

The Locality of Interpretation: The Case of Binding and Coordination

Pauline Jacobson
Brown University

1. Three Related Claims

This paper will first elucidate the relationship between three claims concerning the syntax/semantics interface, and - more importantly - will then provide evidence for all three. The first claim is that surface structures directly receive a model-theoretic interpretation, without the use of mediating levels of representation such as deep structure and/or LF. The second claim is a corollary of the first: I will refer to it as the *hypothesis of local interpretation*. This hypothesis says that each surface syntactic expression does in fact have a meaning. Note that the contrasting position here would claim that some surface constituents have no meaning in and of themselves, and hence surface structures must be mapped into another level of representation in order to be assigned meanings.

Before moving on to the third claim, it will be instructive to elucidate the hypothesis of local interpretation by example. Consider the case of Right Node Raising (RNR), as illustrated in (1):

- (1) Mary loves and John hates model-theoretic semantics.

The traditional view of this is that the semantics cannot directly assign a meaning to (1) because the expression *Mary loves* does not have any meaning, nor does *John hates*, nor does *Mary loves and John hates* (this latter assumption follows under the traditional view that the meaning of *and* connects only propositions). Thus, in both traditional transformational grammar work and within some more recent work, it has been assumed that (1) must be derived from (via a transformation) or mapped into (via reconstruction) another level of representation such as (2), where the semantics actually interprets (2):

- (2) Mary loves model-theoretic semantics and John hates model-theoretic semantics.

However, it has been known for quite some time that (1) is indeed compatible with the hypothesis of local interpretation. One possible analysis, for example, combines the Dowty (1987) / Steedman (1990) analysis of RNR which is framed within a Categorical Grammar syntax with the analysis of the semantics of generalized conjunction developed in Gazdar (1980), Partee and Rooth (1983) and Keenan and Faltz (1985). Thus we can assume (following the basic line in Partee and Rooth (1983)) that proper names are listed in the lexicon as denoting individuals but can typelift to generalize quantifiers; I will assume that this semantic lift is accompanied by a corresponding syntactic lift. Further, here and throughout this paper I will in general be assuming a categorial grammar syntax. (This is probably not crucial; I suspect that the semantic analysis throughout this paper could be implemented in other syntactic theories as well. However, I also suspect that such implementations would be more complex than that developed here, since Categorical Grammar seems to provide just the right fit between syntax and semantics so as to allow the syntactic and semantic composition to proceed in a smooth way.) The analysis of (1) thus proceeds as shown in (3) (note that the

assignment functions do no work, and we could just as easily strip them away, doing without the assignment functions and therefore also the variables as model-theoretic objects or as any part of the semantic machinery. This is the essential idea of a Variable-Free semantics: it makes no use of variables (nor assignment functions) as any part of the semantic machinery.

This way of thinking about meanings is inspired by the work on Combinatory Logic of Curry and Feys (1958), and it was also explicitly put forth for natural language in Quine (1966). More recently it has been explored in a good deal of the Categorical Grammar literature, see especially Szabolcsi (1987, 1992), Hepple (1990), Dowty (1992), and Jacobson (1991, 1992a, 1992b, 1994a, 1994b). These all differ somewhat in their precise implementation; the implementation to be considered here is the one developed in my papers cited above, especially in Jacobson (1994a).

Thus the third claim of concern in this paper is that the semantics makes no use of variables nor assignment functions. (Note, though, that I will often use variables to represent meanings but this is for representational purposes only. To verify that these are not a part of the actual semantic machinery, one can note that all of the variables in my representations are bound. Another way to convince oneself that the use of variables is for notational convenience only is to note that all of the formulas with variables could be reformulated without by using combinators. Unfortunately, such reformulations would often be quite reader-unfriendly; hence the use of representations with variables.)

Let me now elucidate the relationship between the three claims. First, as noted above, the claim that there is no mediating level such as LF entails the hypothesis of local interpretation. The reverse is not the case, but if it is correct that each surface expression is assigned a meaning then it is obvious that there would be little reason for positing a mediating level like LF. Clearly, then, if local interpretation is correct, then the world is simplest if the grammar directly assigns a meaning to each surface expression. As to the relationship between local interpretation and a variable-free semantics: I will show that while the hypothesis of local interpretation does not absolutely require a variable-free semantics, the semantics is indeed simplest if there are no variables. The reason has to do with the location of binding effects and with the location of the semantic effect of pronouns. I will argue that in the variable-free approach to semantics (at least in the particular approach here) the location of binding and the location of the effect of the pronoun are much more local than in the standard semantics with variables. I will further argue that locating the effect of binding and the effect of the pronoun in this very local way is just what is needed in order to maintain the hypothesis of local interpretation. On the other hand, it will be shown in Sec. 9 that a semantics with variables could mimic the effects here and also locate these effects quite locally, but only by undermining much of the motivation for variables in the first place and by having the variables do no real work. Finally, I will consider in detail the interaction of binding and conjunction, and will show that this provides some rather striking evidence for the package which maintains all three claims above.

2. Variable-Free Semantics

2.1. *Standard account:*

2.1.1. *Pronouns in a standard semantics with variables*

Before turning to the variable-free account to be assumed here, it will be useful to review the standard account of pronouns both to make certain assumptions explicit

and to serve as a point of comparison to the variable-free account. Thus consider (5) and (6):

- (5) Every man_i thinks that he_j lost.
 (6) Every man_i loves his_j mother.

The standard view of (5) assumes that *he* translates as just the variable x , the embedded S *he lost* translates as $\text{lost}'(x)$, and the matrix VP translates as $\text{think}'(\text{lost}'(x))$. Similarly in (6) *his mother* translates (roughly) as $\text{the-mother-of}(x)$ and the VP as $\text{loves}'(\text{the-mother-of}(x))$. The use of the term "translates as", however, occasionally masks the more important question: what do each of these expressions actually mean? As noted earlier, the usual answer to this is that each such expression corresponds to a function from assignment functions to something else (and since each of these contains an unbound variable, these will be non-constant functions in each case). In other words, *he* in (5) has as its meaning a (non-constant) function from assignment functions to individuals; *he lost* denotes a (non-constant) function from assignment functions to propositions, while *his mother* denotes a function from assignment functions to individuals.

