, various "spell-out" options are possible to predict the various word orders. In the case of the prenominal Saxon genitive, the P complement raises all the way up to Spec, DP.

The following derivation will assume the following semantic denotations for the predicative head *X* and the possessor DP *John's*:

(10.5) $[\![\mathrm{X}\,]\!] = \lambda\mathcal{P}\lambda R\lambda u\lambda e\mathcal{P}[\lambda v[R(u, v, e)]], \langle u\colon e, v\colon e, e\colon \textsc{eventuality}\rangle$

(10.6) $[\![[_{DP}\ \text{John's}\,]\!] = \lambda P[P(\mathbf{john})], \langle \mathbf{john}\colon \textsc{human}\rangle$

From these two denotations, we can now derive the denotation for the *X'*:

(10.7) a. By syntax:
$[\![[_{X'}\text{ John's}]]\!] = \lambda\mathcal{P}\lambda R\lambda u\lambda e\mathcal{P}[\lambda v[R(u,v,e)]]\ [\lambda Q[Q(\mathbf{john})]],\ \langle u\colon e, v\colon e, e\colon \textsc{eventuality}\mathbf{john}\colon \textsc{hum}$

b. Since $\textsc{human}\sqcap e = \textsc{human}$, we revise the typing context of the functor via *Type Accomodation* as follows: $\lambda\mathcal{P}\lambda R\lambda u\lambda e\mathcal{P}[\lambda v[R(u,v,e)]]\ [\lambda P[P(\mathbf{john})]]$, $\langle u\colon e, v\colon \textsc{human}, e\colon \textsc{eventuality}, \mathbf{john}\colon \textsc{human}\rangle$

c. Finally, by *Application* twice:
$[\![[_{X'}\text{ John's}]]\!] = \lambda R\lambda u\lambda e[R(u,\mathbf{john},e)], \langle u\colon e, \mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle$

Our derivation has shown that however we understand the head noun, we will need to find an eventuality associated with the head noun. Thus, I hypothesize that the genitive also licenses a form of EC converting when needed the denotation of the event noun into a parametrized, eventuality dependent type.[1] Our dependent parameterized types that help model data concerning control verbs and logical metonymy also find a use with genitive constructions. Asher and Denis's (2004) account, however differs from other accounts in the use of eventualities, which is typically what $\epsilon(\textsc{arg}1, \textsc{arg}2)$ returns; previous accounts accounts only use a relation. The use of eventualities has several advantages. For one thing, they will allow us to make the treatment of genitives very similar to that of logical metonymy, as we shall see. Second, eventualities are needed independently to handle temporal NPs (cf. Enç (1981) etc.). Thus, consider a sentence like:

(10.8) John's wife went to Yale.

Interpreting this sentence properly assumes that we can temporarily relate the *state* described by the genitive (i.e., that some individual is the wife of John) to the eventuality described by the whole sentence. Interestingly, this sentence has a preferred reading wherein *John's wife* is interpreted at the utterance time; that is, wherein the state comes *after* the main eventuality. I am at present unsure whether this temporal ”value loading” is a general feature of genitive constructions.

There are several cases that Asher and Denis investigate. I'll go through these derivations briefly.

[1] Asher and Denis (2004) used an ⊗ types to represent among other things what GL describes with *qualia* . The ⊗ types were similar to •; the dependent types used here are much simpler and also more general.

The inherent “relation” case

(10.9) a. John’s teacher (deverbal)

b. John’s friend (relational)

c. John’s laziness (deadjectival)

The derivation for *John’s teacher* is as follows:

(10.10) a. The denotation for *teacher* is:
$\lambda y\lambda x\lambda e[teach(x, y, e)], \langle x, y\colon \textsc{human}, e\colon \textsc{event}\rangle$

b. By syntax, we apply the denotation for *X’* to that of *teacher*:
$\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)], \langle u\colon e, \mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle$
$[\lambda y\lambda x\lambda e'[teach(x, y, e)], \langle x, y\colon \textsc{human}, e'\colon \textsc{event}\rangle]$

c. Various *Type Accomodation*s take place (i.e., on *R* and *u*, and *e*) allowing us to merge the contexts:
$\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)]\ [\lambda y\lambda x\lambda e'[teach(x, y, e)],$
$\langle u, \mathbf{john}\colon \textsc{human}, e\colon \textsc{event}x, y\colon \textsc{human}, e'\colon \textsc{event}\rangle]$

d. *Application* four times gives the following:
$\lambda u\lambda e[teach(u, \mathbf{john}, e)], \langle u, \mathbf{john}\colon \textsc{human}, e\colon \textsc{event}\rangle$

e. Finally, by the *Maximal Projection Rule*:[2]
$\lambda u\exists e[teach(u, \mathbf{john}, e)], \langle u, \mathbf{john}\colon \textsc{human}, e\colon \textsc{event}\rangle$

The derivation of denotation of *John’s friend* will follow exactly the same steps; the only difference is that the denotation of *friend* introduces a state, rather than an event. Below is the final result of the derivation:

(10.11) $[\![\text{John's friend}]\!] = \lambda u\exists e[friend\_of(u, \mathbf{john}, s)], \langle u, \mathbf{john}\colon \textsc{human}, e\colon \textsc{state}\rangle$

10.2.1 States

Many examples of the genitive construction make reference to a state of an individual, e.g., *John’s laziness*. Asher and Denis assume that the denotation of the deadjectival is the following:

(10.12) $[\![\text{laziness}]\!] = \lambda x\lambda s[lazy(x, s)], \langle x\colon \textsc{animate}, s\colon \textsc{state}\rangle$

[2] This is a rule taken from the construction procedure of Asher (1993) for nominals.

This denotation is not relational and consequently will not be able to combine with the denotation given above for *X*. This suggests that the genitive relation relation is not always relational as assumed by most existing acounts. And one has to posit another possible denotation for it:

(10.13) $[\![\mathrm{X}_s]\!] = \lambda\mathcal{P}\lambda S\,\lambda s\mathcal{P}[\lambda v[S\,(v, s)]], \langle v{:}\, e, e{:}\, \textsc{state}\rangle$

Instead of positing two different denotations for *X*, one could actually assume, following Asher and Denis (2005), that*X* has a complex *disjuntive* type.

