The Metaphysics of Words in Context*

Nicholas Asher
Department of Philosophy
University of Texas@Austin
nasher@bertie.la.utexas.edu

James Pustejovsky
Department of Computer Science
Brandeis University
jamesp@cs.brandeis.edu

February 28, 2000

1 Introduction

Recent advances in the study of the lexicon and discourse have suggested a rich and complex theory of composition. At the level of the clause such a theory is needed to explain the systematic variation of meaning behind examples like:

(1) a. Jane enjoyed the movie.
b. Alexis enjoyed the proof.
c. Sheila enjoyed the novel.
d. Elizabeth enjoyed the route.

To determine the meaning of these sentences from lexical items, we need at least a notion of composition on which combining the meanings of two words may result in a change to those meanings themselves. That is, we need at the very least a context-sensitive notion of composition, which is not what we find in the standard theory of composition found, e.g., in Montague

*We would like to thank Sheila Asher, Tim Fernando, Ivan Sag, Stan Peters, David Israel Alex Lascarides, Ann Copestake, and Johanna Seibt for helpful comments on earlier drafts.

Grammar. If we understand these shifts in meaning to be a matter of shifts in categorization, or in *semantic type*—in a manner similar to earlier work on meaning change during composition known as "type shifting" (e.g., Rooth and Partee, 1983), then we have the approach of the *generative lexicon* or GL (Pustejovsky, 1995).

In a similar vein, recent advances in discourse interpretation have furnished a way of integrating pragmatics and semantics together into a context sensitive theory of discourse interpretation. SDRT is one such approach (Asher 1993, Lascarides and Asher 1993); exploiting the rhetorical function of information, it also introduces a complex and context sensitive type structure into the notion of discourse update—viz. new information may be added to the context in a number of different ways reflecting distinct rhetorical functions. In the pair of examples in (2), we see two very different rhetorical functions being exploited in order to create coherent interpretations but different temporal and causal structures:

(2) a. John entered. Max greeted him.
 b. John fell. Max pushed him.

Both GL and SDRT are reactions to much simpler theories of the lexicon and discourse update (i.e., the atomistic Fodorian lexicon, and standard dynamic semantics, respectively) that failed to explain data about composition and about interactions between pragmatics and semantics. What earlier theories lacked was an account of how the "composition" of new information in context could in fact alter the information as well as elements in the context in ways not predictable within a framework coutenancing only operations like application of functions to arguments or merge. That such changes are an inevitable part of interpretation is already implicit in, for instance, DRT's account of anaphora resolution; but only GL and SDRT make this the core of their approach to meaning.

We believe that context-sensitive lexical composition and discourse interpretation share many tools and some of the same problems. We believe further that integrating discourse principles into the lexicon and a rich lexicon into the computation of discourse interpretation will greatly benefit both areas of linguistics. SDRT must already make heavy use of lexical information, as in (2) in computing discourse relations. The interface between the lexicon and SDRT currently is a bottleneck that prevents a significant enlargement of SDRT's coverage. As documented in Asher and Lascarides (1995), we need certain type hierarchical organizational properties in the lexicon to get a proper theory of discourse structure construction; without generalizations on lexical types, the axioms needed to derive the appropriate discourse structures are hopelessly specific. One of the reasons for integration is to understand better discourse inferences. Lexical semantics can also profit from SDRT. A lexical theory should be sensitive to facts about discourse interpretation as well as sentential composition. Discourse structure and context, for instance, can obviously affect lexical meaning:

(3) The goat hated the film but enjoyed the book.

Depending on whether the context is a fairy tale or not, (3) will convey the same or different a different sense of *enjoy* to that assumed in (1a).

A further reason to integrate SDRT and GL is that they also already share a common approach and a common problem. They are both involved in computing logical forms for which model theoretic interpretations can be supplied. SDRT makes use of a special purpose logic with limited access to other domains for building up logical forms of discourses from underspecified logical forms for clauses. This "glue" logic has a limited and partial access to the information content of discourse constituents, Lascarides and Asher (1993) and Asher and Fernando (1997) argue, because a full access to information content would render the task of computing logical forms hopelessly complex. Though we do not believe all of linguistic understanding should necessarily be computationally simple, computing logical forms, which is the prerequisite to any deeper understanding, should be a simple matter. This in turn leads to a strong distinction in SDRT between information available to the linguistic system, of which the glue logic is part, and nonlinguistic, world knowledge. This separates SDRT and us from competing approaches (e.g. Hobbes et al. 1993) based on Quinean scruples in which a general purpose reasoning system is used with no distinctions between linguistic and nonlinguistic knowledge.

We mentioned earlier that GL's method of composition involves a manipulation of types. We can see in this approach a need for a *composition logic*, which, like the glue logic of SDRT, has partial access to common sense, metaphysical information or world knowledge, and which has a similar purpose to the glue logic for SDRT—building up logical forms. The composition logic builds up logical forms for clauses, while SDRT's glue logic builds discourse structures from logical forms for clauses. With partial access to common sense metaphysics, the composition logic can exploit that information in guiding shifts in type during composition efficiently. Thus, like SDRT's view of discourse interpretation, GL's view is that word meaning is distinct from non-linguistic or world knowledge. This is a tricky point, because of course world knowledge informs the meaning of words as well as influences discourse structure. The point again that we want to make is that the linguistic system has at best partial access to that information. Taking our cue from the SDRT picture, we give another argument for why the composition logic could be a logic of types: computational efficiency. Metaphysical information will be simplified into type information; the type logic will be a drastically simplified reflection of the logic that underlies general reasoning. SDRT's approach to computing logical form is thus reflected in a GL composition logic, which is what we will provide below.

We have spoken of integrating SDRT and GL. But we do want to keep discourse interpretation and lexical semantic modules distinct. Many people have advocated just making do with one general purpose pragmatic reasoning system to handle it all. We don't believe that is the right way to go, because the two systems, the composition logic and the discourse logic, interact in subtle and interesting ways. Merging all into one system would ride roughshod over these distinctions.[1] There are conventional associations such as those encoded (1), which are overturned by more specific information in a particular discourse context (3) (for discussion and a theory of this interaction using default unification and the glue logic DICE of SDRT see Asher and Lascarides (1995) or Copestake and Lascarides (1998)). For instance, discourse context can alter types; goats that are animals can become talking persons. Objects of one type can take on characteristics that they don't have. Furthermore, lexical ambiguities can be resolved by discourse in ways that override lexical preferences (Asher and Lascarides 1995).

In this paper, we would like to begin exploring what such an integration might look like, what problems it would face and what advantages it might offer. We'll concentrate on developing the composition logic and the account of complex types, in which we'll make use of various SDRT principles. As we are interested in composition of information to build logical forms, we will build on the standard way of getting logical forms—the lambda calculus in which (functional) types are exploited. By relating types in the lexicon

[1] For more details see Lascarides and Copestake 1995.

we can give partial, implicit definitions, which will help together with how the items compose together determine inferences based on truth conditional contents. Secondly, by developing a strongly typed theory of lexical items and a theory of how such lexical items combine and interact in the process of semantic composition and of discourse interpretation, we can constrain the lexical semantics with predictions of semantically well-formed or ill-formed predications and word combinations. We'll produce a new type calculus that captures one of the general ideas of the generative lexicon: providing a set of techniques governing type shifting possibilities for various lexical items so as to allow for the combination of lexical items in cases where there is an apparent type mismatch. These techniques themselves should follow from the way the lexicon is organized and its underlying logic.

2 Motivating Data

Some of the motivating data for our type logic comes from problems handled by classic GL analyses— for instance, the logical metonymy examples of (1. There are similar data for other verbs like *begin, start, finish* that select particular types of events associated with the objects—typically events in which the object is used for its designated purpose (cf. Pustejovsky and Bouillon, 1995, Saint-Dizier, 1998).

(4) begin a cigarette (i.e., smoking)

(5) enjoy the book (i.e., reading)

To explain these readings, we will appeal to other forms of types besides just the primitive types supplied by metaphysics and functional types. The GL explanatory device for these data about semantic composition is to exploit *type structure* and the accessibility of the qualia structure in composition. In this paper we will enrich the notion of types and analyze composition so that it is directly affected by types. We will argue that information encoded in metaphysical categories is conventionally "lifted" into the type structure and then exploited in semantic composition.

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

(6) 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. The last example above, however, is somewhat difficult, where "water that is moving fast" is the desired interpretation, one which does not seem to involve reference to qualia.

Another application of qualia-based typing has been to account for certain denominal verb formation cases, as discussed in Hale and Keyser (1993). Within GL, the phenomenon has been analyzed as follows: the noun-to-verb transformation is licensed in just those cases where the nominal's TELIC role involves an object in a relation that matches the target verb's direct object. Hale and Keyser's canonical examples are illustrated in (7) below.

(7) 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.

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. This is discussed in Pustejovsky and Busa (1997) and Climent (1999) in more detail.

Event structures introduced by causative and other verbs also motivate a more complex system of types involves reference to the event structure inherent in propositional content. For example, the notion of *event headedness* (cf. Pustejovsky, 1995) is motivated by the observation that many lexically expressed event types are complex (cf. for example Kamp and Rossdeutscher's treatment of *heilen*). Yet adverbial modifiers to verbs expressing such complex types modify typically only one event in the structure. Headedness, acting as a focusing mechanism identifying or distinguishing one event in the complex, is intended to help explain such phenomena. Pustejovsky and Busa (1995) extend the applicability of event headedness in an attempt to explain the polysemy between inchoative and causative verbal pairs, such as *break* and *sink*. Arguing that both forms are fundamentally transitions, heading the initial event produces the transitive form while heading the final event gives rise to the incohative form.

Another enrichment to the type system proposed in Generative Lexicon is the notion of a *dot object*. Dot objects were first introduced to explain copredications in the context of polysemy (Pustejovsky, 1994). Unlike qualia, where there is an asymmetric dependency of the qualia type on the "head" type, with dot objects, the two types are mutually interdependent; you cannot have one without the other. The intuition is that this mutual presence of types should explain copredication. Failure of copredication points to a lack of a dot object. Thus, as in (8 below, we see that contrastively ambiguous words (Pustejovsky, 1995) do not introduce a dot object, where two distinct senses are simultaneously accessed.