Now compare (5) to a corresponding sentence with no pronoun:

- (7) Every man thinks that Bill lost.

One of the central features of the standard account is that the embedded sentence in (7) has exactly the same type of meaning as the embedded sentence in (5). Both denote functions from assignment functions to propositions, but in (7) this simply happens to be a constant function. This is one of the great appeals of the standard theory: since the embedded Ss in (5) and in (7) have (roughly) the same syntactic distribution, the great "coup" of the standard account is that they can also be given the same semantic type.

Since this is so fundamental to the appeal of the standard account, let us explore this in a bit more detail. There is a key generalization and two further assumptions which underlie the standard view. The generalization is as follows:

- (8) Consider any expression C which contains no pronouns which are unbound within C, but which does contain an NP (or a pronoun bound within C). Consider further an expression C' which is exactly like C except that C' contains an unbound pronoun in the position of the NP (or the pronoun bound within C). Then if C can grammatically occur in some environment so can C'. (The reverse is not the case in view of the fact that there are certain environments which demand constituents with resumptive pronouns.)

Put informally: wherever we find constituents with no unbound pronouns, we can find corresponding "open" expressions. The two further assumptions which implicitly underlie the standard account are: (a) Given (8), C and C' must be of the same syntactic category; and (b) they must therefore also be of the same semantic type. As we will see below, however, the generalization in (8) does not in fact warrant these two conclusions.

2.1.2. Binding in the standard Semantics

There is more than one way to accomplish binding under the standard view, but it is worth noting that these all require some kind of type-shift rule which shifts the meaning of an expression containing a pronoun into a new meaning which is such that the pronoun will be bound. To illustrate we will, for convenience, consider just

the case of binding by a subject. One way to effect this which does not require pulling the binder out of subject position is based on the "Derived VP Rule" (Sag, 1976; Williams, 1977; Partee and Bach, 1981; etc.). Under this view, the meaning of a VP can undergo the type-shift rule in (9) which we give here in its most general form:

$$(9) \quad VP' \rightarrow \lambda v[VP'(v)] \text{ (for } v \text{ a variable over variables of type } e)$$

(Note that (9) uses no indexing conventions and hence has no way to ensure that the variable being λ -abstracted over here does indeed correspond to an "open" variable in the translation of the input VP. One can either elaborate (9) so as to ensure this, or one can simply allow vacuous λ -abstraction and thus keep (9) in its maximally simple form.)

The idea is that the meaning of the matrix VP in (5), for example, is the open property think'(lost'(x)) - this is a (non-constant) function from assignment functions to properties. This can undergo (9) to shift into the constant function from assignment functions to properties $\lambda x[\text{think}'(\text{lost}'(x))(x)]$ and this then occurs as argument of the subject NP every-man'. (Actually, to say that the meaning of the VP occurs as argument of the subject is an oversimplification, since both the VP and the subject are functions from assignment functions to something else and thus the meaning of the VP is not the correct type to be the argument of the subject. Rather, using g as a variable over assignment functions, the actual semantic composition yields the function: $\lambda g[\text{every-man}'(g)(VP'(g))]$. In fact, in all cases of "functional application" there is actually this additional layer of applying the meaning of each expression to an assignment function and so, technically speaking, all of the meanings represented above under the standard theory contain this "level" of the assignment functions. For convenience, however, I will generally suppress this, as is quite standard.) Similarly, in (6), loves his mother denotes the open property loves'(the-mother-of'(x)) and shifts by (9) to denote the closed property $\lambda x[\text{love}'(\text{the-mother-of}'(x))(x)]$ which occurs as argument of the subject. Incidentally, while this rule is formulated only to cover the case of binding by a subject NP, it can easily be generalized to cover binding by other arguments.

The second well-known approach to binding assumes a theory with Quantifying-In, Quantifier Lowering, and/or Quantifier Raising. In other words, it assumes an account in which there is a level of representation at which there is only a variable (or indexed pronoun) in the position which is occupied by the binder in the surface syntactic structure. To be neutral between Quantifying In, Quantifier Lowering and Quantifier Raising, I will call any such approach a *Binders-Out* account. Here, then, the relevant rule can be formulated in its most general form as in (10) (the same remarks apply here regarding vacuous λ -abstraction):

$$(10) \quad S' \rightarrow \lambda v[S']$$

Thus here it is not the VP which undergoes the type-shift rule, but rather a "binder-out" version of the meaning of the entire S. In (5), for example, we have a mapping by (10) from the open proposition thinks'(lost'(x))(x) to the closed property $\lambda x[\text{thinks}'(\text{lost}'(x))(x)]$. Once again, this will occur as argument of the "pulled out" generalized quantifier (in this case, every-man').

Under either approach, then, binding is the effect of a type-shift rule which shifts the meaning of a constituent containing a pronoun, and the point at which the

meaning shift occurs is **just before the expression in question combines with the meaning of the binder**. In (9) the shift applies at the level of the meaning of the VP, while in (10) it applies at the level of the meaning of the entire S. The fact that this is the point at which the meaning shift occurs is not surprising once one considers the very rationale for the standard treatment of pronouns/variables: the idea of this treatment is to allow a pronoun (which is unbound within some constituent C) to "pretend" that it is just like an ordinary NP (or bound pronoun). Hence, the semantic effect of the pronoun, then, is not felt until just before the expression combines with the meaning of the binder. This observation will be crucial below.

