The Syntax/Semantics Interface: Compositionality Issues*

Pauline Jacobson
Dept. of Cognitive and Linguistic Sciences
Brown University
pauline_jacobson@brown.edu

in Nick Bezhanishvili, Sebastian Löbner, Kerstin Schwabe and Luca Spada (eds.), *Logic, Language and Computation* (Selected papers from the 8th International Tbilisi Symposium on Logic, Language, and Computation, Tbilisi, Georgia, 2009), Berlin-Heidelberg: Springer-Verlag, 2011, pp. 249-270.

The Syntax/Semantics Interface: Compositionality Issues*

Pauline Jacobson
Dept. of Cognitive and Linguistic Sciences
Brown University
pauline_jacobson@brown.edu

0. Introduction

The claim that natural language semantics is a “compositional” system is often cast as the following slogan: *The meaning of a larger expressions is computed (in some predictable way) from the meanings of the smaller expressions of which the larger expression is composed.* While most (though certainly not all) researchers in natural language semantic theories believe that this is correct, it actually has little content until we decide just what *are* the parts of the larger expressions at issue here. In particular: what exactly is the object that is interpreted? As a theory of natural language semantics, this says very little (and is difficult to falsify) if the correspondence between actual (“pronounced”) expressions of natural language and the representation at which meaning is compositionally computed is not transparent. For whenever it appears that the actual language is not interpreted compositionally, we could save the compositionality hypothesis by positing that meanings are not actually compositionally computed on the actual (pronounced) sentences of the language but on some other representation whose relation to the sentence at question is quite indirect. Thus the focus of this paper is to raise (and suggest an answer to) the following question: Can we maintain that it is the actual (pronounced) expressions of a language which are compositionally interpreted? Put differently, if we see the syntax of a language as a system which “builds” (i.e., proves well-formed) expressions of the language, is it the case that any expression which is well-formed according to the syntax also has an interpretation? (Before proceeding, let me clarify that by “interpretation” I mean a model-theoretic interpretation – not just a representation – and I will thus be embedding the discussion in a theory that assumes a model-theoretic (as opposed to a proof-theoretic) semantics.)

The “yes” answer to the above question is what I will call the position of direct compositionality (hereafter, DC). This position claims that the syntax is a system of rules which proves expressions well-formed, often proving one expression well-formed on the basis of two or more other expressions. Since each expression proven well-formed in the syntax is assigned a meaning, it is an immediate corollary of this that one can view the semantics as working “in tandem” with the syntax; each expression is assigned a meaning as it is “built” in the syntax. This position was held or at least discussed in much work within classical “Montague Grammar” as done in linguistic semantics in the 1970s and 1980s. Note that this view also commits us to what we might call *they hypothesis of local interpretation*: every linguistic expression which is well-formed according to the syntax – no matter how small or “local” – has a meaning.

The “no” answer to the above question (or at least one version of a “no” answer) is that the semantics is indeed compositional but does not work directly with the syntax.

Rather, the syntax proves representations well-formed as syntactic objects, and these are then paired with another representation (“Logical Form” or “LF”), which is what is compositionally interpreted. The syntactic objects which are not themselves directly interpreted can be thought of as the “surface” or “pronounced” sentences (or, more generally, linguistic expressions). Thus these may be quite different from the representations which are interpreted.

The primary goal of this paper is to elucidate the kinds of tools which have been used by the Direct Compositional worldview to allow progress to be made towards the “yes” answer to the question raised above. I thus begin with a survey of some early and classic results which emerged within the tradition of classical “Montague Grammar”. This is then followed with some extensions of this program especially within the tradition of Categorical Grammar (broadly construed) and I conclude with some of my own research of relevance. I hope that a common theme emerges from this brief survey. This theme is that so often phenomena which appear at first glance to threaten the hypothesis of direct compositionality can be smoothly modeled in a way in which the semantics works with the syntax merely by a slightly more complex “packaging” of the semantic information than one might initially assume. Montague’s initial treatment of “generalized quantifiers” (discussed in Sec. 1.1. below) is, to my mind, a model of this kind of strategy - and a model of positing semantic packages in such a way as to respect the syntax of the language in question. Thus, while this is probably quite familiar to many readers, I begin with this case as I think it illustrates some very deep lessons, and sets the stage for results in other domains.

1. Two Classic Cases

1.1. Quantified (and ordinary) “noun phrases”

1.1.1. *Quantified expressions*

Consider classic cases of sentences with quantifiers, as in (1):

- (1) a. Every student left the party.
- b. Some dean fell asleep.

Students in a class on first order logic are always taught to translate these into “logical formulas” roughly along the lines of (2):

- (2) a. $\forall x[\text{student}(x) \rightarrow \text{left-party}(x)]$
- b. $\exists x[\text{dean}(x) \ \& \ \text{fell-asleep}(x)]$

When such an exercise is presented to students it is not generally intended as any claim about linguistic theory. That is, it is not usually intended as a claim to the effect that the actual grammar of English (a formal system which models speakers’ actual unconscious knowledge of the system) maps sentences to meaning by providing a procedure to map (1) into (2) and then providing a compositional model-theoretic interpretation for the

formulas in (2). Rather the goal of such an exercise in a logic class is to illustrate something about logic – not about English grammar. Still, let us for the sake of discussion construct a “Straw Man” who does make the claim that the mapping from (1) to (2) (with rules to provide an interpretation for (2)) is the actual grammar – for this will be a quite useful “straw” position to elucidate what is at stake.

Has our Straw Man provided a reasonable approximation of the natural language semantic compositions? If this were the actual way that English sentences are paired with meanings, it has some rather broadsweeping consequences. (a) First it requires an extra set of rules in the grammar; one needs to actually precisely formulate the rules that map the sentences in (1) into the Logical Forms in (2). The student in the logic class can do this because s/he already knows the meaning of the English sentence! But the grammar’s “job” is to compute this meaning – so the grammar must contain an explicit procedure for the mapping from English to (2). (b) Second, this view entails that *every student*, *some dean* etc. – while well-formed syntactic expressions – have no meaning. (Hence this provides a direct challenge to the hypothesis of local interpretation). Their contribution to meaning is only as a direction to trigger the rules that map into the relevant logical form. (c) Third, *every* and *some* don’t make the same contribution (the necessary connective is different). (d) Fourth, by way of background, let us make the (fairly standard) assumption that both proper names and ordinary definite NPs like *John Smith* or *the most disobedient dog* each have as their semantic value a function from worlds to individuals and that VPs like *fell asleep* have as their value a function from worlds to functions from individuals to truth values. Using the notation $[[\alpha]]$ to indicate the meaning of some linguistic expression α , the semantics combining ordinary NP subjects with VPs is the following: $\lambda w[[[VP]](w) ([[NP]](w))]$. (Hence the value of the sentence in any world is the value of the VP function in that world applied to the value of the NP in that world.) But the consequence of our Straw Man hypothesis is that – although *every student*, *some dean*, etc. – distribute syntactically just like ordinary NPs such as *John Smith* or *the most disobedient dog*, they play a completely different role in the semantic combinatorics. (As noted directly above, they actually have no meaning at all but simply are a direction for contributing a logical form.) I assume without argument that none of these four consequences are particularly happy ones. Especially disturbing is the consequences in (a); it obviously requires a more complex view of the organization of the grammar than would follow under direct compositionality.