(8) !The bank specializes in IPO's and is being quickly eroded by the river.

On the other hand, we see that many words appear to give rise to dot objects, though not all copredications are equal (cf. (9). We believe that this has to do with the fact that dot objects make reference to coherence relations that are essential to structuring discourse. And some cases of copredication are incompatible with the discourse structure implicit in the dot object.

(9)
a. The Sunday newspaper weighs 5 lbs and documents in depth the economic news of the week.
b. ??The newspaper was founded in 1878 and weighs 5 lbs.

(10) a. John works in model theory and is my next appointment.
b. The bay that John swam in curved from the lighthouse to a sandy spit.
c. The city that John lives in just passed a law against smoking in bars.
d. The bottle has a nice label and is a merlot.
e. The temperature is ninety and rising.
f. The temperature is predictable and ninety.

Not all copredications, however, need be dot objects. Some may exploit dependent aspects of the types of the subjects such as qualia structure. In (11), for example, it appears as though some predicates make reference to aspects having to do with the telic role of the subject NP.

(11) a. The novel was 300pp. long and lasted the whole flight.
b. Your last glass of wine was a Merlot and lasted half an hour.
c. Lunch was delicious but took forever.

It is also unclear whether we want to think of grinding as producing dot objects though these too support copredication.

(12) a. Rabbits live in our yard and taste yummy.
b. Rabbits live in our yard and make great stew.
c. Snow is white and there's lots of it around my house right now. (from F. J. Pelletier, p.c.)

More motivating data for dot objects comes from the following minimal pairs involving quantification over different aspects of objects.

(13) a. The student read every book in the library.
b. The student stole every book in the library.

(14) a. The teacher answered every student's question.
b. The teacher repeated every student's question.

(15) a. Many cities have adopted anti-smoking legislation.
b. Many cities vote Democratic.
c. Many cities have good restaurants.

The quantification over books in (13) for example, is sensitive in one case to its informational aspect, and in the other to its physical aspect. In (13a), we simply quantify over all informationally distinct books; it is not necessary, for example, for the student to have read every book in the library that she have read every distinct copy of every book, whereas she must have stolen every copy in order for (13b) to be true.

One might think that a simple account of these examples would just involve coercing *book* to be in some cases a physical object, and in other cases an informational one. But then it's difficult to make sense of copredication data. Further, we need to keep the information around that a book is defined as both, because of examples like the followning:

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

If the verb *master* involves selecting for the informational sense of book, we cannot "use up" the dot object *book* when predicating burning of it in the first sentence. Otherwise we will be unable to bind the anaphor in the second clause, unless we have the unintuitive rule that *master* coerces all its physical objects into physical and informational objects (*master the stairs*, *master the maze*). However, there are predicates that do require a dot object as complement; *read*, for instance, coerces its direct object into something of just this complex type (cf. Pustejovsky, 1998a).

(17) 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.

For both (17b) and (17c) 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 (17b) the coercion effects the introduction of the physical manifestation to the otherwise informational type; for (17c) the coercion results in the introduction of an informational component to an otherwise merely physical type.

It has been suggested by Barbara Partee (p.c.) that the quantificational ambiguity seen above with *read/steal* can be handled by treating the entire phenomenon as an instance of the type/token distinction. According to this suggestion, (13a) makes reference to the type while (13b) refers to the token. While not discounting this approach completely, there appear to be two problems with that solution. First, simply reducing the above phenomenon to a type/token distinction doesn't solve the problem of how the copredication works; if the type /token suggestion were right, we could envision using that distinction along with our dot object apparatus in the analysis, but without the latter, we don't see what the analysis would be. 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 (18b), quantification is over informational tokens which are distinct from the actual physical object tokens that would be available.

(18) a. John stole every Beethoven 5th Concerto score in the library.
b. John mastered every Beethoven 5th Concerto score in the library.

Hence, for a dot object, 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.

A final piece of motivating data for complex types comes from the phenomenon of *cocomposition* as in (19) but not (20) below, where composing verb and object meaning yields different types sometimes for the verb and sometimes for the object.

(19) a. The sun baked the rock.
b. James baked the potato.
c. James baked the cake.

(20) a. The rock fell.
b. John fell.

We will re-examine much of this data from the perspective of our new composition logic in the section after next. First, however, we want to turn to the metaphysical picture suggested by complex types and what in general should be the relation between metaphysics and the lexicon.

3 Lexical Semantics and Common Sense Metaphysics

Common sense metaphysics informs lexical semantics by providing the basic types and basic relations between types, and a lexical semantics reflects thus a background metaphysical picture. To understand various phenomena about composition, such as cocomposition or logical metonymy, we need other forms of types besides just the primitive types supplied by a standard metaphysics and functional types. Thus standard GL proposes four basic structures for the lexicon—argument structure, event-type structures with headedness, qualia, a logic of inheritance including dot objects (cf. Pustejovsky, 1998). We will develop a framework here in which the last three structures are shown to be fundamentally similar. All of these structures have a common element that we make explicit; they all introduce complex types whose components are related by *coherence* relations specified by the background metaphysics. Qualia, headedness and dot objects are all complex types exploiting different types of coherence relations. We'll then define a context sensitive logic of composition for these complex types. These rules will allow us to exploit and introduce complex types as context demands.

In assimilating qualia and GL's event structures to dot objects, we are changing the outlook of the theory from one wedded to rather familiar notions of Aristotelian and other metaphysical systems to something new. We will use two type constructors, one for copresent types or dot objects, $\bullet$ and one for dependent types $\otimes$. The qualia are dependent type constructors—e.g., relating in certain cases object types and eventualities (e.g., Agentive or $\otimes_A$ and Telic or $\otimes_T$). We'll call these QQC's or *quasi qualia constructs*. Similarly, we represent complex events as dot objects. For the most part in the formalization below we will ignore distinctions between qualia. Similarly, we represent complex event types, previously studied in headed structures in GL, as dot objects. So on the one hand, this assimilation brings a cleaner and more unified picture of the type constructors operative in the lexicon.

But the real motivation for these type constructors is metaphysical. There are problems with the Aristotelian picture of (standard) GL. Our view of substances in commonsense metaphysics is itself fluid and can take a number of diverse perspectives. These perspectives can be created on the fly and depend also on rhetorical structure—witness the odd patterns in (9). To look at the problems in more detail, let's consider a particular dot type, that introduced by *book*. How would we analyze *book* without dot types? We could imagine an Aristotelian explanation of the following kind: the FORMAL cause of *book* is the information it contains; the material cause (CONSTITUTIVE role) is the physical realization of the book, while the efficient (AGENTIVE) cause is the writing and its TELIC role is to be read. But in fact, the information in a book has a separate mode of coming into being from the physical object itself; similarly, the constitution of the information refers to notions such as, for example, narratives, chapters, sections, and so on, while the constitution of the physical object refers to pages, binding, and the physical components of the object. Another problem with this approach is that the predicates that are *sui generis* related to entities that are both physical objects and information bearing objects are actually mistyped here; that is, both *read* and *write* are only understandable in terms of a complex typing system. The objects selected for by a predicate such as *read* and the predicates made reference to by objects such as *books* are the characteristic functions for those entities. Many lexical types behave in the same way as *book* For instance, *temperature*, *sonata*, *human* or *city* (an organization, a location, a collection of people), which clearly seem to be dot objects. The Aristotelian model simply breaks down on such objects; consider for instance all the problems medieval philosophers had with the Aristotelaian conception of human, according to the formal cause is the rational agency and the physical object is the material cause.

What binds the perspectives that constitute a dot object together? A necessary condition is that the perspectives are all of a single substance; and hence ambiguous words like *bank* don't denote dot objects. Thus, a suggestive metaphysical picture for dot objects is that of Spinoza's: perspectives that are constitutive of the dot are modes of a single substance—though unlike Spinoza, common sense metaphysics countenances many substances and many different modes. Now what modes can go together? There doesn't seem to be an a apriori answer. Some basic metaphysical relations, however, percolate up to the linguistic system as rhetorical or coherence relations (Asher, forthcoming), and these sometimes serve to relate the elements of a dot object together, as we will see with the analysis of causal verbs. Dot objects like *book* or *city* have more obscure relations that bind the differing perspectives together. Perhaps common sense metaphysics doesn't offer a coherent theory other than that which can be summed up with another simple discourse relation: Elaboration. That is, the constituents are just particular Elaborations of the basic object—exhibiting one aspect of it. Supporting evidence for this comes from reflecting on predications of dot objects. What goes on when we predicate properties of a constituent or subtype of the dot, as in *the book is heavy*, a sentence that we treat in some detail below. Predication singles out or highlights one aspect of the dot object and elaborates on it. So the relation between the "object" of which *heavy* is predicated and the original dot object is something like the discourse relation Elaboration. As we wish to remain agnostic as to whether it is *exactly* the discourse relation Elaboration, we'll call it O(bject)-Elaboration.

The information encoded in metaphysical categories is "lifted" conventionally into the type structure and then exploited in semantic composition. As we've said, we will have constructors for dot objects and for QQC's. Dot objects and QQCs are very similar, except they exploit different coherence relations. Dot objects use extensional coherence relations like Elaboration, whereas tensors don't. This leads to different sorts of inference rules for exploiting the two complex types. Those for dot objects will have no restrictions on their use and always apply, whereas tensor exploitation rules for getting out a dependent type will be default and subject to consistency checks.