2.2. Variable-Free Account

2.2.1. Account of the meaning of expressions with unbound pronouns

The basic premise of the variable-free account which we consider here is that there are no assignment functions and no variables as model-theoretic objects or as crucial parts of the semantic machinery in any sense. This then raises the question of what is indeed the meaning of an expression containing a pronoun (or, which is a pronoun) which is not bound within that expression. The answer here is that such an expression denotes a function from individuals to whatever type of meaning a similar expression without a pronoun would have. (For simplicity, we consider here only the case of expressions containing one pronoun. The whole system easily generalizes to cases with more than one pronoun; see Jacobson (1992a; in preparation) for discussion.) The same remarks hold for pronouns themselves: we thus take a pronoun to denote a function from individuals to individuals. (Incidentally, if there are proforms of other categories besides NP proforms, then an expression containing a proform of category B will not denote a function from individuals, but rather a function from whatever is the semantic type of B.) Thus, for example, a pronoun such as *he* itself denotes the identity function on individuals (more precisely, it probably denotes the identity function on male individuals, but we will ignore the contribution of gender here). An "open" expression such as *he lost* in (5) denotes a function from individuals to propositions; ignoring intensions it is thus a function of type $\langle e, t \rangle$ and its meaning is the same as the verb *lost*. (I will continue to ignore intensions throughout.)

The question then arises as to how the syntactic and semantic combinatorics work so as to ensure that *he lost* has this meaning. First, a comment on the syntax. To ensure that the syntax parallels the semantics, we will encode into the semantic type into the syntactic category. Thus just as a constituent like *he lost* does not denote a proposition (nor an "open proposition") but rather a function of type $\langle e, t \rangle$, so also will we encode into its syntactic category the fact that it contains an unbound pronoun. Thus, let any expression containing an unbound pronoun of category C have as its syntactic category A^C , where A is the category of a corresponding expression without an unbound pronoun. (Once again, by saying that the expression contains an unbound pronoun I mean that the pronoun is not bound within that expression, although it could well be bound from higher up.) Further, I include here also the case of an expression which is itself a pronoun. Hence *he* is of category NP^{NP} while *he lost* is of category SNP . Moreover, as we have already seen, any expression of category A^C denotes a function of type $\langle C, A \rangle$.

Continuing with our example of *he lost*, we have seen that *he* is of category NP^{NP} and denotes the identity function on individuals. In order for it to combine with *lost*, we will assume that *lost* type-shifts by the "Geach rule", given in (11):

- (11) Let α be an expression of category A/B. Then there is a homophonous expression β of category A^C/B^C , where $\beta' = \lambda V[\lambda c[\alpha'(V(c))]]$ (for V a variable of type $\langle C, B' \rangle$ and c a variable of type C').

Since the semantics of (11) is simply the "Geach rule" applied to α' I will sometimes notate this as $g(\alpha')$. (Note that g is a unary version of function composition; thus $g(g)(f) = g \circ f$.) Note further that the syntax is reminiscent of GPSG feature passing conventions: this type-shifts an expression into one which wants as argument a pronoun-containing expression, and it passes on to the result the information that there is an unbound pronoun. The syntax and meaning of *he lost* will thus be put together as shown in (12):

- (12) *lost*; $S/_LNP$; *lost'* \rightarrow_g *lost*; $S^{NP}/_LNP^{NP}$; $\lambda f[\lambda x[\text{lost}'(f(x))]]$ (for f of type $\langle e, e \rangle$)
he; NP^{NP} ; $\lambda y[y]$
he lost; S^{NP} ; $\lambda f[\lambda x[\text{lost}'(f(x))]](\lambda y[y]) = \lambda x[\text{lost}'(x)] = \text{lost}'$

Moreover, a phrase like *his mother* is of category NP^{NP} and denotes a function of type $\langle e, e \rangle$. More particularly, it will end up denoting that function which maps each individual into that individual's mother, which is to say it denotes the-mother-of function. (To show this out in detail requires a treatment of the semantics of genitives; space precludes this here, but see Jacobson (1994a) for details.)

Several important comments are in order. First, recall the generalization in (8) which says that the set of environments allowing expressions with no unbound pronouns is a subset of those which allow expressions with unbound pronouns in characteristic NP positions. As pointed out above, it is probably this generalization which leads to the "standard" intuition that such expressions should thus be of the same syntactic category and hence of the same semantic type, and which further leads to the intuition that (as-yet unbound) pronouns "pretend" to be like ordinary NPs. But notice that the generalization in (8) is equally well accounted for by the existence of the Geach rule (11). This rule says that any constituent which subcategorizes for a constituent of category A can instead take as argument a constituent of category A^C ; the information that there is an unbound pronoun here will also be passed up to the result. (Recall further that the reverse generalization does not hold - the environments in which resumptive pronouns occur appear in certain cases to be environments which require pronouns. This is no problem here: an environment requiring a resumptive pronoun will simply be one which subcategorizes for an A^{NP} ; this basic idea was proposed in Sells (1984).) Note too that the rule in (11) allows pronouns only to pass up from arguments and not from functions. However, one can easily add in additional conventions to allow passing from functions. Alternatively, if there is free type-lifting, then any function can become an argument, and so the pronoun can always be within either the argument or the (would-be) function. The latter is the solution that we will adopt here, although if there turns out to be reason to disallow free type-lifting then one could instead adopt the former solution. Finally, note that the theory proposed here may seem more complex than the standard theory in that here we require the Geach type-shift rule in (11). It is true that we do need this extra bit of apparatus, but it is rather simple and natural, and the trade-off is that we do not need the extra bit of apparatus of having variables and assignment functions as part of the model-theoretic machinery. Moreover, it is worth noting that under the standard theory the semantic

combinatorics are not quite as simple as they at first blush appear to be: as noted above, whenever two expressions α and β combine by "functional application" the actual semantic combinatorics is $\lambda g[\alpha'(g)(\beta'(g))]$.

We are now in a position to comment on the key differences between this and the standard account of binding. In the standard account, an expression which contains an (as yet unbound) pronoun has the same kind of meaning as a corresponding expression without such a pronoun. (As noted above, this is a large part of the motivation for variables - a pronoun "pretends" that it's just like a non-pronoun until it is ready to be bound.) Under the approach here, on the other hand, a pronoun has a different kind of meaning as an ordinary NP, and an expression containing an (as yet unbound) pronoun has a different kind of meaning from an expression without a pronoun. **Therefore, the effect of the pronoun is much more local: its effect on the semantic type and on the syntactic category is there in every expression which contains it.** This fact will become crucial below.