The classic (and well-known) solution taken in “Montague Grammar” and much work in linguistic semantics ever since is extremely elegant. Assume that each such quantified NP does have a meaning – and its meaning is one which combines directly with the meaning of the VP in a smooth and predictable way. (For the moment I continue to refer to these as “NP”s for convenience; their syntactic category is discussed below.) This of course immediately raises the question of what type of object is the meaning of these NPs? Surely it can’t be (given a world) an individual (at least not on the ordinary understanding of “individuals”). Nor can it be a (world-dependent) set of individuals – although indeed this is the solution that students often find very appealing. Although one might plausibly maintain that *every dog* is simply the set of all dogs, it is easy enough to show that this does not generalize to the full set of cases. For the only reasonable

meaning for something like *no dog* would be the null set, but then *no dog* and *no cat* would have the same meaning, wreaking havoc with the compositional semantics: *No dog has fleas* and *No cat has fleas* would be synonymous. But all is solved by letting these have meanings which are (functions from worlds to) sets of sets of individuals. (I will for convenience hereafter suppress the world argument of each such function, and just talk about the extensions of each expression – that is their value in a given world.) Thus *[[every student]]* is the set of sets which have the *[[student]]*-set as a subset; *[[some dean]]* is the set of sets with a non-empty intersection with the *[[dean]]*-set; and *[[no dog]]* is the set of sets with a empty intersection with the *[[dog]]*-set. (Note that *[[no dog]]* and *[[no cat]]* are therefore distinct objects). The determiners themselves then (*no*, *every*, *some*) denote relations between sets of individuals. Or, recasting all of our sets as the characteristic function of that set, let $\langle a, b \rangle$ notate the set of functions from a to b ; let e be the set of individuals and t be the set of truth values. Then any VP (extensionally) has a meaning which is something in the set of $\langle e, t \rangle$ functions, and any quantified NP has a meaning which is in the set of $\langle \langle e, t \rangle, t \rangle$ functions. (Following standard terminology, I will refer to these as generalized quantifiers.)

In one fell swoop, the first three problems above have been solved. The semantic combinatorics of natural language is completely direct compositional and requires no mapping between the sentence and a different level which receives the compositional interpretation. At this point, let us (again for convenience) assume that the syntactic category of quantified NPs is distinct from that of ordinary (individual-denoting) NPs (we return to this directly) and let us call this category QNP. Then the syntax includes a rule to the effect that a well formed S can consist of a QNP followed by a VP, and the semantics is just (ignoring the world arguments) $[[S]] = [[QNP]]([VP])$. This just “flip-flops” the situation involving ordinary NPs; the subject is the function applied to the VP argument. In set (rather than function) terms, *Some dean fell asleep* is true if the *[[fell-asleep]]* set is in the *[[some-dean]]* set, which is the case if the *[[fell-asleep]]* set has a non-empty intersection with the *[[dean]]* set, and we get the correct truth conditions. No new rules are needed mapping these into more abstract representations. Second, each QNP is not only a well-formed syntactic expression but each has a meaning. The hypothesis of local interpretation is not threatened. Third, each quantificational determiner (*every*, *some*, etc.) of course has a different meaning, but each have the same type of meaning – all denote relations between sets.

1.1.2. Ordinary noun phrases and Type Lifting

This still leaves the problem in (d): the meaning of ordinary definite noun phrases is of a different type than that of quantified expressions even though the two have the same syntactic distribution. In the above discussion I gave the two kinds of expressions different category names (NP vs QNP), but that hides the fact that they do indeed have the same distribution; the difference is only in the semantics (NPs have meanings which are taken as argument of the meaning of the VP; QNPs are functions taking VPs as argument). Note that direct compositionality does not by itself require that all items with the same syntactic distribution have exactly the same type of meaning (and make the same contribution to the compositional semantics). For example many theories

call these two types of expressions both NPs (or, DPs) and allow a single syntactic category to have a variety of meanings, and to make different semantic contributions. But there is another reason to believe that definite NPs and proper names at last can have meanings which are of the same type as QNPs, and this is that they can happily conjoin as in (3) (by the terms “conjoin” and “conjunction” I include both *and* and *or*):

- (3) a. Porky and every cat crawled into the pen.
 b. Porky or some cat must have been munching at my hay.

As we will see below, we can give a simple and elegant semantics for *and* and *or* if we assume that two items which conjoin in the syntax always have the same semantic type. This then leads to the conclusion that perhaps a proper name like *Porky* does not (or not only) denote an individual.

Montague’s (1974) solution to this is again well-known. It relies on the observation that every individual can be “repackaged” as the set of sets containing that individual as member, and so definite NPs and proper names can indeed denote sets of sets. Thus let *p* be some individual; $[[Porky]]$ can be the set of sets containing that individual as member. (Since one of those sets is the singleton set $\{p\}$ we are guaranteed a unique such set of sets for each individual; and for every set of sets constructed in this way (i.e., the set of sets containing a given individual) we are guaranteed a mapping back to the individual). Again we can recast in function terms: $[[Porky]]$ is of type $\langle\langle e, t \rangle, t \rangle$ and for any function *f* of type $\langle e, t \rangle$, $[[Porky]](f) = f(p)$.

This solves the fourth problem. Proper names (and definite NPs) have the same type of meaning as do QNPs; they always combine in the syntax with the VPs in the same way, and they can conjoin since they are of the same type. (Hence they might as well be assigned the category QNP in the syntax, rather than NP. I revisit this directly.) But many have felt there is something odd about the hypothesis that - while there are individuals as part of the building blocks of meaning - no actual linguistic expression denotes an individual. Thus Partee and Rooth (1983) suggested a slight revision of Montague’s analysis designed to allow us to “have our cake and eat it too”: ordinary definite “NPs” and proper names do denote individuals but can also have the higher (Montague) type meaning. I will not give exactly their exact implementation of this solution but rather a slight variant.

Thus think of any linguistic expression as a triple of $\langle [\text{sound}], \text{syntactic category}, \text{meaning} \rangle$. A rule of the grammar takes one or more triple as input and returns a triple as output. Note that this builds in direct compositionality; the syntax (and phonology) specify how two expressions may combine and what the output category of the new expression will be, and the semantic part of these will specify the meaning of the whole from the meaning of the parts. For example, a phrase structure rule of the form $A \rightarrow B C$ is an abbreviation for the syntactic and phonological parts of such rules, and each would be coupled with a corresponding semantic rule giving the meaning of the expression of category *A* in terms of the meanings of the two input (i.e., daughter) expressions.

Although not crucial to the points here, it is easiest to recast this in the notation of Categorical Grammar (hereafter, CG), as this allows the statement of just a few very generalized rule schemata. Thus CG posits that the grammar has a small set of primitive syntactic categories, and a recursive definition of others; for any two categories A and B there is a category $A/_RB$ and a category $A/_LB$. (There are a number of versions of CG which differ both on details and on notation; I pick just one here.) From this we can have just two very general rule schemata: one combining $A/_RB$ with a B to its right to give a resulting expression of category A, and one the left inverse. Moreover, the system is set up in such a way that the meaning of any expression of category A/B is a function from B-type meanings to A-type meanings, and so the semantics is also predictable; the function is applied to the argument. We can thus recast this into the following general rule schemata:

- (4) Given a linguistic expression α of the form $\langle [\alpha] \text{ (some phonology)}; A/_RB; [[\alpha]] \rangle$, and a linguistic expression β of the form $\langle [\beta]; B \text{ } [[\beta]] \rangle$ there is an expression γ of the form $\langle [[\alpha] \beta]; A; [[\alpha]]([[\beta]]) \rangle$.

(There will be a corresponding rule for $A/_LB$.) “VP” then is really $S/_LNP$; and what we have been calling QNP can be unpacked as $S/_R(S/_LNP)$.)