Combining this new denotation for *X* with the DP denotation yields the following alternative denotation for the *X' John's*:

(10.14) $[\![[_{X'}\ \text{John's}]]\!] = \lambda S\,\lambda s[S\,(\mathbf{john}, s)], \langle \mathbf{john}{:}\, \textsc{human}, e{:}\, \textsc{state}\rangle$

Finally, this *X'* can now combine with the NP *laziness*, yileding the following meaning:

(10.15) $[\![[_{X'}\ \text{John's}]]\!] = \lambda s[\mathit{lazy}(\mathbf{john}, s)], \langle \mathbf{john}{:}\, \textsc{human}, e{:}\, \textsc{state}\rangle$

10.2.2 Interpretations Exploiting a Dependent Type of the Head NP

Yet another case of the genitive construction in our analysis is where the interpretation of the relation between the possessive DP and the head NP comes from a dependent type associated with the head NP. An example of this is:

(10.16) John's team

(10.17) a. The denotation for *team,* a unary predicate, is as follows:
$[\![\text{team}]\!] = \lambda x[team(x)], \langle x\colon \textsc{group} \otimes \epsilon(\textsc{arg}1, \textsc{arg}2)\rangle$

b. Combining $[\![[_{X'} \text{ John's}]]\!]$ with $[\![\text{team}]\!]$ leads to a type clash:
$\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)], \langle u\colon e, \mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle$
$[\lambda x[team(x)], \langle x\colon \textsc{group})\rangle]$

c. So by EC, we convert *x*'s type to one involving an associated, eventuality dependent type:
$\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)], \langle u\colon e, \mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle$
$[\lambda x[team(x)], y\colon \textsc{group}, x\colon \epsilon(\textsc{a}1, \textsc{group}))\rangle$

d. By *Transfer,*
$\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)], \langle u\colon e, \mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle$
$[\lambda x\lambda y\lambda v[team(y) \wedge \phi_{\epsilon(\textsc{arg}1,\textsc{arg}2)}(x, v, y)], \langle y\colon \textsc{group}$
$x\colon \epsilon(\textsc{a}1, \textsc{group}))\rangle]$

e. Assuming that we have the defeasible rule:
$\top > \epsilon(\textsc{a}1, \textsc{group}) := \textsc{membership}(\textsc{a}1, \textsc{group})$
We can readjust the type of x to $\textsc{membership}(\textsc{arg}1,\ \textsc{arg}2)$ and so fill in $\phi_{\epsilon(\textsc{arg}1,\textsc{arg}2)}(x, v, y)$

f. So we get:
$\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)], \langle u\colon e, \mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle$
$[\lambda x\lambda y\lambda v[team(y) \wedge has\_member(x, v, y)], \langle y\colon \textsc{group} \otimes \epsilon(\textsc{arg}1,\ \textsc{arg}2),$
$x\colon \epsilon(\textsc{arg}1,\ \textsc{arg}2))\rangle]$

g. By *Type Accomodation, Context Merge,* and *Application*:
$\lambda u\lambda e[team(u) \wedge has\_member(u, \mathbf{john}, e)], \langle u\colon \textsc{group}, \mathbf{john}\colon \textsc{human}, e\colon \textsc{state}\rangle$

h. Since we are now at a maximal projection, we close off all the variables in surplus (here, only the e variable): $\lambda u\exists e[team(u) \wedge has\_member(u, \mathbf{john}, e)], \langle u\colon \textsc{group}, \mathbf{john}\colon \textsc{human}, e\colon \textsc{state}\rangle$

The default possessive/control relation

The ownership/control reading is available in most cases of the genitive; the way we propose to model this is that the genitive construction always makes this a possible output of $\epsilon(a1, a1)$. This makes our type function yield potentially a number of values; the set of all these possible values will be denoted by $Choices(\epsilon(a1, a1))$. Some of these values may be preferred in certain contexts, something that we can encode with our type identity statements.

(10.18) $\lambda\mathcal{P}\lambda R\lambda u\lambda e\mathcal{P}[\lambda v[R(u, v, e)]], c(\langle u\colon e\otimes\epsilon(a1, a2)) \rightarrow \textsc{poss}(a1, a2) \in Choices\epsilon(a1, a2)$

Some relational nouns (e.g., *picture*) can have their argument closed off and yield an ownership reading. Crucially then, this default has somehow to come from the construction (i.e., from the *X*), since this reading is possible independently of the existence or not of a dependent type for the head N.

Let's now see how our account handles a genitive like *John's dog*. For simplicity, we assume here that this NP has a simple type.

(10.19) a. $\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)], \langle u\colon e,$
$\mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle \;\; [\lambda x[dog(x)], \langle x\colon \textsc{animal}\rangle]$

b. We can now use EC:
$\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)], \langle /u\colon e,$
$\mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle \;\; [\lambda x[dog(x)], \langle x : \epsilon(a1, \textsc{animal}, y\colon \textsc{animal})\rangle]$

c. Assuming no other dependent type is available to further specify $\epsilon(\textsc{arg}1\textsc{arg}2)$, our type identity rules force us to specify the dependent type to $\textsc{poss}(\textsc{arg}1\textsc{arg}2)$, and we have:
$\lambda R\lambda u\lambda e[R(u, \mathbf{john}, e)], \langle /u\colon e,$
$\mathbf{john}\colon \textsc{human}, e\colon \textsc{eventuality}\rangle \;\; [\lambda x[dog(x)], \langle x : \textsc{poss}(a1, \textsc{animal}, y\colon \textsc{animal})\rangle]$

The rest of the derivation now proceeds as before (10.17c-g), finally yielding:

(10.20) $\lambda u\exists e[dog(u) \wedge owned\_by(u, \mathbf{john}, e)], \langle u\colon \textsc{animal}, \mathbf{john}\colon \textsc{human}, e\colon \textsc{state}\rangle$

Interpretations using the (dependent) type of the possessor

There are examples of the genitive construction where

(10.21) a. The painter's sky

b. The artist's object

c. The janitor's room

These are the equivalents of verbal coercion examples we saw earlier like the following:

(10.22) a. The janitor/The fumigator started (on) the room.

b. The architect/The workers started (on) the house.

These can be given by the default rules on ϵ. The question is how do we get more specific interpretations for the relational abstract *R* when those are given by the DP in SPEC position. It should be noted that some of these are uninterpretable—for instance, *the child's woman*, except as cases of possession, even though *child* is relational and arguably *woman* is not. Other putative examples, however, are *the artist's object*, which we think can mean the object created by the artist. *Object* is not plausibly relational; when it occurs in the genitive construction with another noun that introduces a λ-bound variable of simple type, then the only relation that seems to hold between the two variables is POSS. But in this case, we get a different interpretation. How is this possible? What needs to happen, we think, is that somehow the typing information from the DP in SPEC has to affect the interpretation of *R* in the possessive determiner; and the only way we can do this in our framework is to relax the order in which functional applications are made, as argued for in Asher (1993). At this point our type logic becomes polymorphic. We don't go into details but essentially this means that we can ”freeze” the application of the determiner meaning to the head NP and go ahead and combine the meaning of the DP in SPEC with that result. ***this we've done already*** This would give us the more specific typing information from the DP onto the relational abstract. And this in turn will allow us to construct a more specific $\otimes$-type via $\otimes$-Introduction for the λ-bound variable in the NP. And then we can proceed as above to get a relational meaning for the NP via $\otimes$-Exploitation^R. Our final logical form for (10.23) is (10.23'):

(10.23) The artist's object

(10.23') $\lambda S(\exists! x(artist(x) \wedge \exists! y(object(y) \wedge created - by(y, x) \wedge S(y)))$

10.2.3 Discourse Context and the Genitive

I now return to treat examples where the discourse context enables us to specify the appropriate eventuality and relation in the genitive construction. Here is the example from Asher and Denis (2004) I discussed earlier, repeated below.