We have seen that dot objects are decidedly non Aristotelian. But the Aristotelian model breaks down in other places as well for the lexicon. Systematic metaphysics, of which Aristotle's is one brand, presumably gives us a consistent ontology and relational system, but we cannot presume the same for commonsense metaphysics. It's clear that for example the set of relations on types in natural language metaphysics is not always well defined. The Aristotelian causes that standard GL uses in qualia structures—a thing's function, it's (efficient) cause, its constitutive parts or matter, and its genus—are universal features of substances in Aristotelian metaphysics. Yet many of these modes of explanation seem ill defined or irrelevant to certain substances like natural kinds; so unlike Aristotle they aren't universal features. But then what's the relation between the qualia and metaphysics? Further, it's not at all clear that we want to have these explanatory causes *a priori* associated with types of eventualities. A clear case is *bake* which is intuitively a complex type involving a process and a result. In earlier formulations of GL *bake* has as agentive a baking act, but this doesn't seem right; the baking act *is* the process that is a part of or constitutive of baking and it is more than metaphysically problematic to have a part cause the whole. So we want to have some flexibility with our complex types that the Aristotelian systematic picture of the world doesn't have.

Key to our view is the complex relation between commonsense metaphysics and the lexicon. Metaphysics gives us an insight into basic types (ontology) and basic relations between types. This informs the lexicon, but it's distinct from it. There is a connection between commonsense metaphysics and the lexicon but it's not a direct one. There is a fence (a filter) from commonsense metaphysics to lexical information. Metaphysical categorization becomes *type information* in the lexicon and the complex first order or higher order relations of metaphysics become the underlying coherence and rhetorical relations that are central to SDRT and that we will use in analyzing complex types.

Clearly one reason to distinguish common sense metaphysics from the lexicon is that metaphysics is only one contributory factor to the lexicon; syntax and perhaps also morphology contribute to the semantic argument structure, while metaphysics contributes to the complex semantic types that correspond to syntactic categories. Context may also contribute to certain specialized senses or types in the lexicon.

Another reason for distinguishing between the lexicon and metaphysics is computational. The lexicon simplifies reasoning in many ways. First, type information is quantifier free, whereas it's hard to imagine any formalization of commonsense metaphysics doing without quantification (in fact it typically exploits higher order quantification). In fact, our type logic will only feature conjunction and conditionals. We'll capture incompatibility between types in terms of their common join, bottom. A second way our type logic will be more efficient is that it will exploit type hierarchies, in which for instance incompatibilities are already precomputed (given by metaphysics). This makes defeasible inference feasible—taking it out of the realm of the uncomputable to the possibly quite efficient. More generally, it makes the knowledge of the meaning of words much simpler and computationally much easier. And after all, building logical forms, which is what we are trying to account for should be easy; it's a minimum standard of semantic competence.

In particular we think that our complex types can help simplify reasoning in the lexicon even further. Years of intensive research on attempts to make an `is_a` hierarchy of concepts associated with lexical items have produced some rather baroque systems with multiple inheritance, default inheritance and so on, but these hierarchies have typically run into trouble in trying to keep the hierarchy as a partition of things that are. Pustejovsky (1998) has proposed that these ornate and often unworkable "ontologies" can be avoided, if we abandon the often unexamined assumption that *every* concept associated with a lexical item should end up in the hierarchy. Instead he proposes that many lexical concepts and even supertypes can be *defined* in virtue of simple types of "basic substances" and certain universal features of substances like the Aristotelian causal categories. This allows us to come up with context sensitive hierarchies of defined sorts that supervene on the basic `is_a` hierarchy. For example, the type *artifact* can be defined as Pustejovsky by using agentive qualia. For such substances their origin or the event of their construction is pertinent to understanding what they are. We will recast this suggestion as saying that artifacts are all of a complex type of the form $\sigma \otimes_{A,T} (\tau, \tau')$—this complex type is one in which types of agentive and telic eventualities are associated with the "head" type σ. And this holds just for those objects that have been intuitively created for some purpose. $\otimes$ is one type constructor that we will use.

The language of the lexicon is also less expressive than that of commonsense metaphysics. So we can introduce a bit of "slop" in the move from metaphysics to the lexicon. Consider *enjoy the perfume.* If the TELIC role of *perfume* is to make its wearer smell good, it is not this event that is the referent of *enjoy.* But if we have the QQC telic dependent subtype of *enjoy* be something more underspecified like the type SMELL, then we can in fact understand *enjoy the perfume* as a type of QQC exploitation, which we will detail below.

Finally the fence between metaphysics and lexicon allows us to account for an intractable problem for standard GL: semi-productivity. The lexicon is not fully productive in many cases; and this poses a problem for unconstrained operations in GL, as pointed out in Godard and Jayez (1993) and Pustejovsky (1995a).[2]

[2] To address the problem of overgeneration in GL, one can try to impose constraints directly on the way that qualia structure is made reference to in the type structure for

In particular, sometimes we see that specific types fail to inherit a general feature from their supertypes *without* imposing a particular new value. Here are some standard cases of semi-productivity involving QQCs. While *garden* or *door* have clear TELIC roles (in commonsense metaphysics), they are not accessible in semantic composition (! indicates semantically ill-formed):

(21) a. !John enjoyed the door.
b. !John enjoyed the garden.
c. !John enjoyed the bathroom.

The function, for example, of a door is to offer an entrance to some enclosure, but *enjoy the door* plainly means to enjoy its appearance. So it looks as though *enjoy* can't always select for the telic role. Similarly for *enjoy the pipe* at least as a default or *enjoy the bathroom*.[3]

If the lexicon is distinct from metaphysics, we open up the possibility that QQC's only attach to some words and not others. Conventions will decide what words introduce QQCs and what those QQCs are. We will show how to account for these examples below in detail, but our point here is that by distinguishing metaphysics from the lexicon, we can both maintain that something like a door or a bathroom has a proper function without being required to claim that that function is part of the lexical entry. It is just such functions that sometimes don't make it over to the lexicon from the metaphysical scheme.

---

the composition. This is the approach taken in Pustejovsky (1995a), Bouillon (1997), and Busa (1996), and could be called *Qualia Extended GL*. But this just threatens to reproduce the problem of semi-productivity at another level.

[3]Other aspectual verbs like *begin* involve more complicated forms of coercion and metonymy than *enjoy*. Sometimes the complement selects what we would think is the TELIC role, sometimes the AGENTIVE role, yet sometimes it selects for neither (cf. Pustejovsky and Bouillon, 1995). Consider the following examples:

(22) a. John began the painting.
b. John began the film.
c. John began the sonata.
d. John began the novel.

One can explain the AGENTIVE role for painting because of selectional restrictions on the event object argument for *begin*. But that does not explain the agentive reading for *begin the sonata*. There are actually two agentive readings available for *sonata*, by virtue of its status as a dot object —playing the sonata and composing the sonata. We return to this issue later in the paper.

4 Our Proposal in more detail

Having set out our notions for complex types and the underlying relations between them, we now need a logic for manipulating these types that will allow us to construct logical forms for interpretation that capture the motivating data. The data our logic should address are those that arise from the process of combining meanings. In general this means building a logical form for an entire discourse, thus combining both the composition logic and the glue logic; we will concentrate on the composition logic here—leaving the interactions with discourse contexts for another time (see also Lascarides and Copestake 1995).

Our logic will extend the lambda calculus for functional types with rules for manipulating complex types like dot objects and QQC's. We'll also assume that reasoning with our background lexical axioms obeys the logical rules conjunction introduction and elimination and Modus Ponens (though not necessarily any introduction rules for the conditional nor any rules for negation). This is not the only way one could go about implementing a composition logic to account for GL intuitions. One could simply write down a little theory about the dot objects and other complex types— which is presumably what is in commonsense metaphysics (maybe it's not such a little theory either). But as we have argued in the introduction, there are conceptual and computational advantages to making the distinction. Building logical forms is different and easier than grasping their full content. We want an efficient process for logical form construction and thus one that is distinct from deep understanding which is given by the proof theory for the logical forms themselves. Our logic of the lexicon and of logical form construction at the clause level is like that of unification and other forms a logic of types; its complexity is no worse than simple unification, given that its operations are all driven by type adjustments and information about types in the lexicon.

4.1 The Type Language

The Set of Types

- e the general type of entities (with lots of subtypes) and $\underline{t}$ the type of truth values.

- If σ and τ are types, then so is $(\sigma \multimap \tau)$
- If σ and τ are types, then so is $(\sigma \bullet \tau)$.
- If σ and $\tau_1, \cdots \tau_n$ are types, then so is $(\sigma \otimes (\tau_1 \cdots \tau_n))$.

We now implement the formal details of a system for assigning and manipulating types of terms in semantic composition. We detail two distinct languages. One is the language of the lexicon L that exploits both the object language OL (the semantic logical forms) as well as type assignments. We will assume OL to look something like compositional DRT or SDRT (Asher 1993), in that it will have:

- discourse referents or variables, x, x_1 etc.
- OL formulas φ built up in the usual way
- higher order variables for dynamic properties and quantifiers P, Q, R, X

Our language L extends OL by having in addition a set of terms for types α, β, etc. L formulas are formed from the following rules:

- OL formulas
- formulas of the form $x : \alpha$, and type(x) =type(y), where x, y are OL variables and α is a type. We will call these formulas *type assignments*.
- formulas of the form $(\chi \wedge \psi)$ and $(\psi \rightarrow \chi)$, where ψ and χ are L-formulas.

We define a *type context* to be a sequence of type assignments. The rules of our composition logic below will exploit pairs of OL formulas and typing contexts as well as L formulas. They'll tell us how to manage the typing contexts so as to resolve type conflicts and thus permit the construction of logical forms.

We have to manipulate type assignments, so here is some notation.

- $c - (t : a)$
- $c + (t : a)$

- $c * (t : a)$

- and + have the usual meanings. $c * (t : a)$ is defined to be that type assignment context that results from reassigning the type a to t in c, if t occurs in the domain of c. If t does not occur in c then $c * (t : a)$ just extends c with the assignment of a to t; i.e.; $c * (t : a) = c + (t : a)$. $c(t : a)$ simply means that the type assignment c includes the assignment of a to t.