We can now state a solution similar (although not identical) in spirit to the Partee and Rooth solution. Assume that in addition to the binary combinatory rules like (4) there are also unary rules: rules which take a single triple as input and give a new triple as output. Such rules can alter any one or more of the parts of the triple. The ones of concern here map an expression with one meaning into an expression with another meaning and also modify the syntactic category. Such rules have often gone under the rubric of “type shift” rules but note that here we are assuming a syntactic as well as a semantic shift. (Coupling the semantic shift with a corresponding syntactic shift is most natural given the other premises of CG.)

Thus let proper names be listed in the lexicon as of category NP, and have as their lexical meanings just individuals. (Presumably they are actually functions from world/time pairs to individuals; under a theory in which these are “rigid designators” they are constant functions. But I am systematically suppressing the world and time arguments.) Let definite NPs also denote individuals in their basic meaning. However, they can shift meaning and category without phonological consequence where they shift to $S/_R(S/_LNP)$ and to the set of sets containing that individual as member, as formulated in (5):

- (5) Given a linguistic expression α of the form $\langle [\alpha]; NP; [[\alpha]] \rangle$ there is an expression β of the form $\langle [\alpha]; S/_R(S/_LNP); \lambda P[P([[\alpha]])] \rangle$.

This shifts the basic lexical meaning of a proper name like *Wolfie* - an individual w - to the Montague meaning for *Wolfie* (the set of sets with w as member or, in function terms, the function mapping any function P of type $\langle e, t \rangle$ to the value that P assigns to w). This is often referred to as “Type Lifting”.

Interestingly, this can be stated far more generally, with a payoff to be discussed later. First we define a perfectly general lift operation. For any member a of a set A and any set of functions F in $A \times B$, we can “lift” a over F such that for any function f in F , $\text{lift}(a)(f) = f(a)$. Then we can recast (5) more generally:

- (6) Given a linguistic expression α of the form $\langle [\alpha]; A; [[\alpha]] \rangle$ there is an expression β of the form $\langle [\alpha]; B/_R(B/_L A) \text{ or } B/_L(B/_R A); \text{lift}([\alpha]) \rangle$.

(Note that if we see a syntactic category of the form B/A itself as naming a function from strings of category A to strings of category B , then the syntactic part of this operation can also simply be recast as the lift of the category of the input, eliminating the need to stipulate the directionality on the slashes, and removing the embarrassing result that the syntactic part of the lift “just happens to” preserve the word order one would get from the unlifted version.)

Notice that this solves the problems discussed above. Proper names and definite NPs do denote individuals, but they can shift to homophonous expressions with the higher type meaning. That they can occur anywhere a “QNP” (i.e., an $S/(S/NP)$) can follow from the fact that they can shift into this category. (The reverse does not yet follow; there could be expressions that ask for an NP but not a “QNP”. We return to this in Sec. 2.2.) It also follows that the two can conjoin. But the discomfort with the Montague solution – that there are no individual-denoting expressions – is eliminated; ordinary NPs can and do denote individuals as well as having the higher type meaning. Finally, a word is in order about the common slogan “Type lift only as a last resort”. In much current work in syntactic and semantic theory this has been used in a variety of odd ways, including as a slogan about how the grammar works. But the grammar has no aims and goals, and so it is a somewhat odd metaphor. Space precludes unpacking what is meant by this when taken as a principle of grammar, but I will use it as originally intended by Partee and Rooth: it is a claim about the human *processor*. The idea is that a human listener, encountering an ordinary NP like *Porky* or *the most disobedient puppy* will interpret this as denoting an individual. They will then recast this into the meaning of type $\langle \langle e, t \rangle, t \rangle$ only if either here is no way to semantically compose this with the rest of the sentence if it denotes an individual. In particular, then, if it is conjoined with a quantified NP (as in *Porky or every cat*) the processor will lift the meaning of *Porky* in order to arrive at a semantics for the expression above. More generally, the Partee and Rooth hypothesis to the effect that the processor will use the lowest types possible should be loosened somewhat: assume that higher types are also computed if an interpretation involving the lower types clashes with other facts about the discourse context and/or other pragmatic facts. Below we will see another nice consequence of this (looser) hypothesis regarding processing.

1.2. Coordination

Let us turn to a second domain; one which also played a role in the Partee and Rooth (1983) discussion and one whose analysis also nicely illustrates how the use of certain tools can facilitate analyses compatible with direct compositionality (and with the

concomitant hypothesis of local interpretation.) Consider an ordinary case of VP coordination, as in (7):

- (7) Porky grunted and snored.

Much of the work in early Transformational Grammar assumed that the way to “make sense” of such a sentence was to assume that hiddenly this was the coordination of two sentences. While I do not know this for sure, my hunch is that this conclusion was driven by a somewhat naive view of semantics; a student having taken nothing but a first order logic class learns the connectives $\wedge$ and $\vee$ as propositional connectives, and hence assumes that English must be such that *and* and *or* must be connecting proposition-denoting expressions.

It is easy enough, though, to see that this is naive. Since VPs (relative to worlds and times) are functions of type $\langle e, t \rangle$ they characterize sets, and in (7) can easily be recast as set intersection (or, in function terms *grunted and snored* is $\lambda x[[[\text{grunt}]](x) \wedge [[\text{snore}]](x)]$). Similarly, conjunction with *or* (as in *grunted or snored*) is simply set union. Since quantified expressions (*every pig*, *no dean*, etc.) denote (functions characterizing) sets of sets and since ordinary proper names (and definite NPs) can lift to this kind of meaning, we can easily generalize this to give a direct compositional semantics for complex expressions like the subjects in (8):

- (8) a. Every student and no dean fell asleep.
b. Porky or every cat must have been munching on my hay.

Again intersection or union is involved, and there is no need to “unpack” these into two sentences. *Every student* is the set of sets with the $[[\text{student}]]$ -set as subset; *no dean* is the set of sets whose intersection with the $[[\text{dean}]]$ -set is empty; and we then take the intersection of those two sets, which gives us all sets that contain every student and no dean. When this combines with the VP $[[\text{fell asleep}]]$ we arrive at the right truth conditions; the $[[\text{fell-asleep}]]$ set is in the set of sets each of which every student and no dean. We can also give a semantics for ordinary “NPs” conjoined with *and* and *or*:

- (9) Porky or Wolfie built that beautiful straw house.

(I use *or* rather than *and* here because there are additional complications regarding *and*; arguably it can also combine two individuals to give a plural individual and hence might not always involve intersection.) *Porky* lifts to the set of sets containing *p*, *Wolfie* lifts to the set of sets containing *w* *Porky or Wolfie* is the union of these; the set denoted by the VP is in the subject set which requires that either *p* or *w* be in the VP set.

So far so good. But is $[[\text{and}]]$ really always intersection and $[[\text{or}]]$ union? What about, for example, the ordinary case in which *and* does serve as a sentences connective as in *Porky grunts and Wolfie howls*? In teaching this material in introductory courses I always tell my students not to worry. Since the real meaning of a sentence is a function from worlds to truth values these too are sets: they are sets of worlds, and so here too intersection makes sense. But this is cheating (a fact which students rarely notice), for in

the cases above (VP conjunction and quantified NP conjunction) we ignored the world argument and now we are crucially using it. Moreover, we can also conjoin transitive verbs, as in (10), yet there is no reason to think of transitive verbs as sets; they are (again ignoring the world and time argument) presumably functions of type $\langle e, \langle e, t \rangle \rangle$ (they combine in the syntax first with the object and then with the subject, and hence we assume that in the semantics they are functions applying first to the individual denoted by the object to give a function of type $\langle e, t \rangle$ which takes the subject as argument):

(10) Porky cooked and ate the broccoli.