(1.40) a. All the children were drawing fish.

b. Suzie's salmon was blue.

Here we need to draw on a theory of discourse structure like SDRT, which would posit that (1.40a) and (1.40b) stand in an *Elaboration* relation, or even *Instance* (Asher (1993)). But this will be so, only if Suzie's salmon is interpreted in a very particular way; namely, the salmon is the fish that Suzie is drawing. Without this link, we cannot make any clear discourse link between the two sentences, threatening the discourse's coherence. So how does this work? *salmon* introduces a variable whose subtype is that of the variable introduced *fish*—i.e., $\textsc{salmon} \sqsubseteq \textsc{fish}$. But *fish* has had it's type changed in the context because it is an argument of *draw*. *draw* coerces *fish* into a dependent type though here we don't have ϵ but rather a different dependent types that takes types of objects to types of pictures of those objects—i.e. $\textsc{picture-of}(\textsc{fish})$. SDRT's principle of Maximize Discourse Coherence says that we must maximize discourse coherence in resolving underspecifications, but here we have to interpret this in an interesting way vis a vis dependent types: get whatever complex type involving the original type you can so as to maxmize discourse coherence, which involves, among other things, computing a discourse relation between (1.40a) and (1.40b). Several inferences are needed to compute Elaboration or Instance here. First, *salmon* must inherit the dependent type associated with *fish* in this context, this inheritance being crucial to computing the discourse relation of *Instance*. This complex type now helps specify the relation between Suzie and her salmon, but by itself it doesn't suffice, because Suzie is not the picture of the salmon. We could infer by default that Suzie owns the picture, using our default rules for the genitive and for ϵ. But the discourse structure suggests another specification of ϵ that is independently plausible. What sort of event might hold of an agent and a picture? Naturally, one thinks of the event of drawing; that is we specify $\epsilon(\textsc{agent}, \textsc{picture-of}(\textsc{fish}) = \textsc{draw}(\textsc{agent}, \textsc{picture-of}(\textsc{fish})$. If we also make the additional inference that Suzie is one of the children, then the discourse relation of Instance or Elaboration can be inferred.

Chapter 11

More Types and Types of Predication

In this chapter, I consider some extensions of the TCL system to several phenomena involving lexical semantics and predication. The first concerns presupposition, the second fictional objects and predications to them, the third metaphorical predication, the fourth nominalisation. Finally, I say a few words about vagueness to indicate why it is a problem separate from predication, which is, from the perspective of this book, a matter of constructing logical form.

11.1 Dynamic Types and Presupposition

Although presuppositions and proferred content cannot be assimilated completely into the type system, the type system can help systematize presuppositional predication as well as ordinary (proferred content) predication. Many presuppositions come from particular lexical elements, the so called ”presupposition triggers”. It turns out that there are a lot of presupposition triggers. Most change of state verbs have lexical presuppositions, which are the preconditions that have to obtain before the actions or transitions they denote can take place. Commonplace verbs like *buy*, *sell*, *loan*, *borrow*, *heal*, all have such presuppositions. Their preconditions e.g. of ownership or physical possession obey all the classical tests for presuppositions; they take wide scope over various operators like negation or the operator associated with a question. Such change of state verbs also have *post-conditions*, the conditions that obtain after the event denoted by the verb has taken place. Post-conditions correspond to proferred content, while pre-conditions are a generalization of the standard notion of presupposition. Perhaps all change of state verbs have such pre- and post-conditions.

There's a question as to how a lexical system ought to represent such presuppositions. We have two choices: we can put this information into the lexical entry's contribution to the logical form of the clause in which it occurs as is traditionally the case and is carried out in the work of Kamp and Rossdeutscher Kamp and Rossdeutscher (1994b,a) in their detailed work on the German verb *heilen* (to heal), or we can attempt to stick such information in the type system. While the first option is pretty well understood, it does lead to a vast amount of redundancy. We'd like to say that verbs like *buy, acquire, purchase* have the same ownership pre-and post conditions, while *buy, etc.* and *steal* have the same physical possession pre- and post-conditions. If the presuppositions or preconditions and post-conditions are entered for each lexical entry we miss important generalizations about the presuppositions as well as the post conditions of verb classes. This would suggest that such pre- and post-conditions are to be attached somehow to a general type like CHANGE OF STATE VERB and then inherited by the various subtypes.[1]

To specify the type CHANGE OF STATE VERB, for instance, and of presuppositions more generally, we have two options. The first is to make use of the pair construction again, but with a different aim in mind. The pair will have as first element the type of the presupposition or the precondition, while the second element will contain the type of the postcondition or ”proferred” content. Alternatively but in closely related fashion,[2] we might think of change of state verbs as defining a kind of map from preconditions to postconditions. The pair gives the initial and final point of the map. I'll use the notation $[\alpha,\beta]$ to indicate the type of a change of state verb, or, more generally, the type of an element with presupposed and proferred content; α gives the type of the precondition, while β gives the type of the postcondition. I will call $[\alpha,\beta]$ a *dynamic type.* Using dependent types, we can get much more precise descriptions of what the presuppositions of particular change of state verbs are, since dependent types, when their arguments are filled in, pick out subspecies of propositions in our intensional setting. For instance, let's think again about all those verbs of acquisition whose preconditions specify a lack of ownership of some argument and whose post-conditions specify ownership. On a first approximation, we might come up with this figure, with several parameters

[1] Levin (1993) suggests such an approach.

[2] In a linear system of types, the pair and the arrow can be interdefined. See e.g., Aehling and Schwichtenberg (2000).

(specifiying the relevant arguments of the verb).

(11.1) GET-OWNERSHIP-OF: THEME $\Rightarrow$ (AGENT $\Rightarrow$ [$\neg$OWN(AGENT, THEME), OWN(AGENT, THEME)])

Notice that negation is a constructor here of types, which makes sense given that I have associated a type with each word root. I have also used THEME as a type, which is somewhat strange. It is really there to suggest that the functional types first and second arguments also figure as parameters in the dependent type that is the last term of the functional types. I will come back to this point in a minute, but this at least serves to get across the idea. The type structure in (11.1 fits best with the verb *acquire*. But there are verbs related to *acquire* whose lexical entry also has this type—*steal, buy, borrow, take over, etc.* These latter verbs have subtypes of(11.1. They will specify what kind of ownership is involved and they may also specify further elements about the transition

There is an interesting sort of coercion with acquisition verbs.[3] Real-estate agents, used car salesman and stockbrokers will say,

(11.2) I sold three houses (ten cars, 1 million shares...) this week.

even though they did not in fact own in either a physical or legal or any other sense the objects that they sold. In fact what happened is that these agents do something that results in someone else's change of ownership of whatever the theme of *sell* is. These contexts make salient a map from agents to a collection of people or companies they represent, which allows us to reinterpret the verb in the appropriate way. As usual, we will suppose a particular dependent type REPRESENTATIVE from agents to agents or groups of agents. The context will license a coercion licensing the substitution in (11.1) of REPRESENTATIVE(AGENT) for AGENT. Transfer will then make the appropriate change at the level of logical form.