Our type language, and the notion of a formula together with a typing context contains the information involved in a typed feature structure. And most of the work on coercion etc. has used the framework of typed FSs together with the operation of unification (Pustejovsky 1995, Copestake and Briscoe, etc.). So why bother to develop a new formalism? The operation of unification is an efficient way of representing the replacement of one element with another that is determined to be more specific via some partial ordering. But with coercion, copredication, etc. we must *transform* types during semantic composition. Such coercions can be captured via lexical rules as Copestake and Briscoe (1994) have shown; such rules allow us to rewrite given feature structures as new feature structures. But this approach has several drawbacks. First, these rules allow us to change feature structures in an arbitrary way, whereas for us coercion is precisely the exploitation of something already in the given type structure. Such lexical rules don't discriminate between destructive type shifts like grinding and the ampliative inferences that are part of logical metonymy and copredication. In the latter, we *add* information about objects that are the typical denotata of the expressions involved. In the system of type rules to be introduced below, these two types of rules will be distinguished. Logical metonymy and copredication involve ampliative rules like dot and tensor exploitation. Type structures with these rules aren't transformed; they are preserved but trigger the addition of new information to logical form. Finally, our framework allows a more flexible relation between world knowledge and the lexicon than that for unification. While head types are typically stable, we imagine that QQCs may be highly contextually dependent, and as such we may be able to form such types on the fly. Given the standard treatment of qualia etc., these are taken to be universal features in typed FSs and so are much more rigidly construed.

We'll appeal to rules in the lexicon that for a given predicate of one or more places place restrictions on the types of their arguments. These restrictions may take into account the types of other arguments. We will write these rules as simple implications. As we will only have typing restrictions involving two place restrictions, we give the two rules needed below

Type Constraints

$$\frac{\cdots S(x,y)\cdots, c(x:\alpha),\ (x:\alpha \wedge S(x,y)) \rightarrow y:\beta}{\cdots S(x,y)\cdots, c * y:\beta}$$

$$\frac{\cdots S(x,y)\cdots, c(y:\alpha), (y:\alpha \wedge S(x,y)) \rightarrow x:\beta}{\cdots S(x,y)\cdots, c * x:\beta}$$

These rules like the other rules below tell us that given the information above the line, we can rewrite the formulas or revise the typing contexts in the way indicated below the line. These rules tell us that given a relational predicate S in some logical form for which we have a typing rule like $(x : \alpha \wedge S(x,y)) \rightarrow y:\beta$, we can infer $y:\beta$ if we have $x:\alpha$.

4.2 Basic Type shifting rules

We now present our extension of the lambda calculus with type shifts. A lambda expression denotes a $\multimap$ type. All these rules should be understood as reduction rules and so giving rise to equivalent term expressions.

Merging Contexts:

$$\frac{\lambda x\phi, c[t,c']}{\lambda x\phi[t], (c^\wedge c')}$$

Application:

$$\frac{\lambda x\phi[t],\quad c(x:\alpha, t:\alpha)}{\phi[t/x], c - x:\alpha}$$

In the rules below, we suppose that ϕ a formula that is implied by one of the formulas ϕ, t in the proof premises.

Type Inheritance:

$$\frac{\lambda x\phi[t]c(x:\alpha),\ (y:\alpha\wedge\phi)\rightarrow y:\beta}{\lambda x\phi[t], c*x:\beta}$$

Examples of type inheritance are type generalization or e.g. grinding (which would be a default). We will allow in the type hierarchy rules only implication rules; so the calculation here will be very quick.

We now pass to type accommodation. **Type Accommodation —a default**

$$\frac{\lambda x\phi[t], c(x:\alpha, t:\beta),\ (y:\beta\wedge\phi\rightarrow y:\alpha}{\lambda x\phi[t], /c*(x:\beta)}$$

$$\frac{\lambda x\phi[t], c(x:\beta, t:\alpha),\ (y:\beta\wedge\phi)\rightarrow y:\alpha}{\lambda x\phi[t], /c*(t:\beta)}$$

Type accommodation takes care of what happens in unification, in which a supertype can unify with a subtype yielding the subtype as the result.

4.3 Dot Objects

We'll assume that all higher functional types are unselective until we get down to the individual variables (perhaps an unrealistic assumption). Below let ϕ be of general type $e^{n+1} \multimap \underline{t}$. We'll first look at a rule that allows us to exploit a $\bullet$ type during composition.

$\bullet$ Exploitation Left

$$\frac{\lambda x\phi[t], c(x:\alpha\bullet\beta, t:\alpha),\ \ (\lambda pS(x,y)(\phi)\wedge x:\alpha\bullet\beta)\rightarrow y:\alpha}{(\lambda y\lambda\vec{u}\ \exists x(\lambda pS(x,y)(\phi)[\vec{u}])[t],\ c+y:\alpha}$$

$$\frac{\lambda x\phi[t], c(t:\alpha\bullet\beta, x:\alpha),\;\; (\lambda p S(x,y)(\phi)\wedge x:\alpha\bullet\beta)\rightarrow y:\alpha}{(\lambda y\lambda\vec{u}\;\exists x(\lambda p S(y,x)(\phi)[\vec{u}])[t], c+x:\alpha}$$

- **Exploitation right** is similar.

These two rules say that given a dot object one can when we have a conflict of predication introduce another variable that has the type of one of the constituent types of the dot relating this new variable to the variable of dot type via a certain "coherence" relation S. We've written these rules as allowing any sort of modification of the original formula ϕ in the logical form. But in all of our simple applications one can just take the functional $\lambda p S(x,y)$ to be: $\lambda p(S(x,y)\wedge p)$. Thus, we're simply adding the coherence relation between complex type and its constituents to the logical form.

Let's look at a simple example of our rules at work: *the book is heavy.*

- the book: $\lambda P\exists x(\text{book}(x)\wedge P(x)), \langle P: e \multimap \underline{t}\rangle$
- is heavy: $\lambda u\text{heavy}(u),\;\; \langle u:p\rangle$
- The syntax dictates:
  $\lambda P\exists x(\text{book}(x)\wedge P(x)), \langle P: e\multimap \underline{t}, x:p\bullet i\rangle[\lambda u\text{heavy}(u),\;\; \langle u:p\rangle]$
- By Merging Contexts and Application,
  $\exists x(\text{book}(x)\wedge[\lambda u\text{heavy}(u)[x]), \langle x:p\bullet i,\; u:p\rangle$
- From the lexicon: $(\text{O-Elaboration}(x,y)\wedge y:p\bullet i)\rightarrow x:p$
- By $\bullet$ exploitation left, $\exists x(\text{book}(x)\wedge\lambda y\exists u(\text{heavy}(u)\wedge\text{O-Elaboration}(u,y))[x]), \langle x:p\bullet i, u:p\rangle$
- By importing $(\text{O-Elaboration}(x,y)\wedge y:p\bullet i)\rightarrow p:x$ from the lexicon again and by the rules for exploiting type constraints,
  $\exists x(\text{book}(x)\wedge\lambda y\exists u(\text{heavy}(u)\wedge\text{O-Elaboration}(u,y))[x]), \langle x:p,\;\; u:p,\; y:p\bullet i\rangle$
- By application:
  $\exists x(\text{book}(x)\wedge\exists u\text{heavy}(u)\wedge\text{O-Elaboration}(u,x)),\;\; \langle x:p,\rangle$

4.3.1 • Introduction

Whereas dot exploitation merely related together an argument of with a constituent type of a • type to facilitate application, we sometimes predicates like *read* coerce an argument into being a dot type. Following Pustejovsky (1998a), we'll say that predicates like *read actively select* for dot types of a certain kind (*passively selecting* predicates don't force such coercions throughout the structure).

Such actively selecting predicates can coerce the type of a term that has been already introduced, and so this calls for a more global type change than the previous exploitation rules. A formalism like compositional DRT or SDRT is useful here in that the construction formalisms for those languages allows us to locate contexts (DRSs or subDRSs) in which conditions involving the term whose type is to be coerced are first introduced. Let's suppose that we are looking at a type application $\lambda x\phi[t]$ within a potentially larger context Φ—e.g., Φ is a larger formula containing $\lambda x\phi[t]$. To capture the type coercion for the one new form of dot introduction we need, we'll add a condition relating variables of distinct types as before but at the point (local DRS or subDRS) where the term t whose type is coerced is first introduced. At the same time we will replace occurrences of with a variable of t's old type so as not to introduce type clashes within Φ. In effect, we're doing the same operation as dot exploitation, changing types while avoiding type clash, but at a global level. We'll write $\Phi^y_t(S(t,y))$ to signify this. This means that the retyped term t will inherit whatever quantifiational struture t had before the coercion; we'll assume that y is existentially closed off within its local context (DRS or subDRS)—we'll show this by putting $y^\exists$ in our replacement for Φ. So now we're finally ready to do dot introduction left:

dot introduction left :

Again let $\lambda x\phi$ be of a general type $e^{n+1} \multimap \underline{t}$.

$$\frac{\Phi[\lambda x\phi[t]], c(t:\alpha, x:\alpha\bullet\beta),\ (S(y,z)(\phi)\wedge y:\alpha\bullet\beta)\rightarrow z:\alpha}{(\Phi^{y^\exists}_t(S(t,y))[\lambda x\phi[t]],\ c*(t:\alpha\bullet\beta)}$$

Notice that as the change we are making is global and not local, we don't need to worry about the other lambda abstracts for ϕ; we presume that those will be appropriately dealt with in the context Φ. The other rule for dot introduction left is actually not needed; we have seen already in the second rule of dot exploitation the persistence of the dot type in the global context, which is what dot introduction is supposed to do. Dot introduction right will be just like dot introduction left.

We'll now turn to two examples. First just an example of dot introduction arising from composition of two verbs.

(23) John picked up and read three books.

We suppose that *read* must take a dot object as an argument and *pick up* must take a physical object. These can be expressed as $\rightarrow$ constraints. We'll simplify $lambdapS(x, y)$ in this case to $S(x, y)$ as the Lexical rules simply add here new information conjunctively.