Partee and Rooth thus point out that *and* can be given a generalized definition for all of these cases; in fact they do this by giving a recursive definition using the propositional $\wedge$ as the base (and similarly with *or*). I will essentially be giving their definition, but I would like to again recast their discussion: I will show that their recursive definition is really just a generalization of an operation which has been quite useful in general in the direct compositional program. A full treatment would also extend to the syntax of *and*. From here onwards, however, I will focus primarily on the semantics so as to save space; the interested reader can generally construct the necessary syntactic details.

1.2.1. The “Geach” rule and generalizing this rule

Consider any function f of type $\langle a, b \rangle$. There is a natural mapping of this to a function of type $\langle \langle c, a \rangle, \langle c, b \rangle \rangle$ which – following the tradition in much of the Categorical Grammar literature – I will call the “Geach” operation and will notate as $\mathbf{g}$. Thus $\mathbf{g}(f)$ is $\lambda X[\lambda C[f(X(C))]]$ for X any function of type $\langle c, a \rangle$ and C a member of c . This is nothing more than a unary (“Curry’ed”) version of the function composition operator; $\mathbf{g}(f)(h) = f \circ h$.

This is a very useful tool for a direct compositionalist. For example, when we find an expression which – at first glance – appears to be something which semantically operates on propositions but appears in the syntax as, say, attached to a VP, we can assume that its meaning is merely the “Geach’ed” version of what we initially thought was the meaning. (Note that I am not suggesting in such a case that the grammar necessarily invokes a productive type shifting rule for the case at hand; it could simply be that the lexical meaning of the item in question is the “Geach’ed” version of the “naive” meaning.) More generally, expressions which appear in a smaller domain in the syntax but appear (at first glance) to have meanings over a larger domain can be treated this way. We do not, therefore, need to posit a level such as LF at which such expressions are actually syntactically “raised” out of the smaller expression and given scope over the larger one in order for the semantic composition to proceed; direct (local) compositionality can be maintained by merely assuming the meaning of the expression in question has a slightly fancier packaging than we initially thought.

Take English negation as a concrete example. We are (again perhaps from the first order logic lesson) used to thinking of negation as a propositional operator. In fact,

though, it does not occur this way in English. Actually, it doesn't even occur in the syntax as attached to a VP, but rather (as this very sentence demonstrates) as a clitic *n't* attached to an auxiliary. But for the sake of exposition, let us pretend that it occurs in the syntax just before a VP; our made up language has sentences like *Wolfie not howled*. It is easy to give a meaning for *not*; $[[not]]$ is simply the Geach'ed version of propositional $\sim$. Ignoring world and times arguments (which are easily folded in simply by additional applications of $\mathbf{g}$), its meaning is $\lambda P[\lambda x[\sim P(x)]]$; the reader can verify that this is $\mathbf{g}(\sim)$. The meaning of actual English *n't* can be given by further applications of $\mathbf{g}$. (Again I am not suggesting here that there is a productive rule in English grammar mapping $\sim$ into *not*; rather the lexical meaning of our (fictitious) word *not* is $\mathbf{g}(\sim)$.)

But we can also generalize the $\mathbf{g}$ operation. I will not define it in full generality here, but will give one additional version. Let f be a function of type $\langle a, \langle b, c \rangle \rangle$. Then there is a mapping which we can call $\mathbf{g}_1(f)$ into a function of type $\langle \langle d, a \rangle, \langle \langle d, b \rangle, \langle d, c \rangle \rangle \rangle$ such that $\mathbf{g}_1(f) = \lambda X[\lambda Y[\lambda D[f(X(D))(Y(D))]]]$. Recursively using this operation on propositional $\wedge$ is exactly Partee and Rooth's strategy for giving *and* in English a meaning which allows it to conjoin two expressions with meanings of any type, provided the two expressions have the same type meaning and the final resulting type for each such expression is of type t . In this case (unlike the example above with *not*) it makes sense to think of this as a fully productive "type shift" rule in English applying to a small class of items (perhaps just *and* and *or*) since these can conjoin expressions of any type (whose ultimate result is type t).

Before spelling this out, we need to make one more addendum to the discussion above; we have been acting as if *and* in the syntax combined with its two arguments simultaneously. But there is good reason to believe that it is "Curry'ed" and combines first in the syntax with the argument to its right and then with the argument to its left. (Space precludes giving the motivation for this here.) As usual ignoring world and time arguments (which can be folded in by applying $\mathbf{g}_1$ to the lexical meaning of *and* that I am giving directly), we can assume that $[[and]]$ is listed in the lexicon as being of type $\langle t, \langle t, t \rangle \rangle$ and is simply the Curry'ed version of $\wedge$ and thus its meaning is $\lambda p[\lambda q[p \wedge q]]$. Each additional meaning for *and* is derived by recursive application of $\mathbf{g}_1$; applying this to the basic meaning (and introducing a new individual argument slot) gives us the VP-conjunction *and* whose meaning is of type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, \langle e, t \rangle \rangle \rangle$ and is $\lambda P[\lambda Q[\lambda x[P(x) \wedge Q(x)]]]$. (For P and Q functions in $\langle e, t \rangle$ and x an individual.) Or this can apply to introduce $\langle e, t \rangle$ argument slot, which gives the *and* meaning appropriate to conjoining generalized quantifiers. The *and* meaning appropriate for transitive verbs is simply one more application of $\mathbf{g}_1$ to the VP *and* in such a way as to introduce one more individual argument slot (thus conjoining items of time $\langle e, \langle e, t \rangle \rangle$). The syntax to go along with this is predictable given CG assumptions.

2. Two Trickier Cases

2.1. Non Constituent Coordination

While the generalized definition of *and* (and *or*) allows us to account for a wide variety of cases involving conjunction under the hypothesis of direct compositionality, there are other cases which have gone under the rubric of “non-constituent coordination” which seem even more threatening to direct compositionality. An example of this is shown in (11) (this particular type of case is also known in the literature as Right Node Raising):

(11) Paul likes and Erik loves Georgian wine.

Incidentally, the common wisdom in why (11) necessitates another level of representation centers not only on the semantics but also on the syntax. But the interested reader can verify that with the CG tools discussed earlier, what I will say below about the semantics can be imported directly over into the syntax, allowing both *Paul likes* and *Erik loves* to be composed up as syntactic expressions without recourse to a “level” at which we have two hidden sentences here. We will focus here however just on the semantics. The conventional wisdom on a case like this is as follows. *likes* has a meaning which is such that it must combine with the denotation of its object argument first. Very often it is simply assumed that given a function of type $\langle a, b \rangle$, the only way for semantic composition to proceed is for that function to be “saturated” – i.e., to find its argument of type a . But there is, of course, no reason why this should be so. This view assumes (a) that functions themselves can never be arguments (since they would be unsaturated), and (b) that the only way two linguistic expressions can combine in the semantics is for one to denote a function which takes the meaning of the other as argument. Neither of these premises are necessary, and abandoning either one allows a perfectly simple way to give a semantic (and syntactic) composition for (11) under direct compositionality. The gist of this analysis below comes from a variety of works especially by Steedman (see, e.g., Steedman, 1987 and Dowty, 1988).