Another class of presupposition triggers has to do with factives and referential expressions, both of which have common elements of content between the presuppositions and post-conditions. Following Heim (1982) (1983), people have taken definite noun phrases and perhaps more generally referential expressions to give rise to presuppositions of familiarity. Using the discourse framework of SDRT and the analysis of presuppositions provided by Asher and Lascarides (1998), Hunter and Asher (2005) show how to give the outlines of a general presuppositional account of referential expressions—at least including indexicals, definite descriptions and proper names. Complex demonstratives would be an easy extension of that approach. Briefly, this view assigns a logical form to such expressions that looks something like this:

[3] Thanks to Julie Hunter for bringing this up.

(11.3) $\lambda P\{\pi_p : \mathcal{F}(\exists x\phi(x)), \pi_a : P(x)\}$

π_p is a label denoting the presupposed constituent, while π_a is a label denoting the asserted constituent or constituent containing the proferred content. $\mathcal{F}$ is a particular sort of modal operator that provides instructions for where the presuppostion should go in the finished logical form. These are a form of ”tree” or more generally ”graph” operators, which in general tell us what accessible points are relevant to interpretation. The novelty of this approach is that the graph operators tell us which parts of the discourse context are relevant for the integration of the presupposed information into a final logical form. Such a logical form is assumed to contain a representation of the ”outermost” SDRS which is the speech context. Thus, indexicals like *I* will have a presupposed context prefixed by the operator $\Uparrow_{K_0}$ to indicate that the presupposed information must go in the outermost discourse context. Different referential expressions will have different graph operators; definite descriptions may have several alternative operators to represent attributive and referential readings Donnellan (1966). Sometimes these operators express absolute requirements on the attachment of presuppositions (as in the case of indexicals), but sometimes they may express also a preference for a high or low attachment in the SDRS for the discourse (Asher and Lascarides 1998), which is a generalization of the notion of global or local satisfaction and accommodation as developed in Van der Sandt (1992), Geurts (1999), and others.[4]

To express the general type of a referential determiner, we need to capture the idea that there is a common argument to both presupposition and proferred content. We can suppose an indexing system on types with the following interpretation:

- $[...\alpha_i..., ...\beta_i...]$ have a pair of objects (a, b) as inhabitant iff whatever inhabitant of α is chosen to determine a, the very same inhabitant must be chosen as the inhabitant of β in the calculation of b.

The general type of referential determiners looks then something like this:

(11.4) $(e \Rightarrow \textsc{t})_1 \Rightarrow ((e \Rightarrow \textsc{t})_2 \Rightarrow [\textsc{modal}(\textsc{exists}\ e_1(e \Rightarrow \textsc{t})_1(e_1)), (e \Rightarrow \textsc{t})_2(e_1))])$

Using indexed dynamic types, we can now rewrite the type of acquisition verbs properly.[5]

[4] Although the account of presupposition offered in Asher and Lascarides (1998) owes much to Beaver’s account of presupposition in Beaver(1997) as well.

[5] Indexing, it should be noted allows for repetition, and so it threatens the linearity of the type system.

We can use the notion of co-indexed types within dependent types to define the general lexical entries for factive expressions. Consider the class of factive verbs that take propositional type complements (which we will take to be realized not only by *that* clauses but also by infinitival phrases). We will use both dependent and dynamic types to express them:

(11.5) a. $\textsc{prop}_1 \Rightarrow (e_1 \Rightarrow (e_2 \Rightarrow [\textsc{modal}\ (\textsc{prop}_1), FV(e_1, e_2, \textsc{prop}_1)]))$ with its standard translation into the language of truth conditions where π_p labels the presuppositional content while π_a labels the proferred or post condition content:

b. $\lambda P \lambda e \lambda y(\pi_p : \mathcal{F}(P), \pi_a : FV(e, y, P)$

What does all this have to do with pre and post conditions? We have to understand this in a relatively metalinguistic way for referential expressions and factive verbs. Presupposed content of referential must typically be attached or integrated into the discourse content prior to the proferred content because of the variables within the proferred content that depend for their assignments on the presupposed content. For factives, the connection is even vaguer; the presupposition has a scope behavior that is distinct from the proferred content in view of the modal operator attached to it. The presupposition is ”prior” in the sense that it can move up in the discourse content, whereas the non modalized proferred content cannot. But nothing in this picture says that presuppositions cannot be informative, contra the received wisdom about presuppositions.[6]

This would suggest that we should see variation in the lexicon between factive and non factive verbs versus the presuppositions of change of state verbs. Indeed as far as I know the presuppositions of the various change of state verbs survive across languages, and that should be expected in terms of the metaphysical analysis of such events through the device of dynamic types. Given the analysis in terms of coindexed types of the class of referential terms, we should expect similar behavior across languages, though we can expect perhaps variation in the types of modal operators involved, as Schlenker (2003) and Hunter and Asher (2005) have argued for. But for factive verbs, we should expect lexical alternations and variations across languages, since here it is the least clear why there should be a dependence of the proferred content on presupposed content. One case is perhaps the emotive verbs—verbs or intransitive verb phrases like *regret, be sad, be*

[6] Mandy Simons and Philippe Schlenker in recent papers have argued that standard theories of presupposition are wedded to the *uninformativeness* of presuppositions. But this is not the case with the view of presuppositions as understood *a la* Asher and Lascarides (1998), where Accommodation is modelled via the discourse relation Background.

happy, ... You can't have an emotion about some state of affairs, if the state of affairs doesn't obtain. That is, we might take those emotions as expressed via emotive verbs to be always *de re* or at least *de situ*. For the epistemic verbs, we do see many alternations: *know* versus *believe* (or *justifiably believe*), *inform* versus *assert* or *say*,

We can use such presuppositional types to help understand how narratives work by locally satisfying the presupppositions of each transition clause within the post state provided by the previous verb. To give an example from a particular genre (these are actions)

(11.6) You turn right at the light. You go two blocks. Another right turn and you're there.

11.2 Fiction and Fictional Objects

Fiction and fictional objects present a challenge to a theory of predication. Things that aren't possible according to the standard typing system become possible in fiction.