- *and* links *pick up* and *read*:
  $\lambda P \lambda Q \lambda x(P(x) \wedge Q(x)),\ \langle \text{Type}(P) = \text{Type}(Q), x : \top \rangle$
- Applying the meaning of *and* to the meaning of *pick up*, we get:
  $\lambda Q \lambda x(\lambda y \text{Pickup}(y)[x] \wedge Q(X)),\ \langle \text{Type}(P) = \text{Type}(Q),\ Q : \text{verb-prop},\ y : p,\ x : \top \rangle$
- By type accommodation,
  $\lambda Q \lambda x(\text{Pickup}(x) \wedge Q[x]) \langle \text{Type}(P) = \text{Type}(Q), Q : \text{verb-prop},\ x : p \rangle$
- After applying the result to *read*,
  $\lambda x(\text{Pickup}(x) \wedge \lambda u \text{Read}(u)[x]) \langle x : p,\ u : p \bullet i \rangle$
- We have a type mismatch when trying to apply *read* to x. We could do dot exploitation on the bound variable in *read* but *read* demands a dot object as argument (*read the subway wall*). So by introducing $(\text{O-Elaboration}(v, y) \wedge y : p \bullet i) \rightarrow v : p$ from the lexicon and then by dot introduction
  $\lambda x \exists y(\text{Pickup}(y) \wedge \lambda u \text{O-Elaboration}(y, u) \wedge \text{Read}(u))[x]\ \langle u, x : p \bullet i \rangle$
- By importing $(\text{O-Elaboration}(y, v) \wedge y : p \bullet i) \rightarrow v : p$ from the lexicon and by type constraints,
  $\lambda x \exists y(\text{Pickup}(y) \wedge \lambda u \text{O-Elaboration}(y, u) \wedge \text{Read}(u))[x]\ \langle y : p,\ x, u : p \bullet i \rangle$

- Now we can do application, and we have:
  $\lambda x \exists y(\text{Pickup}(y) \wedge \text{O-Elaboration}(y, x) \wedge \text{Read}(x))\ \langle y : p,\ x : p \bullet i\rangle$

- We can now combine this with the object DP:
  $\exists x(\text{book}(x) \wedge |x| = 3 \wedge \exists y(\text{Pickup}(y) \wedge \text{O-Elaboration}(y, x) \wedge \text{Read}(x))\ \langle y : p,\ x : p \bullet i\rangle$

Our derivation tells us something about what it is to pick up and read three books; if we assume that $p \bullet i$ objects are individuated on the basis of their contents, then picking up and reading three books involves picking up and reading three (content-wise) different books. At least such a reading is possible for $p \bullet i$ objects, whereas it remains a mystery for merely physical objects.

The example from our motivating data concerning *read the subway wall* offers another nice example of $\bullet$ introduction.

(17a) Mary <u>stared at</u> the subway wall.

(17b) Mary <u>read</u> the subway wall.

By using $\bullet$ introduction on *subway wall*, we get the right reading for (17b).

4.4 The Quantificational Puzzle Revisited

Our second example consists of another visit to the quantificational puzzle that was one principal motivation for our approach to complex types. Here is the example again:

(13) a. The student read every book in the library.
     b. The student stole every book in the library.

Though our definitions assume a DRT or SDRT formalism, we can "forego the boxes" with these simple examples and remain with ordinary logic.

- the library: $\lambda P \exists x(\text{Library}(x) \wedge P(x)), \langle P : e \multimap \underline{t},\ x : p \bullet l\rangle$

- in the library: $\lambda Q \lambda y \exists x(\text{Library}(x) \wedge \text{in}(x, y) \wedge Q(y)), \langle Q : e \multimap \underline{t},\ x : p \bullet l,\ y : p\rangle$

- book in the library: $\lambda y \exists x(\text{Library}(x) \wedge \text{in}(x, y) \wedge \lambda v \text{book}(v)[y]), \langle x : p \bullet l,\ y : p,\ v : p \bullet i\rangle$

- by importation from the lexicon, $\bullet$ exploitation, type constraints, and application,
$\lambda y \exists x \exists v(\text{Library}(x) \wedge \text{in}(x, y) \wedge \text{O-Elaboration}(y, v) \wedge \text{Book}(v)),\ \langle x : p \bullet l,\ v : p \bullet i,\ y : p\rangle$

- read every book in the library:
$\forall y(\exists x \exists v(\text{Library}(x) \wedge \text{in}(x, y) \wedge \text{O-Elaboration}(y, v) \wedge \text{Book}(v)) \rightarrow \lambda u \text{read}(u)[y]),\ \langle x : p \bullet l,\ y : p,\ p : v,\ u : p \bullet i\rangle$

- Again since *read* must take a dot object as argument, we do dot introduction, which introduces w existentially bound in the antecedent of the conditional while changing the type of y to allow for application.
read every book in the library:
$\forall y(\exists x \exists v \exists w(\text{Library}(x) \wedge \text{in}(x, v) \wedge \text{O-Elaboration}(w, v) \wedge \text{Book}(v) \wedge \text{O-Elaboration}(w, y))$
$\rightarrow \lambda y \text{read}(u)[y]),\ \langle v, y : p \bullet i,\ x : p \bullet l,\ w : p\rangle$

- After application we get the desired quantificational reading for the VP. We are quantifying over $p \bullet i$ objects, not physical objects only.

Contrast this with *steal every book*. The derivation is the same down the last step above. We don't need to do dot introduction. Again the quantificational reading desired is obtained.

4.5 Copredication Revisited

Let's quickly look again at co composition and copredication, two more

Copredication examples are similar to the simple conjoined verb example that we looked at above. Conjoined lambda terms whose bound variables have different types α and β are applied serially to their argument. Such copredications succeed only if the argument is of the right dot type and gives rise to a coherent discourse (Lascarides, Copestake and Briscoe 1996).

(24) a. The bay curves from the lighthouse to a sandy spit and has lots of shoals.

b. ??The bay is polluted and curves from the lighthouse to a sandy spit.

Treating (24a):

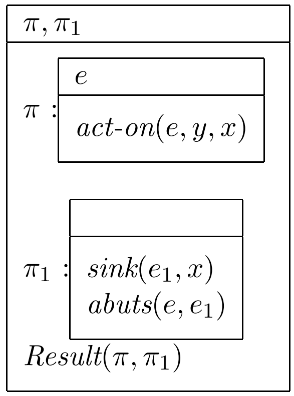
- We have to combine the two VPs; they are of incompatible types, but by dot introduction we can combine them to have a common variable of type line • location.
- from the lexicon:
  $(x : \text{line} \bullet \text{location} \wedge \text{Figure}(y, x)) \rightarrow y : \text{line}$
  $(x : \text{line} \bullet \text{location} \wedge \text{Ground}(y, x)) \rightarrow y : \text{location}$
- combining the two VPs via dot introduction:
  $\lambda y(\exists v(\text{Figure}(y, v) \wedge v$ curves from the lighthouse to a sandy spit$) \wedge (\exists x Ground(y, x) \wedge$ x has lots of shoals$) \langle v : \text{line} x : \text{location}, y : \text{line} \bullet \text{location} \rangle)$.
- $\|$the bay$\|$: $\lambda P \exists x(\text{Bay}(x) \wedge P(x)), \langle P : e \multimap \underline{t}, x : \text{line} \bullet \text{location} \rangle$

The types of the subject DP and the conjoined VP are now compatible and can combine via application. We get a coherent discourse because giving the shape of the bay is a good background information that helps locate it and make it more determinate. Shifting the predication around as in (24b) is not as good, because the locating purpose of the Background is now otiose.

4.6 A more complex application of dot types: Unaccusativity

Causative verbs type their event arguments as being of complex type. In effect they introduce two eventuality types one of which is the Result of the other. This invites us to use an SDRT formalism for describing their contributions to logical form:

(25) $\lambda e_1 \lambda x \lambda y$



The type of the "head" eventuality (the one abstracted over here) is that of process • result. In contexts like (26a), both subject and object arguments are filled in and we have a small SDRS with an underspecified causing event, which can be made more precise via an "anaphoric subsumption" from later discourse as in (26c) or by a PP (26d).

(26)
a. The enemy sank the boat.
b. The boat sank.
c. The enemy sank the boat. They torpedoed it.
d. The enemy sank the boat by bombing.
e. The boat sank. The enemy torpedoed it.

On the other hand, a sentence like (26b) seems to change the type of the verb. Suppose that a causative verb constructs a VP with no (internal) subject. Then at the tense node T we will introduce an event of type unaccusative (Result) with a different argument structure that must combine with the causative event. We now have a classic type clash problem that we can resolve with dot exploitation. But we need a more complex rule from the lexicon that tells us we need to "add" an operator to our LF and apply it to the condition introduced by the causative; the operator existentially closes off the subject argument. This is an example of a higher order addition to LF that goes well beyond the first order additions given so far but that our general rule for dot exploitation allows for. The crucial lexical axiom looks like this:

(27) $(\lambda p \lambda e' \exists y p[\phi(e)][e'] \wedge e : \text{process} \bullet \text{result}) \rightarrow e : \text{unaccus.}$

With this generalized dot exploitation, we get the right transformation to give coherent truth conditions to (26b). But notice that our underspecified process is still there in logical form and can be picked up and made more precise later on in discourse—as in (26e).

5 Quasi Qualia Constructs

We now turn to analyzing GL's qualia structures as complex types—QQCs or quasi qualia constructs. We have two different sorts of exploitation rules for QQCs. QQC left exploitation allows us to coerce from a QQC type of the form $\alpha \otimes (\cdots \beta_i \cdots)$ to the type α. QQC left exploitation is just like dot exploitation and so we won't write down the rule here. QQC right exploitation is, however, different. QQC elmination right has to be a default rule and hence lead only by default to a revision of the type assignment (noted with the slash), because it can be overridden in context—witness (3). We'll here just use the simplified "first order" conjunctive case of our general exploitation format from • exploitation below to simplify matters.