2.2.2. Binding in a Variable-Free Semantics

We have yet to show how a pronoun can ultimately be bound. The basic idea here is to build the effect of binding into a type-shift rule which shifts the syntactic category and meaning of the verb which takes both the binder and the pronoun-containing constituent as arguments. This rule is given formally in (13); I will refer to this as the z rule:

- (13) Let α be an expression of category (A/NP)/B and with meaning of type $\langle X, \langle e, Y \rangle \rangle$. Then there is a homophonous expression β of category (A/NP)/B^{NP} with meaning of type $\langle \langle e, X \rangle, \langle e, Y \rangle \rangle$, where $\beta' = z(\alpha')$. z is defined as follows: Let F be a function of type $\langle X, \langle e, Y \rangle \rangle$. Then $z(F)$ is a function of type $\langle \langle e, X \rangle, \langle e, Y \rangle \rangle$ where $z(F) = \lambda g[\lambda x[F(g(x))(x)]]$.

(Note: (13) is defined here in such a way as to bind only "NP" pronouns. I know of no reason not to generalize this, but this will suffice here.) The rule is given here only for the case of binding by the "next higher" argument. It can, however, easily be generalized to the case of, for example, binding by a subject into an indirect object; see Jacobson (1992a) for details.

The intuition behind (13) is easiest to appreciate by example. Consider the case of binding in (6) (*Every man_i loves his_i mother.*) *Love* denotes an ordinary 2-place relation between individuals (i.e., it denotes a function of type $\langle e, \langle e, t \rangle \rangle$). In this case, though, it shifts by (13) to denote the relation $z(\text{love})$, where this is a relation of type $\langle \langle e, e \rangle, \langle e, t \rangle \rangle$ such that to $z(\text{love}) f$ is to be an x who ordinary-loves $f(x)$. The syntax is parallel: the shifted verb wants not an ordinary NP object but rather an object of category NP^{NP} - i.e., an object containing a pronoun. Recall further that *his mother* is of category NP^{NP} and denotes the mother-of function. Hence when $z(\text{love})$ combines with *his mother*, the resulting VP denotes the set of x 's who stand in the $z(\text{love})$ relation to the mother-of function - this is the set of x 's who love x 's mother. This VP then occurs as argument of the generalized quantifier *every man*. Similarly for (7). Here we begin with *think* which denotes a relation between individuals and propositions. It shifts by z to denote a relation between individuals and properties such that to $z(\text{think}) P$ is to be an x who thinks $P(x)$. Moreover, again the syntax is parallel: the shifted verb is of category (S/LNP)/R^{SNP}. The complement *he lost* is of category S^{NP} and denotes the

property lost'; when this is taken as argument of $z(\text{think}')$ the resulting VP denotes the set of x 's who stand in the $z(\text{think}')$ relation to lost', which in turn is the set of x 's who think that x lost. Then this occurs as argument of the subject.

It is worth pointing out that positing the type-shift rule z in (13) involves no machinery above and beyond that which is also needed in the standard view. For even in the latter, a type-shift rule is needed to effect binding; two possibilities were explored above in (9) and (10). But there is a key difference between the two accounts. Here, the type-shift rule - and hence the effect of binding - is much more local; it operates on a much "smaller" expression. As noted above, in the standard account the type-shift rule operates on the meaning of a constituent containing a pronoun just before that expression combines with the binder. (Hence, it applies at either the VP level or at the S level). Here, on the other hand, **the binding effect is located more locally: it is the result of a shift on the meaning of a verb (or other function) which wants two arguments - one being the pronoun-containing constituent and the other being the binder.** (Hence in *Every man loves his mother*, for example, the binding effect operates on *loves*, while in the standard account it operates on either the meaning of the VP or the whole S.) This difference will also be crucial in the remarks below. Moreover, it will be shown in Sec. 9 that this kind of locality crucially is tied in to a variable-free semantics. That is, one could attempt to mimic this in a semantics with variables, but only by essentially undermining the entire rationale for having variables in the first place: were one to try to mimic this locality and still have variables, the variables would perform no real work and would only "get in the way".

Finally, a remark is in order about free (unbound) pronouns, as in a sentence such as *He lost*. I assume that these are no different from ordinary pronouns which are eventually bound - it is a striking fact that free and bound pronouns (in English at least) are morphologically identical, and I take this as evidence that the two should probably be treated alike. Thus here too we can assume that lost' undergoes g and takes he' as argument. The result is that *he lost* denotes not a proposition, but again only the property lost'. Of course in order to extract information from this the listener must presumably get some propositional information out of this property; a plausible story is that the listener thus applies this function to some contextually salient individual. At first blush, the conclusion that *he lost* denotes a property and not a proposition might seem rather counterintuitive. But a moment's reflection reveals that the standard account must make an analogous move. Note that entailments, etc., are defined on propositions - not on functions from assignment functions to propositions. But in the standard account, all sentences actually denote functions from assignment functions to propositions. Most such sentences, however, are closed expressions - they denote constant functions and so we can assume that entailments are arrived at by "stripping away" the assignment functions. (I thank Ed Keenan for this observation.) But in the case of an open sentence like *he lost*, the denotation is not in fact a constant function. Nonetheless, in normal discourse, a listener can extract certain entailments and other sorts of propositional information from this. How? Presumably by applying this to some contextually salient assignment function (or, to some equivalence class of assignment functions which all agree on the value that they assign to \underline{x}). Such an account of the essentially "propositional" nature of *he lost* seems no more counterintuitive than an account which claims that the listener applies a property to a contextually salient individual.