Is fictional predication a kind of coercion? In cases of coercion we have seen that the basic types of terms are not changed but a local adjustment is made—and terms are added—to resolve the type conflict in the predication. Predication involving fictional objects is different. We really need to shift the types of individuals. For instance, goats can become creatures that read and talk in fairy tales. Trees literally talk in fairy tales or fantasy novels like *The Lord of the the Rings*. These predications should be precluded by normal type constraints, and we know that there are no reference preserving maps (in this world) from goats to talking agents or from trees to talking agents. Thus, something different from the sort of coercion processes that we have seen in the last two chapters is at work in predications to fictional objects.

What are fictional objects anyway? Outside of a Meinongian framework where there are necessarily nonexistent entities, fictional objects are puzzling. Fictional objects can't be real objects, or objects that could exist in some possible world but not the actual one. Fictional objects are thus clearly different from possibilia; the sister I might have had is not the same type as a sister I made up to amuse people or to make excuses the way Algernon does with his made up friend Bunberry in Oscar Wilde's *the Importance of Being Ernest*. The type of fictional objects has no inhabitants, but is a subtype of e. Furthermore, there are many subtypes of the type of fictional objects. Fictional horses are quite different from fictional cats or fictional sisters or fictional friends. However we analyse predication of fictional objects, our account must allow that fictional objects have, in some sense, many of the qualities of real objects of the same type. Perhaps fictional introduces a dependent type mapping actually realized types into unrealizable counterpart types with the proviso at the very least that the generic properties we associate with inhabitants of the actually realized types carry over to their fictional counterparts, however fictional predication is to be analysed. Thus, if Sherlock Holmes is a fictional object he is also a fictional man, and he has (in fiction) at least by default the qualities generically associated with men. He has two legs, two arms, wears a coat when it's cold outside, sleeps at night and works during the day.

Unlike metaphorical predication or loose predication, in fictional predication the objects literally have the properties they are predicated to have. That is, in *the Lord of the Rings* the trees actually speak to one another, just as you and I would to our respective conversational partners. Another important difference is that in fiction objects come to acquire the properties they do in a matter of stipulation, at least in the fiction in which they are introduced. They have the properties they do in virtue of the predication of those properties to them in the story.[7] Thus, predication to fictional objects must be marked in a particular way, as the evaluation of the logical form with those predications will proceed in a quite different way from the way logical forms of non fictional discourse is evaluated.[8]

Let us look a bit more closely at the way fictional discourse updates the discourse context contrasting it with discourse update in nonfictional discourse. In nonfictional discourse, updates proceed in two ways in dynamic semantics. An information context is understood as a set of world assignment pairs, although it is necessary to postulate more complex contexts to handle updates with modals or plural quantifiers (Asher and McCready 2007, Van Rooij 2005, Asher and Wang 2003, Wang 2005). Existential quantifiers in logical form will reset or extend the assignment elements of the context, while other elements of the logical form will eliminate worlds from the context. Thus, an update in non fictional discourse adds information by eliminating possibilities and by resetting assignments to variables. Obviously, we don't want updates in fictional discourse to perform these operations on *real* possibilities and we don't want existential quantifiers in the logical form of a fictional discourse to make assignments to variables of *objects* since

[7] There can be even revision of those properties, so it may be not completely clear from a text whether a fictional object has property P or not.

[8] Here we must be careful to distinguish fictional discourse from discourse about fiction!

we've said that there aren't any fictional objects in any possible world.

But the problem is that fictional discourse works an awful lot like non fictional discourse from the dynamic semantic viewpoint. We need to have devices permitting long anaphoric links between fictional noun phrases and pronouns. We learn more about fictional characters as the story proceeds and this learning procedure seems very similar to the elimination of possibilities in the course of non fictional discourse update. So what can we do? It seems sensible in these cases to suppose a set of pseudo possibilities, possibilities for this fictional story along with assignments to fictional objects in these pseudo possibilities, and update in the standard way of dynamic semantics.[9] Truth for what is said about fictional objects just concerns what is established relative to all of the pseudo possibilities for a given fictional context. (It may be the case that a fictional context may be constructed from more than one fictional work; I leave that delicate question aside here, as it far outruns our questions concerning types).

To shift to pseudo possibilities and these fictional objects that aren't possibilia, we need the device of a type shift that percolates "all the way up" to the top of the discourse or perhaps the other way round. Thus we suppose an "arrow" from any type TYPE to its fictional counterpart.

- TYPE ⟶ FICTIONAL(TYPE)

This type shift now changes all the denotations to pseudo-denotations in pseudo possibilities. Interestingly, this type shift leaves the associated generics unchanged—and thus the fine grained conceptual content of fictional types remains similar to the non fictional parts. It is just that all of these generic statements are interpreted relative to the "pseudo" domain.

11.3 Metaphorical Predication

Metaphorical predication is another form of predication. Like loose predication, metaphorical predication doesn't imply that the property or relation involved applies to its terms literally. There are some similarities between metaphor and loose talk as I defined it above. When Shakespeare says

(11.7) Juliet is the sun.

[9] This seems to be the formal counterpart to the popular "pretense" theory of fictional objects.

one can analyze that in the manner of loose predication: Juliet is more like the sun than anything else in the comparison class of heavenly bodies. The predicate gives the alternatives to the predication just as we saw with loose talk. Black (1962) proposed a theory vaguely along such lines, and more recently Asher and Lascarides (1999) proposed an analysis of metaphorical predication in which they tried to put some more meat on the notion of closest fit. They looked in particular at metaphorical interpretations of change of location verbs. They argued that in effect each type has a definition in terms of genus and differentiae. For change of location verbs like *enter*, the genus is the type CHANGE-OF-LOCATION-VERB, which is a subtype of CHANGE-OF-STATE-VERB. The differentiae that distinguish *come* from *go*, another change of location verb concerns the particular pre- and post-conditions for each verb—the precondition gives the starting point of the motion, while post-conditions include at least the end point of the motion and may specify something about the trajectory from the starting point as well Asher and Sablayrolles (1995). Thus for a somewhat leaden metaphorical predication like

(11.8) Nicholas went into a blue funk.

Asher and Lascarides would claim that the differentiae of *go* persist—namely that Nicholas was not in a blue funk prior to the metaphorical change of motion, but was in one after the change. What changes is that the genus is in fact reinterpreted replacing the type LOCATION with some sort of qualitative space pertaining to emotions.