QQC Exploitation right$_i$:

$$\frac{\lambda x \phi[t], c(x : \alpha \otimes (\cdots \beta_i \cdots), t : \beta_i),\ (S(z,y) \wedge z : \alpha \otimes (\cdots \beta_i \cdots)) \rightarrow /y : \beta_i}{\lambda x \lambda \vec{u}\ \exists y(/S(y,x) \wedge \phi[y/x][\vec{u}])[t],\ c + (y : \alpha \otimes (\cdots \beta_i \cdots))}$$

$$\frac{\lambda x \phi[t], c(t : \alpha \otimes (\cdots \beta_i \cdots), x : \beta_i),\ (S(z,y) \wedge z : \alpha \otimes (\cdots \beta_i \cdots) \rightarrow\ y : \beta_i}{(\lambda x \lambda \vec{u}\ \exists y(/S(x,y) \wedge \phi[y/x][\vec{u}])[t],\ c + (/y : \beta_i)}$$

Similarly there is a QQC introduction rule that is just an instance of the dot introduction rule.

To see how these rules work we now turn to some of our motivating examples. Below we'll make some assumptions about QQC types. Recall that $\otimes_T$ is a complex QQC with a type that plays a telic role on the right, while $\otimes_A$ has an agentive while $\otimes_{A,T}$ contains a sequence of an agentive and telic on the right.

- *book, novel, pamphlet,* etc.: e $(p \bullet i) \otimes$ (write,read).
- *sonata*: $(p \bullet i) \otimes_{A,T} (compose,\ play)$.
- *class*: *people* $\otimes_T$ *teach.*
- *cigarette* $p \otimes_T$ *smoke.*
- *wine* liquid $\otimes_T$ *drink.*
- all artifactual objects (physical and informational) inherit a general dependent type that gives their cause; i.e.,
  $(x : \text{Object} \otimes_{A,T} (\cdots))$
- all objects that are extended are typically visible, and all objects that involve sound are typically audible. This gives rise to two more general QQC types: $p \otimes hear$ and $p \otimes see.$
  - $x : \text{door} \rightarrow x : p \otimes see.$
  - With the dependent type on sonata *play*, we can classify sonatas as objects that are of audible type; i.e.,$x : \text{sonata} \rightarrow x : p \otimes hear.$

We note that *play* in the above is itself a complex dot relation like *read*: *play* has the type *read* $\bullet$ *make sound*, with Result being the coherence relation that holds between the two propositions that result when the relations are applied to the appropriate arguments. Further, we note that most physical objects don't have such specific QQCs—e.g., *bathroom, door, garden*, etc.

To analyze our examples concerning $p \bullet i$ objects, we'll assume the following lexical rule for exploiting QQC types:

$$x : (p \bullet i \otimes (\tau_1 \cdots \tau_n) \wedge y : \text{ev} \wedge \text{Telic}(x, y)) \rightarrow /y : \text{read} \qquad (28)$$

Finally, we need to say something about the defeasible inferences when we do QQC exploitation right. As in many applications of defaults and nonmonotonicity, not all of the defaults are equal; some have precedence over others. Again as in many applications of nonmonotinicity, we prefer the more specific default in cases of conflict. So we will suppose an ordering of defaults, which is the transitive closure of $\preceq$ defined below; the relation is isomorphic to the subtyping relation on complex types.

- $(x:\alpha \rightarrow x:\beta) \rightarrow \alpha \preceq \beta$
- $\beta \preceq \gamma \rightarrow ((\alpha \bullet \beta) \preceq (\alpha \bullet \gamma)$
- $\beta \preceq \gamma \rightarrow ((\beta \bullet \alpha) \preceq (\gamma \bullet \alpha)$
- $\beta \preceq \gamma \rightarrow ((\alpha \otimes (\cdots \beta \cdots)) \preceq (\alpha \otimes (\cdots \gamma \cdots)))$
- $\beta \preceq \gamma \rightarrow ((\beta \otimes (\cdots \alpha \cdots)) \preceq (\gamma \otimes (\cdots \alpha \cdots)))$

Defaults like QQC exploitation right are tied to a particular QQC type in their antecedents; the more highly preferred defaults will be those tied to the more specific subtype. QQC exploitation right rules are also ordered with respect to their application. For any QQC of the form $\alpha \otimes (\beta_1, \cdots \beta_{n-1}, \beta_n)$, we prioritize the QQC exploitation rule for the rightmost; i.e. $\beta_n \preceq \beta_{n-1} \preceq \cdots \preceq \beta_1$. QQC exploitation rules then will proceed with respect to a particular QQC type and then the ordering of the various QQC exploitation right rules will be determined by the sequence of QQC types. As in Lascarides et al. (1996), we'll account for blocking by having a default of a more specific type takes priority over a default of a less specific type.

5.1 Logical Metonymy Revisited

In this section, we return to our motivating examples concerning *enjoy.* Consider

(29) a. Sheila enjoyed the book.
b. Sheila enjoyed the sonata.
c. Sheila enjoyed the garden.

We will analyze these examples to illustrate how to deal with data having to do with QQC types. Similar analyses would apply to the adjectival examples like *quick cigarette.*

First, let's consider (29a) using our previous assumptions.

- $\exists x(\text{Book}(x) \wedge \lambda u \text{enjoy}(u)[x]),\ \langle x : (p \bullet i) \otimes (\text{wr, rd}),\ u : \text{ev}\rangle$

Since *book* is a QQC type. We will do QQC exploitation on the lowest or most specific type—namely, $p \bullet i \otimes (\text{wr, rd})$. And also on the rightmost QQC exploitation. But first we have to do Type accommodation to on the object of *enjoy* to get the right premises:

- $\exists x(\text{Book}(x) \wedge \lambda u \text{enjoy}(u)[x]),\ \langle x : (p \bullet i) \otimes (\text{wr, rd}),\ u : \text{rd}\rangle$
- Now we import from the lexicon:
  $(\text{Telic}(u, e) \wedge u : (p \bullet i) \otimes (\text{wr, rd})) \rightarrow /e : \text{rd}$
- Now by QQC exploitation:
  $\exists x(\text{Book}(x) \wedge \lambda v \exists e(/\text{Telic}(v, e) \wedge \text{enjoy}(e)[x]),$
  $\langle x : (p \bullet i) \otimes (\text{wr, rd}), u : \text{ev},\ /e : \text{rd}\rangle$

Now the type constraints allow us to type v as the appropriate tensor type and we can do application:

- $\exists x(\text{Book}(x) \wedge \exists e(/\text{Telic}(x, e) \wedge \text{enjoy}(e)),\ \langle x : (p \bullet i) \otimes (\text{wr, rd}),\ /e : \text{rd}\rangle$

Thus, we derive for (29a) that by default Sheila enjoyed reading the sonata. Following very similar inferences, we predict (29b) to mean that Sheila enjoyed playing the sonata. This default can be overridden if we know that Sheila doesn't play sonatas; we then exploit the more general QQC construct, $p \otimes hear$. It might be in our metaphysical scheme that we have a general default that people normally don't create either sonatas themselves (i.e., compose them) or physical performances of them (i.e., play them)—only abnormal people, musicians, do. We would then get a "Nixon diamond": since the defaults tied to *sonata* and to the human agent aren't ordered with respect to $\preceq$, our default system modelled on Lascarides et al. (1996) won't yield a conclusion. However, while we can't draw a conclusion from these two defaults, we can exploit the more general default for objects that emit a sound and thus get the reading that Sheila enjoyed listening to the sonata.

Let's now turn to (29c). Because *garden*, like *door* lacks a particular, conventionalized telic QQC, we predict simply that (29c) means only that Sheila enjoyed looking at the garden:

- $\exists x(\text{Garden}(x) \wedge \lambda u \text{enjoy}(u)[x]),\ \langle x : p,\ \text{ev} : u\rangle$

- From our lexical assumptions, x : garden $\rightarrow x : p\otimes$ see. So by Type Inheritance:
- $\exists x(\text{Garden}(x) \wedge \lambda u\text{enjoy}(u)[x]),\ \langle x : p \otimes \textit{see}\ u : \text{ev}\rangle$

Now the derivation is the same as for *enjoy the book*. A similar analysis would hold of *enjoy the door*. We would say that there is no special telic QQC associated with *door* or with *garden*. However, it appears that in some cases, we can coerce such such a dependent telic type with QQC introduction verbs like *use*.

(30) Cathie used the door.

(30) means that Cathie used the door for its intended purpose (e.g., as an aperture). Many light verbs (*have, take, give*) are verbs that coerce their objects into something of QQC type; like the verbs that coerce their objects into things of • type, however, the set of QQC introducing constructions is restricted.

There is an interesting contrast between (29b) and

(29d) Alexis began the sonata.

Unlike *enjoy*, *begin* coerces a non-event object argument into an event that results in modifying or creating the object itself or some instance of it. This selectional restriction has to be codified in a lexical rule (we'll use the predicate 'act-on' to express this restriction):

$$(\lambda e\text{Begin}(e, e')[x] \wedge x : p) \rightarrow \forall e(\text{Begin}(e, e') \rightarrow \forall e(\text{Act-on}(x, e) \wedge \\ \forall y(\text{Agent}(e, y) \leftrightarrow \text{Agent}(e', y)))$$

This rule constrains further compositions. If we have:
$\text{Begin}(e, e')$,
then we must have:
$\text{Act-on}(x, e)) \wedge \text{Agent}(y, e) \leftrightarrow \text{Agent}(y, e')$.
This establishes a constraint on all defaults using QQC exploitation. Now given the type structure,

$$\lambda e\lambda e'\text{Begin}(e, e')[x], \langle x : (p \bullet i) \otimes \text{compose, play},\ e, e' : \text{ev}\rangle$$

we will, in conjunction with the background lexical rule for *begin*, infer:

$$\lambda y \lambda e' \exists e(\text{Act-on}(e, y) \wedge \text{Begin}(e, e'))[x], \langle y : p \bullet i,\ e', e : \text{ev}\rangle$$

Now we can here exploit one of the two events in the QQC type; either *play* or *compose* are types of acting-on events; they both bring either the sonata or a physical instance (performance) of it into existence. We cannot infer here that Alexis began to listen to the sonata, because listening is not the right sort of eventuality given our special lexical rule for *begin*. Our general default that most people don't act-on sonatas here is overridden by our lexical rule as well; thus, we will pick up perforce for (29d) some eventuality in which the subject modifies the sonata. Given our preference for the rightmost QQC type, we will prefer by default the interpretation that Alexis began to play the sonata, though with other contextual information we could easily get that Alexis began to compose the sonata. Thus, we end up with for the VP of (29d):

$$\lambda y \lambda e' \exists e(\text{Act-on}(e, y) \wedge \text{Begin}(e, e'))[x], \langle y : p \bullet i,\ e' : \text{ev},\ e : \text{play}\rangle$$

One last comment is that our analysis of QQC types keeps all information around, and we predict that copredication is possible with QQC types. So unlike the account of Briscoe et al. where the verb *enjoy* is coerced or in Pustejovsky (1995) where the NP denotation is coerced (destructively), we can make sense of

(31) a. Sheila enjoyed the book and the sonata.
b. Cathie carried home and enjoyed the book.