Thus recall that *Paul* in its lexical meaning denotes the individual p , but can type lift to $\lambda P[P(p)]$. If the operations available to the semantic composition include function composition, this can function compose with $[[likes]]$ and the result is $\lambda x[[likes]](x)(p)$ (i.e., the function characterizing the set of things that p likes). Similarly for *Eric loves*. $[[and]]$ can conjoin these two to give the (characteristic function of the) intersection of these sets, which applies to $[[Georgian wine]]$. Note, incidentally, that an alternative to the sketch here would be not to directly apply function composition, but rather to apply the “Geach” rule to lifted *Paul* – which gives $\lambda R[\lambda x[\lambda P[P(p)](R(x))]]$ (for R a function of type $\langle e, \langle e, t \rangle \rangle$) which is equivalent to $\lambda R[\lambda x[R(x)(p)]]$ which can then take $[[likes]]$ as argument. (Whether or not function composition should apply in one step or be broken down into the two steps of Geach + application depends, in part, on whether the intermediate stage – where the Geach rule has applied but application has not yet applied – is a useful object for other purposes; we will not pursue this here.)

Dowty (1988) – with the help of the generalized type lift rule given in (6) – develops a direct compositional analysis for even fancier cases such as (12):

(12) Duane ate the lobster yesterday and the leftover stew today.

(A caution: I am taking liberties with Dowty's actual discussion as he himself did not endorse the claim that there is a general type lifting rule in the grammar, but claimed that type lifting applies only in certain circumstances. The issues surrounding this will not be dealt with here.) Thus let *yesterday* and *today* be syntactically listed as VP modifiers; thus of type $\langle\langle e, t \rangle, \langle e, t \rangle\rangle$ and let *the lobster* and *the leftover stew* denote individuals. These cannot in and of themselves combine. But *the lobster* can lift its meaning over transitive verb meanings: thus it lifts to $\lambda R[R([\text{the-lobster}]]]$ (for R of type $\langle e, \langle e, t \rangle \rangle$). In other words, this new meaning is something which takes a transitive-verb meaning (a two place-relation R ; and returns the value of R applied to the-lobster individual - this is the set of individuals who stand in the R relation to "the-lobster". Now consider $[[\text{yesterday}]]$ of type $\langle\langle e, t \rangle, \langle e, t \rangle\rangle$. These two can function compose as shown in (13):

$$(13) \quad [[\text{yesterday}]] \circ \lambda R[R([\text{the-lobster}]]] = \lambda R[[[\text{yesterday}]](R([\text{the-lobster}]]))]$$

It will be helpful to rewrite the above in an expanded way (putting in the subject argument slot), as $\lambda R[\lambda x [[\text{yesterday}]](R([\text{the-lobster}]](x)))]$. The same will happen to *the leftover stew today*. These two somewhat strange expressions (functions from 2-place relations to VP meanings) can conjoin with generalized *and* and the result is something that wants a 2-place relation R and then an individual x to give the proposition that x R 'ed the lobster yesterday and x R 'ed the leftover stew today. This then combines with the transitive verb *ate* and the rest is unsurprising. But most research within linguistics assumes in the face of examples like (12) that direct compositionality is doomed, and assumes that the semantics of (12) can be put together only by unpacking this into two sentences (or, perhaps, two VPs).

Let us take stock. We have seen that the treatment of quantified NPs, the commonality of quantified NPs and "ordinary NPs", the treatment of coordination and the treatment of "trickier" cases of coordination all at first glance might look like challenges to direct compositionality. But for the most part these challenges arise only if insists that the interpretation of English is (roughly) like the interpretation of first order logic and if one insists that functions never serve as arguments. By "fancier packagings" of meanings – including meanings derived by type lifting and by the Geach rule – and by allowing functions to serve as arguments (and/or to make use of function composition) – we have no problem giving a smooth fit between the (surface, pronounced) syntax and the semantic composition for these cases. The remainder of the paper will examine a few additional cases illustrating these general types of tools and their consequences.

2.2. Quantified NPs in object position and scopes

Treating a QNP like *every student* as a generalized quantifiers (expressions whose meanings of type $\langle\langle e, t \rangle, t \rangle$ gave a very nice answer to how it is that these combine with verb phrases meanings under direct compositionality. But these do not occur only as subjects of verb phrases. They occur everywhere ordinary NPs do. For simplicity, we will restrict the discussion here to the case of QNPs in direct object

position as in (14), but the solution to be discussed here will generalize to all other NP positions:

(14) Carmen trained every husky.

The problem is that *train* denotes a two-place relation; we assume that this is “Curry’ed” and hence $[[read]]$ is a function of type $\langle e, \langle e, t \rangle \rangle$. How, then, the meaning of the object NP, which is of type $\langle \langle e, t \rangle, t \rangle$ enter into the semantic composition in such a way as to combine with the meaning of *train*?

There are a few different direct compositional solutions to this. But first we should briefly consider an obvious one which is not satisfactory. Given the remarks above about function composition, it might appear that the answer is simple. Let the meaning of (14) be put together in such a way that both in the syntax and in the semantics (lifted) $[[Carmen]]$ first combines with $[[train]]$ by function composition. This then can occur as argument of *every husky*. Of course the syntactic details also need to be worked out, but we will assume that that can straightforwardly be done.

Indeed in this case this is possible. But it will not give a full solution to the problems surrounding the existence of QNPs in object position. This is because there are actually two facts which must be accounted for: (i) that these are possible at all, and (ii) that when there is also a QNP in subject position, in general the sentence is ambiguous as to the scope of the two quantifiers. (In reality, some kinds of object QNPs resist the wide scope interpretation and so I am oversimplifying here.) Thus we need a solution to both problems, and ideally the mechanism to allow QNPs in object position in the first place should generalize to provide an account of scope ambiguities. But using function composition alone here solves only the first of these problems. Thus consider a sentence like (15):

(15) Some basketball player trained every husky.

$[[some\ basketball\ player]]$ can function compose with the ordinary transitive 2-place relation $[[train]]$ of type $\langle \langle e, t \rangle, t \rangle$, which result then occur as argument of $[[every\ husky]]$. But this procedure gives only the reading where *every husky* has widest scope, as the interested reader may verify. Not only does the simple addition of function composition to the set of operations here fail to give both readings, but this view actually gives as its only reading the reading which is less preferred. Something more is needed.

Before returning to (one of) the directly compositional solution(s), it is worth a brief digression to walk through the “standard” non-direct compositional solution. According to this view (put forth in, among others, McCawley, 1970, Lakoff, 1971, May, 1977, and Heim and Kratzer, 1998) the semantic combinatorics does not proceed in tandem with the syntax - rather what is interpreted is a distinct level of “Logical Form” (LF). Thus the “type mismatch” between the meaning of a QNP in object position and the meaning of an ordinary transitive verb is resolved by claiming that at the level of LF the QNP itself is not the object of the verb, and the relevant structure for (14) is roughly

as in (16). (The technical details vary considerably from proposal to proposal; I am modelling my discussion most closely on that in Heim and Kratzer, 1998).

Of course this says little without a procedure to interpret such structures. In fact, the compositional details are not straightforward. The usual view embeds this within a theory with variables and assignment functions. Think of the variables as simply the positive integers, and each assignment function g is a function from the variables to an individual (we consider here only variables over individuals; most theories also make use of higher typed variables). Moreover, an indexed “trace” (such as t_8 above) corresponds to a variable; thus $[[t_8]]$ on an assignment function g is $g(8)$. Technically, each expression has a meanings which is a function from the set of assignment functions to something else, but an expression with no unbound traces or pronouns (i.e., “variables”) within it is simply a constant function from the set of assignment functions. As the composition proceeds, the meaning of the lower S on any assignment function g will be the proposition that Carmen trained $g(8)$. The interpretation of the node which I have labelled Λ involves closing off the variable 8. The exact procedure for this is the semantics of “ λ -abstraction”, but rather than spell out the rule in generality we just illustrate its application here. It maps the meaning of S - which is a non-constant functions from assignment functions to propositions (and varies according to what individual the assignment function assigns to 8) to a closed function from assignment functions to functions of type $\langle e, t \rangle$. Since there are no other open variables within this expression, all assignment functions will be assigned the same value. Thus for any g , $[[\Lambda]](g)$ is the function characterizing the set that Carmen trained. Ignoring the g argument, The composition of the topmost S , then, is achieved by taking that function as the argument of $[[\text{every husky}]]$. (Note that $[[\text{every husky}]]$ is also not just a simple function of type $\langle \langle e, t \rangle, t \rangle$ but actually a function from assignment functions to such object. And $[[\Lambda]]$ is also a function from assignment functions to $\langle e, t \rangle$. But both are “closed” and hence constant functions, and so for expository simplicity we can think of the last step in the semantic composition as applying $[[\text{every husky}]]$ to the value of the note labelled Λ).