The problem with this view is that it does not do well with many metaphors, especially creative and powerful ones. The sun's *differentiae* with respect to other heavenly bodies in Shakespeare's language isn't known to modern readers who still find the metaphor powerful, and modern differentiae distinguishing the sun from other stars and planets do not seem to give the metaphor its content. More importantly, this puts a heavy burden on lexical semantics in that it requires Aristotelian definitions for particular types with respect to other elements in their class. One might be rightly skeptical that many lexical items have definitions in the requisite sense. We have to go to a higher level of generality (this sentence itself provides another example of hackneyed metaphor). More recent theories of metaphor speak of the predicate as giving a frame for the interpretation of the subject of the predication. Gentner's Structure Mapping Theory Gentner and Markman (2006), Bowdle and Gentner (2006) analyses metaphor by postulating the existence of a map from the subject term and its associated properties to the predicate and its associated properties to define metaphorical interpretation. This notion is much more general than what Asher and Lascarides proposed, because these associated properties need not be part of a definition of the term or a predicate. But it's quite unclear what these associated properties are, how metaphorical predication gives rise to such a mapping, or indeed how this mapping interacts with the linguistic system and what it does to linguistic interpretation. Furthermore, without a well defined idea of associations, this idea is rather empty.

An alternative suggests itself from the machinery of TCL that is quite general and yet also quite precise. Let us suppose that a metaphorical predication is somewhat like the definitional predications given in the previous section. To be more specific, suppose that we have a predication of t to t', $t\colon\alpha$, $t'\colon\beta$ and $\alpha \sqcap \beta = \bot$. Metaphorical predication forces us to consider t' as an object of complex type in which both types are combined but where one has a certain metaphysical priority. To say of Juliet that she is the sun does not convert her into having an aspect that is litterally a heavenly body massing more than a thousand Earth masses so as to support thermonuclear fusion at her core. Furthermore, metaphors are asymmetric.[10] For instance, these two metaphorical predications have vastly different meanings.

(11.9) a. My surgeon is a butcher.
b. My butcher is a surgeon.

The suggestion is that metaphorical predications type the subject of the predication as having a directed $\bullet$ type. We can suppose that $\alpha \bullet \beta$ is symmetric, but we can also postulate an asymmetric $\bullet$, one in which one type has metaphysical priority. I'll write such types with subscript types on the $\bullet$; e.g., $\alpha \bullet_{\alpha} \beta$ denotes a $\bullet$ type in which α has metaphysical priority. What does it mean to have metaphysical priority? For one thing, it means this:

- $t\colon \alpha \bullet_{\alpha} \beta \rightarrow t\colon \alpha$

Thus, Juliet under the type $\textsc{woman} \bullet_{\textsc{human}} \textsc{sun}$ remains a woman, a human being despite being assigned an aspect in which she shares some qualities of the sun.

The aspect in which Juliet is the sun is the one that now requires interpretation. We can appeal to the finegrained contents associated with types conveyed by true generics involving inhabitants of that type and to predications to objects of that type in prior discourse. To say that Juliet has an aspect in which she is the sun is to say that many of these predications apply mutatis mutandis to her. These predications have to be filtered or reinterpreted in order not to clash with

[10] Thanks to Tony Veale for this point and for the suggestion that metaphor might be modelled using some sort of complex type.

the fact that Juliet is a human being. This interpretation process may not be rule bound and it may depend on different ways of using structural analogies on the associated conceptual information, something which Gentner was after. The TCL system by itself, however, does not have much to say on the organization of this finegrained conceptual information. Indeed it may lie outside the linguistic system altogether. Interestingly though this sketch of an analysis of metaphorical predication gives us a way of understanding how the non-linguistic conceptual repertoire might eventually end up affecting lexical content (frozen metaphors) through the incorporation of this information into the the type system.

11.4 Vagueness

Many predicates in natural language are vague. Predicates like *is bald*, *is a heap*, *is red*, *etc.* are considered vague, because while they clearly apply to some individuals and clearly do not apply to others, there seems to be no fact of the matter as to whether they apply to certain individuals. A great deal of effort has gone into understanding how vague predication works at the level of denotations—supervaluation, partial logic, epistemic logic have all been marshalled in the service of understanding vague predication and the truth, falsity or indeterminate alethic status of such predications. This is a fascinating subject but one that really does not concern types and their internal semantics. Since vagueness is so pervasive in natural language predication, it does not seem as though one should distinguish terms as being vague or not, especially whether the predication literally succeeds in giving a definite truth value will depend a great deal on context (Kamp, 1980).

Chapter 12

Conclusions: A Sea of Arrows

The TCL framework presented in this book provides a general framework for predication. The basic idea is that if word meanings are relatively simple, then predication is more complicated than one might have thought. I have surveyed a number of constructions which make the story of predication complicated: predication involving dual aspect nouns, relative predication, and various types of predication involving coercion. To handle these phenomena I extended the standard typed lambda calculus with a two new, complex types— dot types and dependent types. All of these phenomena have a common thread: a conflict between the typing demands of a predicate and its argument in a predication lead to an accommodation of type requirements. These accommodations led in turn to adjustments in logical form. I modelled these type accommodations and their accompanying adjustments in logical form using the framework of category theory. Coercions and predications of aspects of dual aspect nouns all involved maps. Every dual aspect noun, that is a noun of $\bullet$ type, supports a map from the inhabitants of the $\bullet$ type to inhabitants of one of the constituent type.

Maps feature also in our analysis of coercion. But the maps involved in typical coercions do not take us from objects to their dependent aspects but rather to other types of entities. The most noticed dependent type in the literature is our type ϵ that maps objects into processes involving them. This seems to reflect a basic metaphysical premise that objects are conceptualized in part by the processes they undergo or can undergo. There are many types of processes that objects can be mapped to—noise making events, events with result states, events that result in the coming to be of an object (a particular kind of result state), events that involve the use of the object (another particular kind of event with a result state). The qualia of classic GL pick out a small subset of the events that are involved in coercion.

A type related to events used in the analysis of evaluative adjectives is the notion of a disposition, analyzed in terms of generic quantification over events. I postulated a dependent type that maps objects to dispositions involving them to analyse predication involving evaluative adjectives. Another important dependent type involved in grinding and in material adjectives is the map from objects to their matter. This is in principle covered by the qualia—we are speaking here of the Aristotelian material cause of objects. On the other hand, a coercion that is completely missing in the classic GL list is the coercion to a collective or distributive reading of a plural term. I postulated dependent types that converted groups into sets of individuals and vice versa; these correspond to a pair of maps from groups (acting collectively) to sets (of the members of the group) and from sets to groups. Metonymy and metonymic predication led me to postulate a dependent type from objects to their parts. Yet another dependent type is the one from objects to representations of them. Another important dependent type for the non subsective adjectives defines a map from objects to the associated traits of that object. We needed this to make sense of loose talk.

Finally, there are certain maps from abstract entities to more concrete ones. Following the analysis of Asher (1993) of facts, there is a map from atomic and conjunctive facts to eventualities. By their nature, such facts are guaranteed to have a physical instantation. Thus, one can expect a map from facts to eventualities, at least for atomic facts. For negative facts, the situation is not so clear. Negative facts like the fact that no one coughed during the concert do not map onto any clear eventuality; they assert the lack of certain eventualities. Perhaps one can use a map here to some sort of state of the concert. However, this map does not extend to all facts—in particular general or modal facts. A final but kind of scary dependent type is involved in nominalization. Nominalization takes anything of some type and converts it into something of type e. So in looks like nominalization involves the dependent type $\nu : \alpha \rightarrow e$. There must be some restriction on the size of the set of inhabitants of types in order for ν not to lead to paradoxes and some bounds on the number of subtypes in e.