5.2 Cocomposition Revisited

Let's return briefly to some well-known examples concerning *bake*.

(19) a. The sun baked the rock.
b. James baked the potato.
c. James baked the cake.

In (19c) we see an interaction between the verb's meaning and that of its complement. More generally, agentive QQCs may help specify the type of an eventuality argument of a verb, which leads us to the following lexical rule for process•result verbs.

$$(\lambda z \lambda e \phi[x] \wedge x : \tau \otimes_A \phi \wedge e : \text{process} \bullet \text{result}) \rightarrow e : \text{process} \bullet \text{created}(x)$$

With type inheritance, we will now infer for (19c) that baking the cake is an event of type process • created(x), where x is the cake. By the semantics of the finished logical form in which we have a Result relation between the process and the creation of the cake, we may then infer that the cake was created during the baking event (Asher forthcoming). Thus, in this example the QQC type of the complement affects the meaning of the verb but it doesn't destroy the earlier typing information on the variable; it's consistent with the earlier information, because we realize that the creation of an object is a type of result.

On the other hand in (19b), there is also a shift from *potato* as a natural kind or simple type to the QQC type, $p \otimes_{T_i} (\cdots \text{eat}_i \cdots)$. That is, once cooked a potato has as its natural end its consumption; thus, *bake* transforms a natural kind here into a food. The type transformation involved here is a QQC introduction rule, but we need here two lexical rules to get the right conditions to do QQC introduction.

- $(\lambda u \text{Bake}(u)[x] \wedge x : \text{plant}) \rightarrow (\exists y \text{Cooked}(x, y)$
- $\text{Cooked}(x, y) \wedge y : \text{plant} \otimes \text{eat} \rightarrow x : \text{plant}.$

We exploit the first axiom to get the appropriate consequences of applying *bake* to a plant. Then we use QQC introduction to get the appropriate typing that potato when baked becomes a food (plant $\otimes$ eat). For the whole VP (ignoring the subject argument) we get:

$$\exists y(\text{Bake}(x) \wedge \text{Cooked}(y, x)),\ \langle y : \text{plant}, x : \text{plant} \otimes \text{eat} \rangle$$

The last kind of change operative in these examples is one that has to do with intentionality. Verbs like *bake* and *sink* and any verbs in which we have a substantial modification or creation of the object in the complement seem to require intentional readings when they have intentional agents as subjects. In those cases then, we have another lexical rule like that given by *read*. We can formalize this requirement as a case of QQC type introduction which transforms the conditions introduced by the verb in a rather complex way. Suppose that ϕ is any verb that implies an *act-on* eventuality which selectionally restricts its subject to be physical. Then we have the following background lexical axiom, which requires an SDRS notation, as it introduces a discourse relation Result between two constituents. Let $S(z, y)$ is the coherence relation that obtains between the physical aspect of agents and their complex type.

- $(\lambda p \lambda z[\pi, \pi' | \pi : [|\text{intend}(z, {}^{\cap}[u|p(u), S(u, z)],$
  $\pi' : [u|p[z], S(z, u)]],$
  $\text{Result}(\pi, \pi')[\lambda v \phi] \wedge x : p \bullet \text{intentional} \rightarrow z : \top$

So at the end we get the following using QQC introduction for (19b):

(32)
$$\begin{array}{|l|}\hline \pi, \pi_1 \\ \hline \pi : \begin{array}{|l|}\hline y, \pi_2 \\ \hline \text{James}(y) \\ \textit{intend}(y, \pi_2) \\ \pi_2 : \begin{array}{|l|}\hline z \\ \hline \text{potato}(z) \\ \text{bake}(u, z)) \\ \hline \end{array} \\ S(y, u) \\ \hline \end{array} \\ \\ \pi_1 : \begin{array}{|l|}\hline z \\ \hline \textit{bake}(u, z) \\ \hline \end{array} \\ \\ \textit{Result}(\pi, \pi_1) \\ \hline \end{array} \quad \langle u : p, z : \text{plant} \otimes \text{eat}, y : p \bullet \text{intentional} \rangle$$

This gives the cocomposition in (19b) as resulting from two QQC introduction rules. Similarly, we could reconstruct the cocomposition in (19c).

6 Further Topics: Causatives, Experiencer Verbs and Discourse Structure

We have concentrated largely on dot types among noun denotations up to now. As we have seen, verbs like *read* require as complements dot objects; and in fact *read* does so because it itself is a complex type. The complex type of the denotation of *read* is similar to that for process•result verbs. It involves two action types, one a process involving a physical object (processing the print on the page/screen) that results in a second process involving the grasping of the informational content. So we have a dot object involving two eventuality types related by the coherence relation, Result. In effect, we distribute the dot object complement across the dot object verb type by two cases of dot exploitation.

Many verbs denote complex types of eventualities. As we have seen, all unaccusatives and causative verbs denote complex relations. All these verbs roughly denote a complex type like that denoted by *bake*.

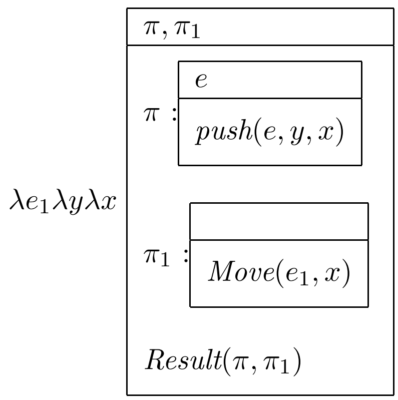
(33)

$$\lambda x \lambda e \begin{array}{|l|}\hline \pi, \pi_1 \\ \hline \pi : \begin{array}{|l|}\hline y \\ \hline \mathit{act\text{-}on}(e,y,x) \\ \hline \end{array} \\ \\ \pi_1 : \begin{array}{|l|}\hline \mathrm{s} \\ \hline \mathit{baked}(s,x) \\ \hline \end{array} \\ \\ \mathit{Result}(\pi,\pi_1) \\ \hline \end{array}$$

Such verbs allow for intentional interpretations, when a dot object with intentionality is the subject of such verbs, as we have seen.

Verbs like *push* and other verbs that have only a prepositional phrase construction for dative might have a similar structure to causatives (e.g. cf Krifka 1999). But unlike causatives like *bake* or *sink*, we have an underspecified result action, which we'll call *Move*:

(34)



Many lexical items like *push* introduce *underspecified* relations on event types. We can exploit these in determining discourse relations. Suppose for instance that *fall* is a verb which in effect introduces a unified event structure (we might also take it to be unaccusative and so also have underspecified cause). The discourse relation Explanation in the *push/fall* example (2b) is inferred through anaphora resolution. To find a suitable antecedent for the type of eventuality e in the entry for *push*, we need to find a discourse constituent that we can identify with π, which in this case means a discourse constituent that is an Elaboration or a specification of π.[4] And John's falling described in the preceding clause fits the bill. Once we have done that, we have the discourse relation between the constituents from the lexical entry. Of course in the *push fall* example, we want to infer Explanation between the falling and the pushing, but that we can do since the causal relation inferred from Result is sufficient to infer Explanation. If we take *fall* to be an unaccusative, then we will unify both pairs of constituents in the lexical representations. In the simple cases considered in Asher and Lascarides there's only one candidate for a resolving constituent. Further, this is quite general. Of course, if there are more, life (and linguistics) become more complicated. In that case, we have to use world knowledge to find the most likely candidate. We put the following forward as a tentative hypothesis: in many cases where the lexicon helps us decide which discourse relations relate two objects, it is because these relations are already built into the lexicon and discourse relations come about as a result in effect of anaphora resolution.

[4] It should be noted that this type of anaphora is different from pronominal anaphora; if we cannot find an antecedent, we simply existentially quantify over the '?' to get a coherent logical form.

We can use the similarly underspecified process of many causative or experiencer verbs to make similar discourse links. Let us consider psych verbs like *frighten, scare, anger, annoy.* Such verbs involve some underspecified action on the experiencer resulting in a psychological state (cf. Pustejovsky, 1995). These form a complex type very similar to the causatives.

(35) $$\lambda y \lambda x \lambda e_1 \begin{array}{|l|}\hline \pi, \pi_1 \\ \hline \pi : \begin{array}{|l|}\hline e \\ \hline \mathit{Exp}(e,x,y) \\ \hline \end{array} \\ \\ \pi_1 : \begin{array}{|l|}\hline \ \\ \hline \mathit{annoyed}(e_1, x) \\ \hline \end{array} \\ \\ \mathit{Result}(\pi, \pi_1) \\ \hline \end{array}$$

With such entries, we invite further specification of the experiencing event. Pustejovsky (1995) discusses related cases where the qualia of the subject noun are exploited to identify the initial event. Notice, however, that such information is defeasible.

(36) That alarm annoys Mary. It rings too loud (It never rings).

Similarly, if the discourse specifies an event of this type, then we can identify that constituent with π. This is very natural in cases like:

(37) John annoys Mary. He never calls (he always calls).

More generally, any causative verb introduces a complex event structure modelled as an SDRS consisting of a description of a causing event and a description of a result state linked by Result) (Danlos 1998, 1999).

(38) John hit the carafe against the sink. He shattered it.

(39) John hit the carafe against the sink. He shattered it though, only when he dropped it.