2.3. *The central claims here*

The central claims which I hope to elucidate in the remainder of this paper, then, are as follows:

(i) The local interpretation hypothesis requires this kind of locality in the effect of the pronoun and in the location of binding. The reason is that natural language syntax is such that "small" expressions can "move around" or stand alone (without surrounding material) in a variety of constructions. When this happens, these "small" expressions act with respect to binding phenomena in just the same way that they do when surrounded by more material. Thus in order to maintain the hypothesis of local interpretation, we need to build the effect of the pronoun and the effect of binding into these small expressions. Hence direct model-theoretic interpretation (without LF) entails the hypothesis of local interpretation, which in turn entails this kind of local approach to pronoun meaning and to binding. And this in turn entails either a variable-free semantics or, as will be discussed in Sec. 9, a semantics with variables but where they do no real work.

(ii) Alternatively, we could give up the hypothesis of local interpretation (and hence the hypothesis of direct model-theoretic interpretation). For the relevant phenomena, we could instead posit a level of representation at which the "small" expressions are indeed surrounded by more stuff, and assume that binding applies to that level of representation. However, it will be shown below that the interaction of Right Node Raising and binding argues against this, and thus we will provide positive evidence for all three claims elucidated above.

3. Independent Motivation

A number of pieces of motivation for a variable-free semantics and for this particular implementation are provided in Jacobson (1991, 1992a, 1992b, 1994a, 1994b). These arguments center on a variety of phenomena, including the analysis of functional questions and their answers; the analysis of certain kinds of unexpected inferences which have been noted in Higginbotham (1992), Chierchia (1990), Reinhart (1990) and others; the account of binding connectivity in copular sentences; *i*-within-*i* effects; the effects of Weak Crossover in Bach-Peters sentences which were originally noted in Jacobson (1977); and the interaction of binding and Antecedent Contained Deletion. Obviously space precludes reviewing these arguments here, but there is one worth reviewing as it will set the stage for the discussion of Right Node Raising. This concerns the analysis of functional questions and of their answers.

Consider a functional question such as (14a) under the reading where (14b) is an appropriate answer:

- (14) a. Who does every Englishman_{*i*} love?
b. His_{*i*} mother.

Under the analysis of these in Groenendijk and Stokhof (1983) and Engdahl (1986) the meaning of (14a) is as shown (informally) in (15):

- (15) what is the function *f* (of type $\langle e, e \rangle$) such that:
every-Englishman'(λx [love'(f(x))(x)])

I will assume that this is indeed the meaning of (14a), but the question at issue here is just how the compositional semantics works so as to yield that meaning.

Under the analyses developed in Groenendijk and Stokhof (1983) and Engdahl (1986), the trace or "gap" in object position in (14) has the meaning $f(x)$. Moreover, the individual variable x is bound by the normal mechanisms for variable-binding (hence by a rule such as either (9) or (10) above). Further, there will be some rule which λ -abstracts over the f variable and takes the result to be the argument of the question pronoun *who*'; the details of this will presumably depend on one's analysis of questions in general.

But there is at least one and possibly two unsatisfying aspects to this analysis. The first is somewhat theory-internal and is relevant to any theory such as (most versions of) Categorical Grammar (as well as some versions of G/HPSG) in which there are no traces and in which an extraction gap is simply the failure of an expected argument to be introduced in the expected way. Under theories like this, there is no trace or any other expression which can be assigned the complex meaning $f(x)$ - and thus it is not clear just how the gap can indeed be given this meaning. To make this more concrete, consider the view of extraction in, e.g. Steedman (1987), where an extraction "gap" is just the result of function composition of the verb and the subject. Since in this case the verb is expecting only an NP object and is of type $\langle e, \langle e, t \rangle \rangle$, the result of composing *every Englishman* with *loves* will yield the meaning $\lambda x [\text{every-Englishman}'(\text{love}'(x))]$ and not the meaning shown in (15). Similar remarks will apply to any theory of extraction in which the "gap" is nothing more than the failure of some expected argument to be introduced in the normal way - under any such view, there is nothing which can be assigned the complex meaning.

Of course, if one allows for traces in the syntax then this argument does not hold - it is the trace which is assigned the complex meaning $f(x)$ rather than being assigned the simple meaning x . Nonetheless, even under such a view there is one unsatisfying aspect to the analysis: we need something extra to allow not only for individual traces, but also for traces with complex (structured) meanings: these traces denote functions of type $\langle e, e \rangle$ applied to arguments of type e . (It should be noted that my main objection here is not that the trace can correspond to a variable of type $\langle e, e \rangle$ as well as to a variable of type e , because there are various ways to generalize the meaning of traces such that this result is quite natural. But the "extra" piece of machinery needed here is that the trace does not denote simply a variable over functions of type $\langle e, e \rangle$ - it must denote such a variable applied to an individual level variable.) The same point would hold for a trace-free syntax where one instead tried to introduce the variables in the semantics by means of a type-shift rule. For example, one could presumably compose the meaning of (14a) by allowing *love* to typeshift from its meaning *love'* to a new meaning *love'*($f(x)$) (without having to posit a trace to supply the meaning $f(x)$). But again there is no obvious mechanism which would generalize so as to supply not only the variable f but also its argument x .

Now compare this to the account of (14a) under the variable-free approach. The key point to note here is that nothing extra is needed in order to get *every Englishman loves* to have the desired meaning - the fact that a functional reading is possible follows directly from the mechanisms for variable-binding in general. We do not need to translate the "gap" as a complex meaning $f(x)$ - the gap can be treated simply as the failure of an argument to be introduced in the expected way. But the fact that it can correspond to a function of type $\langle e, e \rangle$ is simply a consequence of the fact that *loves* can type-shift by z . In other words, the meaning of (14a) can be paraphrased as (16a) which is equivalent to (15); (16b) illustrates the compositional semantics in more detail under a Steedman-style approach to extraction:

- (16) a. what is the function f such that every-Englishman'(z(love')(f))?
 b. *every-Englishman-loves'* = every-Englishman' o z(love') =
 $\lambda f[\text{every-Englishman}'(\lambda x[\text{love}'(f(x))(x)])]$
 (this then occurs as argument of who')

Thus we do not need to posit traces to assign them the complex meaning $f(x)$. Indeed, no special mechanism is needed to get the functional reading beyond that used for variable-binding in general - it is the fact that *love* can undergo z which gives rise to the functional reading. (Actually, it is not quite true that the functional reading comes "for free" from the general mechanism for variable-binding; we need the question pronoun to be polymorphic here such that it can take as argument a function of type $\langle\langle e, e \rangle, t \rangle$ as well as a function of type $\langle e, t \rangle$. But the same is true under the Groenendijk and Stokhof/Engdahl analysis and, as far as I can tell, would be true under any analysis in which the meaning of (14a) is as in (15).)