The advantage of this strategy is that it also automatically gives both scope possibilities for a sentence like (15). If only the object is pulled out and we replace the subject in the tree in (16) with *some basketball player*, then we get the reading with wide

scope object, as the interested reader can verify. But the subject QNP could also be “pulled out” above that, giving the wide scope subject reading. There is, incidentally, one unfortunate prediction here: the derivation with the minimal number of operations applying is the one that gives the non-preferred reading. (It is generally agreed that the wide scope object reading needs additional context and/or intonational cues to bring it out.) In light of the processing strategies discussed above one might not be happy with this. (To be sure, this has been addressed in a great deal of literature which suggests that processing strategies are also sensitive to the left-to-right order of the quantifiers). In any case, this view does have the happy consequence that the mechanism which allows QNPs in object position automatically generalizes to give the scope ambiguity. Of course all of this comes at what I would argue is a heavy cost: direct compositionality is abandoned and an additional set of rules are needed to give a correspondence between the surface syntax and the LFs. (Incidentally, Montague (1974) also had a version of this kind of solution, but his was actually compatible with a direct compositional architecture. Because it made use of a somewhat more complex set of assumptions about the syntax, we will not consider it here; see Jacobson (2002) for discussion of this with respect to the issue of direct compositionality.)

Are there ways that direct compositionality can also account for scope ambiguities as a consequences of the mechanism for allowing QNPs in object position? Indeed there are - in fact the literature offers us a choice, but I will focus on just one technique here. This solution is again inspired by Montague’s original answer to why it is that QNPs can occur in object position; Montague’s solution was again modified by Partee and Rooth (1983), and their modification was generalized by Hendriks (1993) in such a way that the ambiguity is a byproduct of the general procedure used to allow for QNPs in object position. Thus Montague’s technique for allowing QNPs in object position was to “up the lexical type” for transitive verbs. Rather than denoting two-place relations between individuals (i.e., functions of type $\langle e, \langle e, t \rangle \rangle$) he proposed that they actually “expect” generalized quantifiers in object position and that their meanings are functions of type $\langle \langle e, t \rangle, t \rangle, \langle e, t \rangle \rangle$. To get the intuition, let us use the notation train_* to signify the ordinary “train”-relation that holds between individuals. (In rendering Montague’s actual proposal I am taking some liberties in that I am oversimplifying an additional dimension; he also was concerned with verbs which crucially took intensional objects. I am ignoring intensions altogether here.) This, however, is not the meaning of the English verb *train*; that instead is (17):

$$(17) \quad \lambda P[\lambda y[\mathcal{P}(\lambda x [\text{train}_*(x)(y)])]]$$

Informally, this takes a generalized quantifier $\mathcal{P}$ in object position and then an individual y and returns true if the property of being ordinary-trained by y is mapped to true by $\mathcal{P}$.

This in and of itself is not enough. For while it does allow generalized quantifiers in object position, it still gives only one scope possibility for (15) (the one with subject wide scope). (Montague thus supplemented this with a “Quantifying-In” rule in the syntax; I will not consider this here since his full package of tools easily lends itself to the suspicion that it is overly loaded with machinery.) Moreover - as was the case for

Montague's hypothesis regarding the meaning of, e.g., proper names (according to which their lexical meaning was the generalized quantifier built from the individual) - one might object to the fact that no lexical item has the ordinary "train" relation (train_*) as its meaning. To remove this anomaly, Partee and Rooth proposed that *train* is listed in the lexicon as denoting the ordinary relation between individuals, but there is a type-shift rule mapping the verb *train* into one with the higher type Montague meaning. What is particularly interesting is that Hendriks (1993) subsequently proposed a generalization of the Partee and Rooth strategy. Let any function f which takes any number of arguments followed by an individual argument followed by any other number of other arguments to ultimately yield a truth value map to a function f' which is such that the individual argument slot is instead a generalized quantifier argument slot. I think that the intuition is best given in prose rather than in notation, so I spell it out (slightly informally) in (18), and (following Hendriks) refer to the general process as Argument Lifting:

- (18) Consider any expression α whose meaning $[[\alpha]]$ is a function which first takes n arguments (of any type), then takes an e -type (individual) argument and then takes m arguments (of any type) and finally returns a truth value. This maps into a new expression β with the same phonology, and where $[[\beta]]$ takes the same sequence of arguments, then an $\langle\langle e, t \rangle, t \rangle$ argument $\mathcal{P}$ and then the remaining sequence of arguments and returns true iff $\mathcal{P}$ contains as one of its sets: the set of individuals x such that $[[\alpha]]$ applied to the first set of arguments, then to x and then to the remaining set of arguments yields true.

(One interesting sidelight: given a simple function of type $\langle e, t \rangle$, normal Type Lifting of that function and Argument Lifting of the e -argument slot turn out to be equivalent.)

The happy result here is that this automatically gives scope ambiguities between any two (or more) co-arguments of the same verb. Consider again (15):

- (15) Some basketball player trained every husky.

If the object slot is first argument-lifted give the Montague meaning, *train every husky* can be composed up and then taken as argument of the subject; this gives wide scope subject. (Another derivation would be one in which the lexical meaning *train* first maps to the variant with argument-lifting on the object slot and then to the variant with argument-lifting on the subject slot; the fancy new verb then combines with the object and then with the subject. This is equivalent to the derivation in which only the object argument-slot lifts and where the VP-meaning is the argument *some basketball player*.) But a third possible derivation is one in which the subject slot is argument-lifted first and then the object slot is argument-lifted; this gives the wide scope interpretation to the subject. Notice that the minimal derivation is the first one spelled out above - consistent with the Partee and Rooth hypothesis that lifting (including the lifting of an argument slot) comes with a processing cost and that the fewer lifting operations apply the more accessible the reading.

There is an additional textbook argument against DC drawn from this general domain, and this concerns the interpretation of sentences like (19) (the phenomenon here is known in the literature as Antecedent Contained Deletion (see Bouton, 1970)):

(19) Carmen will read every book that Sally will.

The natural understanding of this sentence can be paraphrased as *Carmen will read every book that Sally will read*. For the sake of discussion, assume that the interpretation of (19) involves either the grammar supplying or the listener picking up the meaning of some other linguistic expression. A textbook argument for the LF approach (due originally to Sag, 1976) goes (very roughly) as follows. (a) *[[will]]* is a VP modifier, so it had better find a meaning of type $\langle e, t \rangle$ to combine with. (b) There must be a linguistically overt VP to supply this meaning. (c) It cannot be the meaning of the matrix VP *[[read every book that Sally will]]* because that would lead to an infinite regress. (d) Enter LF: if *every book that Sally will* is actually “raised” out of the VP in the manner shown in (16), we have the expression *read t_g* whose meaning on any g is the property of reading $g(8)$. This can thus supply the “missing meaning”. But of course we have already seen there is no reason to believe the assumption in (a): *[[will]]* does not need to find an $\langle e, t \rangle$ type meaning. It could instead find an $\langle e, \langle e, t \rangle \rangle$ meaning (a 2-place relation) and function compose with this. (Or, it could undergo **g** such that it is now actually looking for a 2-place relation). All it needs, then, is to combine with the basic lexical meaning of *[[read]]*. (In the Hendriks view, *read* in (19) would have undergone Argument Lift to accomodate the QNP in object position. Nonetheless there is no reason to believe that its lexical meaning would also not be available). This answer to the “textbook” argument against DC is discussed in Cormack (1984), Evans (1988) and in greater detail in Jacobson (1992, 2008).