Let us now contrast this with *intentional* verbs. Curiously, these verbs don't license explanatory relations very easily:

(40) a. John fetched the children from school. They wanted to go home.

b. John went to fetch the children. They wanted to go home.

c. John got up. Fred greeted him.

d. ?John met Mary at the paper store. He saw her going down the street and then followed her there.

e. John met Mary at the paper store. He had seen her going down the street and then had followed her there.

Although in (40a) the children's wanting to go home could be a perfectly acceptable cause for John's coming to get them, we don't find the Explanation relation to be one that holds there. At least it's quite plausible to understand that discourse as a Narrative sequence, in which we describe the children's actions from John's perspective. It seems that the verb *fetch* has a strong intentional component that licenses this intentional reading of the next clause. More importantly for our purposes here, however, intentional verbs have particular causes in the lexicon; or rather they give their own causes: namely, the volition of the agent (we suggest here rather tentatively). *Get up* similarly describes an intentional action and doesn't license the Explanation relation in (40c), and an attempt at constructing an Explanation in (40d) produces an ill formed discourse. By putting it second sentence in the pluperfect, however, coherence and Explanation are restored. Our tentative explanation is that intentional verbs supply at least by default their own causal conditions—the agent's volition. And so only constituents that supply further information about the agent's cognitive state license an Explanation relation.

We believe that by extending one's examination of the functions of words to discourse, we may see many verbs as introducing complex types whose subtypes can be further specified in subsequent discourse. For instance, there are some time-honored arguments to the effect that *kill* cannot be defined as *cause to die* in spite of the initial plausibility of the definition. For (41) seems to be nonsense.

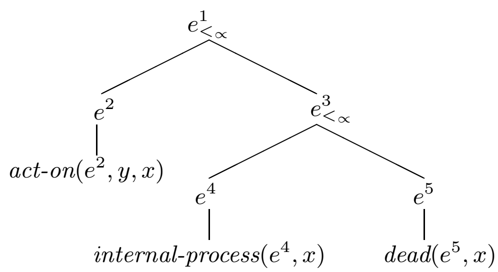
(41) Kim killed Sandy on Tuesday by stabbing him on Monday.

But we might be able to maintain that *kill* does in fact introduce a complex type and still be able to explain the ungrammaticality of (41, if there is only one event within the complex type denoted by *kill* that can be modified through composition:[5]

---

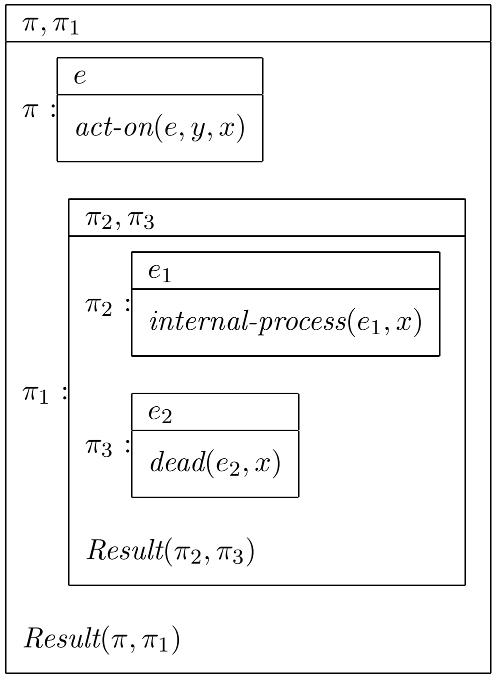
[5] Using the notation standard in GL, the SDRS below corresponds to the following annotated event structure:

(i)



There are two differences from the analysis given for causatives such as *kill* in Pustejovsky (1995), for example. First, *act-on* is here given as a relation between individuals, rather than between an individual and an event. Secondly, the internal structure of the event is richer than that assumed for most transitional verbs; in essense, this is the Pustejovsky (1988) analysis for how some causatives entail the event semantics of the associated inchoatives by incorporating the inchoative event structure in its own. From considerations of how discourse anaphora may make reference to facets of subevent structure for coherence purposes, Danlos (1999) reaches very similar conclusions.

(42) $\lambda e_1 \lambda x \lambda y$



The manner adverbial *by stabbing him on Monday* would have to modify the head eventuality e_1*, which would result in something incoherent (cf. Pustejovsky, 1995, for definition of event headedness). But in addition, our complex type definition could capture the discourse structure of (43), as pointed out by Danlos (1999) using the same mechanisms alluded to above:

(43) Kim killed Sandy. He stabbed him on Monday, and Sandy died of a hemorrhage on Tuesday.

This discourse seems OK to us and it also seems clear that the stabbing and hemorrhaging are elaborations of the killing.

In closing, we would like to speculate on further extensions of this work. We have only briefly looked at verbs with a causative structure here, but there are other places to look for discourse structure. For instance, a verb like *buy* may introduce in fact a complex type, in which one type of eventuality serves as a Background to the other. And the same anaphoric mechanisms for further specifying these types that we referred to earlier and discussed by Danlos might apply here:

(44) Kim sold her truck. Sandy bought it.

By merging concerns of the lexicon with those of discourse interpretation together, we can explore these hypotheses further.

7 References

Asher, Nicholas. forthcoming. The Metaphysics of the Text from a Formal Point of View. Wilks and Umberto Eco eds., *Categories Linguistic and Ontological*, Kluwer Academic Publishers.

Asher Nicholas. 1993. *Reference To Abstract Objects in Discourse*, Kluwer Academic Publishers.

Asher, Nicholas and Alex Lascarides. 1995. “Lexical Disambiguation in a Discourse Context”, *Journal of Semantics*, **12** number 1, pp69–108, Oxford University Press.

Asher, Nicholas, Michel Aurnague, Myriam Bras, Laure Vieu [1996]. “De l'espace-temps dans l'analyse du discours,” *Semiotique : Numero Special ”Theories sémantiques et modalisation*, **9**.

Asher, Nicholas and Tim Fernando. 1997. ”Effective Labeling for Disambiguation”, *Proceedings of the Second International Workshop in Computational Linguistics*, Tilburg, the Netherlands.

Asher, Nicholas and Alex Lascarides. 1998. “The Semantics and Pragmatics of Presupposition”, manuscript.

Bouillon, Pierrette. 1997. *Polymorphie et sémantique lexicale: le cas des adjectifs*. Lille: Presses Universitaires du Spetentrion.

Busa, F. (1996). *Compositionality and the Semantics of Nominals*, Ph.D. Dissertation, Brandeis University.

Busa, F., Calzolari, N., Lenci, A. and Pustejovsky, J. (1999). Building a Semantic Lexicon: Structuring and Generating Concepts. *Proceedings of*

IWCS-III, Tilberg, The Netherlands.

Climent, S. 1999. “Individuation by partitive constructions in Spanish,” in Bouillon P. and F. Busa (eds.) *The Syntax of Word Meaning*, Cambridge University Press.

Copestake, Ann and Ted Briscoe. 1995. “Semi-productive Polysemy and Sense Extension”, *Journal of Semantics*, 15.

Danlos, L. 1999. “Event Coherence in causal discourses. in P. Bouillon and F. Busa (eds.) *The Syntax of Word Meaning*, Cambridge University Press.

Godard, D. and J. Jayez. 1993. “Towards a proper treatment of Coercion Phenomena,” in *Proceeding of the 1993 European ACL*.

Fodor, Jerry and Ernie Lepore. 1998. “The Emptiness of the Lexicon: Critical Reflections on J. Pustejovsky’s *The Generative Lexicon*”, *Linguistic Inquiry* .

Hale, Kenneth and Samuel Kayser. 1993. ”On Argument Structure and the Lexical Expression of Syntactic Relations”, in eds. Hale, Kenneth and Samuel Kayser, *The View from Building 20*, MIT Press.

Hobbs, Jerry, M. E. Stickel, D. E. Appelt, P. Martin. 1993. ”Interpretation as Abduction”, *Artificial Intelligence*, **63**, pp69–142.

Lascarides, Alex and Anne Copestake. 1995. ”The Pragmatics of Word Meaning” *Proceedings SALT V*.

Lascarides, Alex and Nicholas Asher. 1993. Temporal Interpretation, Discourse Relations and Commonsense Entailment, in *Linguistics and Philosophy*, **16**, pp.437–493.

Levin, Beth. 1985. “Introduction,” in B. Levin, (ed.), *Lexical Semantics in Review*, Lexicon Project Working Papers 1, Center for Cognitive Science, MIT, 1-62.

Levin, Beth. 1993. *Towards a Lexical Organization of English Verbs*, University of Chicago Press, Chicago.

Levin, Beth and Malka Rappaport Hovav. 1995. *Unaccusatives: At the Syntax-Lexical Semantics Interface*, MIT Press, Cambridge, MA.

Moravcsik, Julius. 1975. "Aitia as Generative Factor in Aristotle's Philosophy," *Dialogue*, 14, pp. 622-36.

Pustejovsky, James. 1995. *The Generative Lexicon*, MIT Press, Cambridge, MA.

Pustejovsky, J. 1995a. "Linguistic Constraints on Type Coercion," in P. Saint-Dizier and E. Viegas (eds.), *Computational Lexical Semantics*, Cambridge University Press.

Pustejovsky, James. 1997. "Multiplicity of Meaning," ms., 1997 Gustaf Stern Memorial Lectures, Gothenburg, Sweden.

Pustejovsky, James. 1998a. "The Semantics of Lexical Underspecification," *Folia Linguistica* XXXII.

Pustejovsky, James. 1998b. "Generativity and Explanation in Semantics," *Linguistic Inquiry*.

Pustejovsky, J. 1998c. "Underspecification and Argument Binding," in E. Viegas (ed) *Breadth and Depth of the Lexicon*, Kluwer, The Netherlands.

Pustejovsky, J. 1999. "Lexical Shadowing and Argument Binding," in Y. Ravin and C. Leacock (eds.) *Polysemy*, Oxford University Press.

Pustejovsky, J. and F. Busa 1997. "Denominal Verbs," ms. Brandeis University.

Verspoor, K. 1997. *Contextually Dependent Lexical Semantics*, Ph.D. Dissertation, University of Edinburgh.