The reason this analysis works so smoothly is precisely because the type-shift which accomplishes binding is far more local than in the standard account. In the standard account, we need an expression at least as big as a full VP to undergo the shift rule (9 or 10) which accomplishes binding. And since the type shift rule λ -abstracts over the variable which is unbound in the input to the rule, we need to somehow supply that individual as part of the meaning of the "VP" (or, V) *love* in (14). What we have seen above, though, is that something extra will be needed to supply this variable. Here, on the other hand, we have located the binding effect locally on the meaning of *love*. As a result, there is no need to elaborate the syntactic representation and/or meaning of *love* in such a way as to surround it with the variable x or with a trace whose translation can contain the relevant variable.

On the other side of the coin, let us turn now to the analysis of answers to functional questions. Under the standard view, it is actually somewhat of a mystery as to why (14b) (*his mother*) is an appropriate answer to a functional question. After all, recall that this does not denote a function of type $\langle e, e \rangle$ (which is what the question is asking for), but rather an individual (actually, a non-constant function from assignment functions to individuals). We thus need some kind of rule to type-shift this meaning into a function of type $\langle e, e \rangle$ in order to explain why this can serve as answer to a functional question. Indeed, this is exactly the tack taken in Gawron and Peters (1990): they posit a special rule allowing certain "open individuals" to shift into functions of type $\langle e, e \rangle$. As such, the meaning of *his mother* is the-mother-of(x) but it can shift by this rule into $\lambda x[\text{the-mother-of}(x)]$. This, however, requires an additional rule.

There is one caveat here. von Stechow (1990) points out that the relevant type-shift rule can actually be seen as a generalization of the binding type-shift rule given in (10); thus let any expression containing an unbound variable shift by λ -abstraction over that variable. (von Stechow was not concerned with answers to functional questions, but his remarks generalize directly to this case.) Under this view, the shift from the-mother-of(x) to $\lambda x[\text{the-mother-of}(x)]$ does not in fact involve extra machinery. Note, though, that this makes sense only under a Binders Out approach in general - since it is only under such an approach that the rule in (10) is motivated. But that in itself requires extra machinery (and also, of course, crucially denies the hypothesis of local interpretation). Of course quantifier scope phenomena have long been taken to provide independent motivation for a Binders Out approach; but recent work in Categorical Grammar has shown that scope itself can be handled even when all quantified NPs are interpreted in situ (see, e.g., Hendriks, 1987).

There are, then, perhaps some open questions as to whether or not the answers to functional questions require extra apparatus under the standard account - whether or not it does depends on what other assumptions one makes. But in any case, we can note that under the approach here it is clear that nothing extra is needed: it follows immediately that *his mother* has the right kind of meaning to serve as answer to the functional question. The reason is, of course, that this is a function of type $\langle e, e \rangle$ - in particular, it denotes the-mother-of function. Once again, the key centers on the locality property of the pronouns in the variable-free approach: the semantic effect of an (as-yet unbound) pronoun is "felt" in any constituent containing that pronoun - thus *his mother* is not like an ordinary NP but is necessarily a function of type $\langle e, e \rangle$. Again, the conventional wisdom behind variables is that pronouns "pretend" to be like ordinary NPs until the point in the semantic composition where they are ready to be bound. The result of this is that *his mother* should denote an ordinary individual and not a function of the appropriate type.

4. Conjunction and Across-the-Board Binding

We are now in a position to see why the interaction of conjunction (specifically, RNR constructions) and binding provides some rather striking evidence for this approach to binding and, in fact, for all three claims elucidated in Sec. 1. The exposition here will take place in several steps. This section will discuss the phenomenon of "Across the Board" (ATB) binding. I will show here that this phenomenon is quite similar to the case of functional questions - under the variable-free semantics, this phenomenon comes "for free". It is, moreover, completely compatible with the analysis of RNR which maintains the hypothesis of local interpretation and direct model-theoretic interpretation (without a mediating level). In Secs. 5 and 6, I will show that under the standard view of variables, this phenomenon requires either a reconstruction analysis of RNR or one making use of functional traces (or an equivalent type-shift rule). However, it will be shown that both of these solutions suffer from a serious problem: both allow for binding out of just one conjunct which, in fact, is impossible. In Sec. 7 we will return to the variable-free account, to show that the problem does not arise here.

The existence of ATB binding has been noted from time to time in the literature; see, e.g. Dahl (1981), Hohle (1990), von Stechow (1990):

- (17) 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 first noteworthy point about (17) is that under the variable-free analysis discussed above (combined with the analysis of RNR discussed in Sec. 1), nothing more needs to be said - the existence of the ATB reading is an automatic consequence of the analysis. Informally, the analysis of (17) proceeds in essentially the same way as does the analysis of (1) which was sketched in (3). The only difference is that here the subject function composes with $z(\text{loves})'$ rather than with loves'. Thus the first conjunct - *every man loves* - denotes the set of functions f of type $\langle e, e \rangle$ such that every man stands in the z -love relation to f . Similarly, *no man hates* denotes the set of functions g such that no man z -hates g . Hence the full conjoined expression - *every man loves and no man hates* - denotes the function characterizing the intersection of these two sets. Furthermore, *his mother* is of course of the right type to serve as argument of this function. Because it contains a pronoun it is of type $\langle e, e \rangle$, and denotes the-mother-of function. The entire derivation is given in detail in (18):