3. More on coordination: Interaction with “binding”

I close this discussion with some of my own work on the so-called binding of pronouns and the relationship between pronominal binding and issues surrounding direct compositionality. A good illustration of the issues and results once again centers on coordination. Thus we saw in Sec. 2.1. that simple cases of so-called “non-constituent coordination” (or, Right Node Raising) do not threaten direct compositionality (I say “so-called because in the CG treatments referenced above these boil down to constituent coordination). But there are more complex and interesting cases where Right Node Raising interacts with pronominal binding, as in (2). I give (20a) because it is more natural than (20b), but it also involves a number of irrelevant complications, so in fact we will focus the discussion on (20b) as a stripped down version of (20a):

- (20) a. Every man_i loves but no man_j wants to marry his _{i/j} mother.
 b. Every man_i loves and no man_j marries his _{i/j} mother.

The indices here are not meant as part of the grammatical apparatus, but are just being used to illustrate the relevant reading.

Conventional wisdom within much of linguistic theory would tell us that the analysis given earlier for RNR simply cannot extend to this case; rather this case requires a level of representation in which there are actually two sentences, and at which, therefore (20b) is really represented as (21)

(21) Every man_i loves his_i mother and no man_j marries his_j mother.

This follows - given the standard assumptions regarding the interpretation of pronouns - for two reasons. First, a pronoun corresponds to a variable, and it makes no sense to have a single variable bound by two different things. Second, the apparatus to give an actual semantics for “binding” under the usual assumptions takes place in a large domain. In (21), the “binding effect” is usually accomplished by a rule acting on a representation of the form *t_i loves his_i mother* although there are variants that operate on just the VP *loves his mother*. Still, these accounts need at least this big of a chunk of representation to get the semantics to work.

To flesh these remarks out in a bit more detail, in the standard account all pronouns (like traces) come with an index. (Note that this means that there are actually an infinite number of pronouns in the lexicon.) The use of these is exactly like we saw above for traces; $[[he_7]]$ on an assignment function *g* is *g*(7). The interpretation of something like the first conjunct in (21) involves a Logical Form such as (22):

(22) [every man] [_Λ 7 [_S t₇ loves his₇ mother]]

For any *g*, the value of the lowest *S* is the proposition that *g*(7) marries the mother of *g*(7). This is then mapped into the closed property - the function that (on each assignment function) maps an individual to true if they are a self’s mother lover. This then argument of $[[every\ man]]$. The same sort of procedure would happen separately for the second conjunct in (21). There are two key points about this analysis. First, we need two separate occurrences of *his* (in 20) to get the binding to work out properly, and second, the domain at which binding takes place in (22) is not local - it comes about because of the interpretation of the node that I have labelled Λ . The consequences of this for (20) are again that we need the full two sentences to have a “big enough” domain for binding to take place. (As noted above, there are theories in which the relevant shift happens at the VP level. But the consequences for the analysis of (20b) are the same - we need to be computing the meaning of expressions like *loves his mother* and *marries his mother* and so there would appear to be no way to directly interpret (20b) without recasting this as two sentences as in (21).)

But there is an alternative approach under a theory that makes no use of variables (and assignment functions) which allows all of the meanings to be assigned purely locally and which therefore allows for a direct compositional treatment of (20) (and of many related constructions). The approach I will discuss here is one developed in a series of papers of mine (see in particular Jacobson, 1999 for the case at hand). This account makes no use of indices in the syntax (there is, then, modulo case, only one 3 person singular masculine pronoun in the language) and no use of assignment functions. The

meaning of any expression with a pronoun unbound within it is not a function from assignment functions to something else, but rather a function from individuals to whatever type of meaning the corresponding expression with no unbound pronoun would have. In other words, in an expression like *Every man thinks that he lost*, the meaning of *he lost* is a function from individuals to the proposition that that individual lost. (It thus means the same as *lost* except that gender information is included, so that it is a partial function, but we will ignore gender here). This means that the meaning of a pronoun itself is also a function from individuals to individuals, and we will assume that it is the identity function.

How do we get *he lost* to have the above meaning? After all, $[[\text{lost}]]$ is a function of type $\langle e, t \rangle$ while $[[\text{he}]]$ is of type $\langle e, e \rangle$. Once again, we can make use of the “Geach” rule, so that $[[\text{lost}]]$ of type $\langle e, t \rangle$ maps to a function of type $\langle \langle e, e \rangle, \langle e, t \rangle \rangle$, where $[[\text{lost}]]$ is $\lambda f[\lambda x[[\text{lost}]](f(x))]$ (for f a function of type $\langle e, e \rangle$). When this combines with $[[\text{he}]]$ (the identity function) we get back $[[\text{lost}]]$. By the same token, an expression like *his mother* will denote a function from individuals to individuals; more specifically its meaning is $\lambda x[\text{the-mother-of}(x)]$ which I will refer to as “the-mother-of” function. Exactly how *his mother* gets this meaning would require a digression on the semantics of genitives; see Jacobson (1999) for details. It should also be noted that the entire system is coupled with a syntax and this is quite crucial to get the “right semantic” combinatorics to be at work with the syntax. But the coupling with a CG syntax is very natural.

The use of **g** allows pronouns to appear in slots characteristically reserved for individuals, but we have yet to introduce apparatus to get the effect of “binding”. Here the crucial idea is that this is a meaning shift rule but one that can be stated very locally. Note that the standard theory uses a meanings shift as well; the interpretation of the node labelled Λ in (22) involves a shift from the meaning of S - a non-constant function from assignment functions to propositions - to a constant function to functions of type $\langle e, t \rangle$. In the variable-free program under consideration here, the relevant shift applies to the meaning of expressions of type $\langle a, \langle e, t \rangle \rangle$. Thus for any function f of type $\langle a, \langle e, t \rangle \rangle$ (for any type) there is a mapping into a function which I will call $\mathbf{z}(f)$ as follows:

- (23) Given an function f of type $\langle a, \langle e, t \rangle \rangle$ (for any type), $\mathbf{z}(f)$ is a function of type $\langle \langle e, a \rangle, \langle e, t \rangle \rangle$ such that $\mathbf{z}(f) = \lambda B[\lambda x[f(B(x))(x)]]$ (for B a function of type $\langle e, a \rangle$).

Assume that any expression α is of type $\langle a, \langle e, t \rangle \rangle$ can map without phonological consequences to an expression β whose meaning is $\mathbf{z}(\alpha)$. (Once again embedding this all within a CG syntax also effects a syntactic change on the category of the expression. Moreover, the entire system is spelled out in much fuller generality in Jacobson (1999).)