Nevertheless, there are no coercions for some things. For instance there is no general map from an abstract entity (notably not an object of type $\text{P}\bullet\text{I}$) to a physical object. One could imagine the existence of such a map: a map from an object to its concrete physical instantiations, but there is no guarantee that there is a physical instantiation for an arbitrary abstract object. Thus, postulating such type coercions does not guarantee the existence of a corresponding map; such coercions would not be sound operations. We can state the generalization relative to our semantic model: a coercion exists in a given typing context, if it can be established that a corresponding map at the level of denotations exists given the information in that context. Thus, coercions like those noticed by Nunberg, between drivers and their cars or between books and their authors or between individuals and the dishes they have ordered may be felicitous only in certain contexts. These are clearly more fragile and contextually determined than the other coercion processes I have discussed above. It is, finally a matter of considerable subltety and philosophical reflection of when coercions are sound in a given context—i.e., when we can demonstrate the existence of a suitable map between objects of one type and objects of another. As Mel'cuk and others also found, there is a rich set of maps underlying lexical meaning. Even more interestingly this set of maps evolves and changes as discourse proceeds. It is this sea of maps and how it changes that constitutes the web of words, truly a marvelous creation of human kind.

Bibliography

Abney, S.: 1987, *The English Noun Phrase in its sentential aspect*, PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts. Distributed by MIT Working Papers in Linguistics.

Aehling, K. and Schwichtenberg, H.: 2000, A syntactical analysis of non-size-increasing polynomial time computation, *Proceedings of the 15th Symposium on Logic in Computer Science (LICS 2000)*, IEEE Computer Science Press, Los Alamitos, CA, pp. 84–91.

Asher, N.: 1993, *Reference to Abstract Objects in Discourse*, number 50 in *Studies in Linguistics and Philosophy*, Kluwer, Dordrecht.

Asher, N.: 2007, Objects, locations and complex types, *in* L. Vieu and M. Aurnague (eds), *The Categorization of Spatial Entities in Language and Cognition*, Benjamin.

Asher, N. and Lascarides, A.: 1995, Lexical disambiguation in a discourse context, *Journal of Semantics* **12**, 69–108.

Asher, N. and Lascarides, A.: 1998, The semantics and pragmatics of presupposition, *Journal of Semantics* **15**, 239–299.

Asher, N. and Lascarides, A.: 2001, Indirect speech acts, *Synthese* **129**(1–2), 183–228.

Asher, N. and Lascarides, A.: 2003, *Logics of Conversation*, Cambridge University Press.

Asher, N. and Pustejovsky, J.: 2004, Word meaning and commonsense metaphysics, Available from `nasher@la.utexas.edu`.

Asher, N. and Reese, B.: 2005, Negative bias in polar questions, *in* E. Maier, C. Bary and J. Huitink (eds), *Proceedings of SuB9*, pp. 30–43. `www.ru.nl/ncs/sub9`.

Asher, N. and Sablayrolles, P.: 1995, A typology and discourse semantics for motion verbs and spatial pps in french, *Journal of Semantics*.

Asperti, A. and Longo, G.: 1991, *Catgories, Types and Structures: An Introduction to Category Theory for the Working Computer Scientist*, MIT Press.

Barr, M. and Wells, C.: 1990, Category theory for computing science, 3d edition, available University of Montreal's centre de recherches mathématiques.

Black, M.: 1962, *Models and Metaphors*, Cornell University Press.

Black, M.: 1979, More about metaphor, *in* A. Ortony (ed.), *Metaphor and Thought*, Cambridge University Press, Cambridge, pp. 19–45.

Blackburn, P.: 1997, Modal logic and attribute value structures, *in* M. de Rijke (ed.), *Diamonds and Defaults*, Kluwer, Dordrecht, Holland, pp. 151–168.

Bouillon, P.: 1997, *Polymorphie et sémantique lexicale: le cas des adjectifs*, Presses Universitaires du Septentrion, Lille.

Bowdle, B. and Gentner, D.: 2006, The career of metaphor, *Psychological Review* **112**(1), 193–216.

Busa, F.: 1996, *Compositionality and the Semantics of Nominals*, PhD thesis, Brandeis.

Carston, R.: 2002, *Thought and Utterances: The Pragmatics of Explicit Communication*, Blackwell's Publishing, Oxford.

Climent, S.: 2001, Individuation by partitive constructions in spanish, *in* P. Bouillon and F. Busa (eds), *The Syntax of Word Meaning*, Cambridge University Press, Cambridge, pp. 92–123.

Cooper, R.: 2005, Do delicious lunches take a long time?, ESSLLI 2005 presentation.

Copestake, A. and Briscoe, E. J.: 1995, Semi-productive polysemy and sense extension, *Journal of Semantics* **12**, 15–67.

Coquand, T.: 1986, An analysis of girard's paradox, *Proceedings of the IEEE Symposium in Logic and Computer Science*, pp. 227–236.

Crole, R. L.: 1993, *Categories for Types*, Cambridge University Press.

Danlos, L.: 2001, Event coherence in causal discourses, *in* P. Bouillon and F. Busa (eds), *The Syntax of Word Meaning*, Cambridge University Press, Cambridge.

Davidson, D.: 1968/69, The logical form of action sentences, *in* P. Ludlow (ed.), *Readings in the Philosophy of Language*, MIT Press, pp. 337–346.

de Groote, P.: 2001, Towards abstract categorial grammars, *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pp. 148–155.

de Groote, P.: 2002, Tree-adjoining grammars as abstract categorial grammars, *TAG+6, Proceedings of the sixth International Workshop on Tree Adjoining Grammars and Related Frameworks*, Università di Venezia, pp. 145–150.

Donnellan, K.: 1966, Reference and definite descriptions, *Philosophical Review* **75**, 281–304.

Egg, M.: 2003, Beginning novels and finishing hamburgers: Remarks on the semantics of *to begin*, *Journal of Semantics* **20**, 163–191.

Fernando, T.: 2004, A finite-state approach to events in natural language semantics, *J. Log. Comput.* **14**(1), 79–92.

Fodor, J. and Lepore, E.: 1998, The emptiness of the lexicon: Critical reflections on j. pustejovsky's *the generative lexicon*, *Linguistic Inquiry*.

Frank, A. and van Genabith, J.: 2001, Glue tag: Linear logic based semantics construction for ltag- and what it teaches us about the relation between lfg and ltag, *in* M. Butt and T. H. King (eds), *Proceedings of the LFG '01 Conference*, CSLI Publications, Online Proceedings, Stanford, CA. http://cslipublications.stanford.edu/LFG/6/lfg01.html.