- (18) $\text{every-man-loves}' = \text{every-man}' \circ z(\text{love}') =$
 $\text{every-man}' \circ \lambda f[\lambda x[\text{love}'(f(x))(x)]] = \lambda f[\text{every-man}'(\lambda x[\text{love}'(f(x))(x)])]$
 $\text{no-man-marries}' \text{ (similarly)} = \lambda g[\text{no-man}'(\lambda y[\text{marries}'(g(y))(y)])]$
 $\text{every-man-loves-and-no-man-marries}' =$
 $\lambda f[\text{every-man}'(\lambda x[\text{love}'(f(x))(x)])] \lambda g[\text{no-man}'(\lambda y[\text{marries}'(g(y))(y)])]$
 $= \lambda f[\text{every-man}'(\lambda x[\text{love}'(f(x))(x)]) \text{ no-man}'(\lambda y[\text{marries}'(f(y))(y)])]$
 $\text{his-mother}' = \lambda x[\text{the-mother-of}(x)] = \text{the-mother-of}$

Again, the reasons that this proceeds so smoothly and works directly with the surface syntax is due to the locality of the binding effect and to the locality of the effect of pronoun meaning. To clarify this point, let us consider the implications of ATB binding for a standard theory with variables. Consider again how binding works for a simple case like (5) (*Every man_i loves his_i mother*) in the standard view. As discussed in Sec. 2.1.2, binding under the standard view is the result of a type-shift rule which operates on the meaning of an expression containing a pronoun, and operates just before this expression combines with the binder. Thus it operates either on the meaning of the entire VP *loves his mother* or on the meaning of the entire S under a Binders Out approach. Since the latter requires an even bigger domain than the former, we will confine our remarks to the former approach. The problem posed by a sentence like (17) is obvious - here there simply is no (apparent) surface VP to undergo the binding type-shift rule. But under the variable-free view there is no problem - we do not need a full VP in order to accomplish the effect of binding. Rather, here binding is the effect of a very local type-shift rule which operates on the meanings of *love* and *marry*. Thus the fact that these expressions on the surface do not appear to be surrounded by additional material is no problem - the locality of binding here means that we do not need to posit additional material surrounding these verbs in order to effect binding. (Moreover, the fact that *his mother* has the right meaning is also a consequence of the fact that pronouns always "make their effect known" in every local domain in which they occur; the relevant remarks here are exactly the same as those made in the previous section with respect to the analysis of the answers to functional questions.)

Under the standard account, then, we are forced to do one of two things. One possibility is to give up the hypothesis of local interpretation, and posit a reconstruction level at which the "Right Node Raised" constituent (*his mother*) is in the position of each gap. The other would be to maintain local interpretation, but to assign a complex meaning to the gap - in essence, to treat these analogously to the account of functional questions in Groenendijk and Stokhof (1983) and Engdahl (1986). We will turn to each of these alternatives below, and show that both have a serious defect.

5. ATB Binding Under Reconstruction

A proponent of the standard view of binding might conclude that the existence of the ATB binding reading in (17) simply shows that the hypothesis of local interpretation is incorrect and that, rather, the interpretation of RNR sentences involves positing a "reconstruction" level at which copies of the Right Node Raised

constituent is in the position of each gap. The basic idea, then, is that (17b), for example, is mapped into (or, derived from) a representation like (19) and that the binding effect takes place at this level. Here we have two full VPs, and so there is no problem with either the view of binding given in (9) or in (10):

(19) Every man loves his mother and no man marries his mother.

The basic idea here could be implemented in a variety of ways - we will pick one for the sake of illustration (all varieties which I have been able to think of have the same basic problem). First, assume that LFs are derived from surface structures (rather than the other way around). Second, assume a theory where indexing conventions first apply to some level of representation, and the indexed representations are ultimately mapped into final LFs where indexed pronouns correspond to variables in the obvious ways. Finally, one can imagine reconstruction applying first and then indexing as in (20), or the processes could apply in the opposite order as in (21)::

- (20) a. Every man loves and no man marries his mother. ==> (reconstruction)
 b. Every man loves his mother and no man marries his mother.
 ==> (indexing)
 c. Every man_i loves his_i mother and no man_j marries his_j mother.
- (21) a. Every man loves and no man marries his mother. ==> (indexing)
 b. Every man_i loves and no man_j marries his_j mother.
 ==> (reconstruction)
 c. Every man_i loves his_i mother and no man_j marries his_j mother.

I will consider only the second possibility here; the reader can verify that the problem(s) to be discussed below hold equally well for the first alternative.

The problem which arises here is one which is noted (with respect to slightly different kinds of examples) in Hohle (1990) and von Stechow (1990). This is that there is no reading where the pronoun is understood as bound within the interpretation of one of the conjuncts but is understood as free in the interpretation of the other. In other words, (22a) does not have the reading shown in (22b). But this should be possible, as there is no obvious way to block the derivation shown in (23) (note that (23c) gives rise to the reading in (22b)):

- (22) a. Every man loves and no man marries his mother. ≠
 b. Every man_i loves his_i mother and no man_j marries his_k mother.
- (23) a. Every man loves and no man marries his mother. ==> (indexing)
 b. Every man_i loves and no man_j marries his_j mother.
 ==> (reconstruction)
 c. Every man_i loves his_i mother and no man_j marries his_j mother.

As an initial attempt at a solution, suppose we were to assume that free pronouns simply are indexed in some way which is entirely different from the indices for bound pronouns (this, in fact, is essentially the solution taken in von Stechow, 1990, although he was adopting a view like that explored in the next section). In the first place, though, this seems suspicious in view of the fact noted above that morphologically free and bound pronouns are identical (in English, at least). More seriously, however, is the fact that the problem here is much more general: one cannot have the "ATB pronoun" interpreted as being bound within the interpretation of one of the conjuncts but being bound from outside in the

interpretation of the other. In other words, (24a) can't have the reading shown in (24b). Again this should be possible since there is no obvious way to block the derivation in (25).