It is easiest to illustrate the effects of $\mathbf{z}$ by example. Let *love* denote the ordinary 2-place relation between individuals. Then $\mathbf{z}([[\text{love}]])$ is of type $\langle \langle e, e \rangle, \langle e, t \rangle \rangle$; it is the relation that holds between individuals and functions of type $\langle e, e \rangle$ such that to $\mathbf{z}\text{-love}$ a function f (of type $\langle e, e \rangle$) is to be an x who stand in the ordinary love-relation to

$f(x)$. Now consider the combinatorics for an ordinary sentence like *Every man loves his mother* (on the bound reading). *loves* undergoes **z** and then takes $[[\text{his mother}]]$ as argument. This denotes the-mother-of function; so *loves his mother* maps an x to true if x **z**-loves the-mother-of function, which is to say that *loves his mother* maps x to true just in case x loves the mother of x . Note that this is exactly the meaning (with the assignment functions stripped away) that the “standard” view gives to the expression labelled Λ in the LF in (22), but we have accomplished this in a purely local and direct way, making no use of LFs and no use of assignment functions. (There is, incidentally, another reading of this sentence where *his* remains free. This is gotten simply by the application of **g** on both *loves* and on the subject rather than **z**.)

We now have no difficulty directly composing the meaning of (20b). Here, $[[\text{every-man}]]$ simply function composes with **z** ($[[\text{love}]]$) in the first conjunct. Rather than this yielding a function of type $\langle e, t \rangle$ it yields a function of type $\langle \langle e, e \rangle, t \rangle$; the function that characterizes the set of $\langle e, e \rangle$ functions that every man **z** loves. (Spelling this out in more detail, we get the set of functions F such that for all f in F , the $[[\text{man}]]$ set is a subset of the set of individuals x such that x ordinary-loves $f(x)$.) Similarly with *no man marries*; function composition of $[[\text{no-man}]]$ with **z** ($[[\text{marries}]]$) yields the set of functions f such that no man **z**-marries f . Generalized *and* then intersects these two sets. There is, moreover, no problem in getting the effect of “binding” of the pronoun *his* in the “right node raised” constituent *his mother*. This automatically denotes the-mother-of function, and the entire sentence says that this function is in the intersection of the set of functions that every-man **z**-loves and the set that no man **z** marries.

There is thus no need to posit a level of representation for (20b) at which the sentence is “unpacked” into two sentences for the purposes of the semantics. The interpretation is local throughout, and direct compositionality is respected. In fact, there are many other related constructions which show the same moral. To mention just one here, consider the analysis of what are known as functional questions, as in (24):

(24) Q. Who does no man invite to his wedding? A. His ex-wife.

Groenendijk and Stokhof (1983) and Engdahl (1986) showed that the question part here can be given a sensible semantics if this is taken to be a question about functions of type $\langle e, e \rangle$; it can be paraphrased as in (25):

(25) What is the function f such that no man invites $f(x)$ to his wedding?

But in order to allow (24) to have this reading, both Groenendijk and Stokhof and Engdahl needed to posit extra apparatus in the theory (basically, allowing traces to have a complex index and to correspond to variables over functions applied to variables over individuals). Put differently, in the standard view, the existence of functional questions is rather surprising. But in the variable-free program, nothing extra is needed to get the functional reading: indeed we would be quite surprised had it not existed. For this simply involves **z**(invite) rather than ordinary-invite. The fact that *his ex-wife* is a good answer to a functional question is also automatic - since it necessarily denotes a function of type

<e,e>. (Note that the “binding” of *his* in the standard view again requires that the answer has a more complex representation which includes enough material to effect the binding. But this is not necessary; a smooth and local interpretation of all of the pieces is possible using the variable-free apparatus.)

4. Conclusion

The moral that I have attempted to draw from all of the above examples is that we should not be too hasty in abandoning the hypothesis of direct compositionality. Indeed, over and over we find that apparent challenges to this simply require a more complex “packaging” of meanings (often accomplished by means of rules shifting the meanings of linguistic expressions) than we might have thought at first glance. Of course the ultimate viability of the direct compositional program can only be judged in terms of the overall simplicity of the requisite operations (should they become too baroque one might be suspicious) as well as how it hooks into a full theory of the syntax here. There most certainly remain any number of domains which as of yet have not been given a satisfactory direct compositional analysis. Still, it seems that the strategy being endorsed here -which arguably begins with Montague’s treatment of generalized quantifiers - is one well worth pursuing.

Acknowledgments

* This paper is a version of the tutorial on language given at the 8th Tbilisi Symposium on Language, Logic, and Computation, Bakuriani, Georgia, 2009. I would like to thank both the audience there and the organizers for both helpful discussion and for a very wonderful conference in a beautiful setting. This work was partially supported by NSF Grant BCS 0646081.

References

- Bouton, L.: 1970, “Antecedent contained pro-forms”, in *Proceedings of the Chicago Linguistic Society* 6. Chicago, Ill.: Chicago Linguistic Society.
- Cormack, A.: 1984. “VP anaphora, variables, and scope”, in F. Landman and F. Veltman (eds.), *Varieties of Formal Semantics*. Dordrecht: Foris.
- Dowty, D.: 1988, “Type raising, functional composition, and non-constituent conjunction”, in R. Oehrle, E. Bach, and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*. Dordrecht: Reidel.
- Engdahl, E.: 1986, *Constituent Questions*. Dordrecht: D.Reidel Co.
- Evans, F.: 1988, “Binding into anaphoric verb phrases”, in *Proceedings of the Fifth Annual Eastern States Conference on Linguistics*. Columbus, Ohio: Ohio State University Linguistics Department.

- Groenendijk, J. and M. Stokhof: 1983, "Interrogative quantifiers and skolem functions", in K. Ehlich and H. van Reimsdijk (eds.), *Connectedness in grammar, discourse, and text*. Tilburg: Tilburg University.
- Heim, I. and a. Krater: 1998, *Semantic Intepretation in Generative Grammar*. Malden, MA: Blackwell.
- Hendriks, H: 1993, *Studied Flexibility*. Ph.D. Dissertation, University of Amsterdam, Amsterdam: ILLC Dissertation Series.
- Jacobson, P.: 1992, "Antecedent Contained Deletion in a Variable-Free Semantics," in *Proceedings of Semantics and Linguistic Theory 2*. Columbus, Ohio: Ohio State University.
- Jacobson, P.: 2002, "The (dis)Organization of the Grammar: 25 Years, in *Linguistics and Philosophy* 25.5-6. 601-626.
- Jacobson, P.: 2008, "Direct compositionality and variable-free semantics: The case of Antecedent Contained Deletion", in K. Johnson (ed.), *Topics in Ellipsis*. Cambridge: Cambridge University Press.
- Lakoff, G.: 1971, "On Generative Semantics", in D. Steinberg and L. Jakobovits (eds.), *Semantics*. Cambridge: Cambridge University Press.
- May, R.: 1977, *The rammar of Quantification*. Ph.D. Dissertation, MIT, Cambridge, MA.
- McCawley, J: 1970, "Where do Noun Phrases Come from?", in R. Jacobs and P. Rosenbaum (eds.), *Readings in English Transformational Grammar*. Waltham, MA: Ginn & Co.
- Montague, R.: 1974, "The Proper Treatment of quantification in Ordinary English", in R. Thomason (ed.), *Formal Philosophy: Selected Papers of Richard Montague*. New Haven: Yale University Press.
- Partee, B. and M. Rooth: 1983, "Generalized Connunciton and Type Ambituity", in R. Bauerle et al. (eds.), *Meaning, Use, and the Interpretation of Language*. Berlin: de Gruyter.
- Sag, I.: 1976, *Deletion and Logical Form*. Ph.D. Dissertation, MIT, Cambridge, MA.
- Steedman, M.: 1987, "Combinatory grammars and parasitic gaps", in *Natural Language and Linguistic Theory* 5, 403-439.