Gentner, D. and Markman, A.: 2006, Defining structural similarity, *The Journal of Cognitive Science* **6**, 1–20.

Girard, J.-Y.: 1987, Linear logic, *Theoretical Computer Science* **50**, 1–102.

Godard, D. and Jayez, J.: n.d., Towards a proper treatment of coercion phenomena, *Proceeding of the 1993 European ACL*.

Groenendijk, J. and Stokhof, M.: 1991, Dynamic predicate logic, *Linguistics and Philosophy* **14**(1), 39–100.

Gross, M.: 1975, *Méthodes en Syntaxe*, Hermann, Paris.

Howard, W. A.: 2001, The formulas-as-types notion of construction, *Linguistic Inquiry* **32**(4), 555–598.

Hunter, J. and Asher, N.: 2005, A presuppositional account of indexicals, *in* P. Dekker and M. Francke (eds), *Proceedings of Fifteenth Amsterdam Colloquium*, ILLC, Department of Philosophy, University of Amsterdam, pp. 119–124.

Jaeger, G.: 2003, Toward an explanation of copula effects, *Linguistics and Philosophy*.

Kamp, H.: 1981, A theory of truth and semantic representation, *in* J. Groenendijk, T. Janssen and M. Stokhof (eds), *Formal Methods in the Study of Language, Part 1*, Vol. 135 of *Mathematical Center Tracts*, Amsterdam, pp. 277–322. Reprinted in Jeroen Groenendijk, Theo Janssen and Martin Stokhof (eds), 1984, *Truth, Interpretation, and Information; Selected Papers from the Third Amsterdam Colloquiam*, Foris, Dordrecht, pp. 1–41.

Kamp, H. and Reyle, U.: 1993, *From Discourse to Logic*, Kluwer, Dordrecht.

Kamp, H. and Rossdeutscher, A.: 1994a, Drs-construction and lexically driven inference, *Theoretical Linguistics* **20**(2), 165–235.

Kamp, H. and Rossdeutscher, A.: 1994b, Remarks on lexical structure and drs construction, *Theoretical Linguistics* **20**(2), 97–164.

Kleiber, G.: 1999, *La Polysémie en question*, Presses Universitaires de Septentrion.

Lakoff, G. and Johnson, M.: 1980, *Metaphors We Live By*, University of Chicago Press.

Lambek, J. and Scott, P. J.: 1986, *Introduction to Higher Order Categorial Logic*, Cambridge University Press.

Lascarides, A. and Copestake, A.: 1995, The pragmatics of word meaning, *Proceedings SALT V*, Ithaca, NY.

Lascarides, A., Briscoe, E., Asher, N. and Copestake, A.: 1996, Order independent and persistent typed default unification, *Linguistics and Philosophy* **12**, 1–89.

Levin, B.: 1993, *Towards a Lexical Organization of English Verbs*, University of Chicago Press, Chicago.

Martin-Löf, P.: 1980, *Intuitionistic Type Theory*, Bibliopolis, Naples, Italy.

Moravscik, J.: 1975, Aitia as generative factors in aristotle's philosophy, *Dialogue* **14**, 622–636.

Muskens, R.: 2007, Intensional models for the theory of types, *Journal of Symbolic Logic* **72**(1), 98–118.

Nunberg, G.: 1979, The non-uniqueness of semantic solutions: Polysemy, *Linguistics and Philosophy* **3**, 143–184.

Nunberg, G.: 1995, Transfers of meaning, *Journal of Semantics* **12**, 109–132.

Ortony, A.: 1979, *Metaphor and Thought*, Cambridge University Press.

Partee, B. and Rooth, M.: 1983, Generalized conjunction and type ambiguity, *in* R. Baurle, C. Schwartze and A. von Stechow (eds), *Meaning, Use and Interpretation of Language*, De Gruyter, Berlin, pp. 361–383.

Pogodalla, S.: 2004, Computing semantic representation: Towards acg abstract terms as derivation trees, *TAG+8, Proceedings of the sixth International Workshop on Tree Adjoining Grammars and Related Frameworks*, University of Vancouver.

Pustejovsky, J.: 1995, *The Generative Lexicon*, MIT Press.

Pustejovsky, J.: 2001, Type construction and the logic of concepts, *in* P. Bouillon and F. Busa (eds), *The Syntax of Word Meaning*, Cambridge University Press, Cambridge, pp. 92–123.

Ranta, A.: 2004, Grammatical framework: A type-theoretical grammar formalism, *Journal of Functional Programming* **14(2)**, 145–189.

Recanati, F.: 1993, *Direct Reference: From Language to Thought*, Blackwell's Publishing, Oxford, U.K.

Recanati, F.: 2002, Unarticulated constituents, *Linguistics and Philosophy* **25**, 299–345.

Recanati, F.: 2004, *Literal Meaning*, Cambridge University Press, Cambridge, U.K.

Reese, B. and Asher, N.: 2006, The nuclear option: The interpretation of tag questions, *Proceedings of SuB11*.

Rothstein, S.: 2004, *Structuring Events: A Study in the Semantics of Lexical Aspect*, Blackwell's Publishing.

Sag, I.: 1981, Formal semantics and extra-linguistic content, *in* P. Cole (ed.), *Syntax and Semantics 13: Radical Pragmatics*, Academic Press, New York, pp. 273–294.

Saul, J. M.: 1997a, Reply to forbes, *Analysis* **57**(2), 114–118.

Saul, J. M.: 1997b, Substitution and simple sentences, *Analysis* **57**(2), 102–108.

Saul, J. M.: 1998, The pragmatics of attitude ascription, *Philosophical Studies* **92**, 363–389.

Saul, J. M.: 1999, Substitution, simple sentences and sex scandals, *Analysis* **59**, 106–112.

Searle, J.: 1979, Metaphor, *in* A. Ortony (ed.), *Metaphor and Thought*, Cambridge University Press, Cambridge, pp. 92–123.

Sperber, D. and Wilson, D.: 1986, *Relevance: Communication and Cognition*, Blackwell's, Oxford.

Szabo, Z.: 2003, On qualification, *Philosophical Perspectives* **17**, 409–438.

van der Sandt, R.: 1992, Presupposition projection as anaphora resolution, *Journal of Semantics* **9**, 333–377.

Veltman, F.: 1996, Defaults in update semantics, *Journal of Philosophical Logic* **25**, 221–261.

Verspoor, K.: 1997, *Contextually Dependent Lexical Semantics*, PhD thesis, University of Edinburgh.

Vespoor, C.: 1996, Lexical limits on the influence of context, *Cognitive Science*.

Wechsler, S.: 2005, Resultatives under the 'event-argument homomorphism', *in* N. Erteschik-Shir and T. Rapoport (eds), *The Syntax of Aspect*, Oxford University Press.