- (24) a. Each boy believes that every man loves and (that) no man marries his mother. \neq
 b. Each boy_k believes that every man_i loves his_i mother and (that) no man_j marries his_k mother.
- (25) a. Each boy believes that every man loves and (that) no man marries his mother. \implies (indexing)
 b. Each boy_i believes that every man_i loves and (that) no man_j marries his_i mother. \implies (reconstruction)
 c. Each boy_i believes that every man_i loves his_i mother and (that) no man_j marries his_j mother.

Note that (25c) gives rise to the reading shown in (24b) (given reasonable assumptions about what these indexed structures mean), because in the interpretation of the first conjunct *his* will be bound by *every man* as that is the closest c-commanding co-indexed NP, while in the second conjunct it is bound *each boy*. Note further that since both pronouns are bound here, the solution to (24) cannot rely on anything having to do with the treatment of free pronouns since here both pronouns are, in fact, bound. One could, or course, imagine various ways to try to rule out the derivation in (25). One obvious way which comes to mind is to rule out the LF in (25c) by a constraint against having the same index on two NPs if one c-commands the other (hence *each boy* and *every man* in (25c) cannot be coindexed). Although this works, it has two problems. First, while it would rule out (25c) it would be of no help for the case in (22) - thus this would still have to be combined with an account which treats free pronouns differently from bound pronouns. Second, such a constraint on co-indexation is an added stipulation. One might counter this objection by claiming that it is a very "natural stipulation", but I would argue here that a "natural stipulation" simply means something which should follow *without* stipulation given a deeper understanding of the phenomenon in question. But as long as we are relying on indices and/or their corresponding translation into variables, the fact is that this stipulation does not follow from anything else.

One final attempt at a solution for both (22) and (24) would account for both via some sort of "parallelism" constraint on the position of the binder vis-a-vis the reconstruction site. Again, though, any such solution that I can imagine will require an additional stipulation. Along these lines, note that the grammaticality of (26) cast some doubt on the existence of any kind of "parallelism" constraint; a final case that casts doubt on this is discussed in Sec. 8:

- (26) a. Every man_i thinks that the world ought to love but indeed no man_j is prepared to marry his_{i/j} mother.
 b. No man_i actually loves although Mary thinks that every man_j ought to love his_{i/j} mother.
 c. Every man_i says that the whole world should love although Mary knows that in fact no man_j (himself) actually loves the woman who brought him_{i/j} into this world.

Thus the fact that (22b) and (24b) are both impossible - combined with the grammaticality of (26) - suggests that the appropriate generalization is as follows:

- (27) Binding is not possible out of one conjunct unless there is binding out of both. (This is also discussed, although with a somewhat different class of cases, in Chierchia (1988).)

This, of course, is reminiscent of Coordinate Structure Constraint on extraction, but it is interesting to note that the effect here is far more robust than run-of-the-mill Coordinate Structure Constraint effects on extraction. One might well argue - as Lakoff (1986) and others have done - that Coordinate Structure Constraint effects on extraction are actually pragmatically based. Here, however, no amount of "pragmatic tinkering" seems to improve sentences like (22b) and (24b). As evidence that the effect here indeed is not a pragmatic one, Maria Bittner (personal communication) has offered the following sort of sentence, whose pragmatics demand the reading of the type in (24b) but where this reading is nevertheless robustly impossible:

- (28) *Every man_i thought that every other man_j had already deposited and that the bursar_k still had his_j_i paycheck.

6. ATB Binding with functional gaps

What we have seen so far, then, is that one way to maintain the standard theory of variables in view of the existence of ATB binding would be to posit a level derived by reconstruction. Such a solution, of course, abandons the hypothesis of local interpretation and, concomitantly, the hypothesis of direct model-theoretic interpretation. But this strategy provides no natural account of the generalization in (27). There is, however, another strategy which one might try here: this strategy remains compatible with the standard view of variables while also still maintaining the hypothesis of local interpretation. This involves a simple extension of the Groenendijk and Stokhof / Engdahl analysis of functional questions. In fact, just this analysis is explored in von Stechow (1990).

Thus in (17), one could assume that each conjunct contains a trace in the position of the "gap", and that each of these traces is interpreted as $f(x)$, exactly as in the Groenendijk and Stokhof / Engdahl analysis of questions. (Note once again that traces in the syntax are not really crucial here: one could have a type-shift rule shifting the meaning of *love* from *love'* into *love'(f(x))* and similarly for *marry*. What is crucial, though, is the abandonment of the idea that the RNR "gap" is nothing more than a missing argument - thus the functional gap analysis is not, for example, compatible with the Dowty/Steedman analysis of RNR.) Furthermore, each individual variable will be bound by the subject in just the way that variable-binding normally proceeds. As to the meaning of the Right Node Raised constituent itself, we will again need to assume that it translates as *the-mother-of'(x)* and that x is then λ -abstracted over in such a way as to convert this into the function $\lambda x[\text{the-mother-of}'(x)]$ which is of course equivalent to *the-mother-of'* function. The detailed derivation is sketched in (29):

- (29) *loves* $t_f(t_x)$; *loves'*($f(x)$) --> (shift by binding rule in (9))
 $\lambda x[\text{loves}'(f(x))(x)]$
every man loves $t_f(t_x)$; *every-man'*($\lambda x[\text{loves}'(f(x))(x)]$)
 similarly for *no man marries* $t_f(t_x)$; *no-man'*($\lambda y[\text{marries}'(f(y))(y)]$)
every man loves $t_f(t_x)$ and *no man marries* $t_f(t_x)$;
 $\text{every-man}'(\lambda x[\text{loves}'(f(x))(x)]) \wedge \text{no-man}'(\lambda y[\text{marries}'(f(y))(y)])$
 --> (λ -abstraction over the f variable, by presumably a general process
 which would apply in the case of RNR in general):
 $\lambda f[\text{every-man}'(\lambda x[\text{loves}'(f(x))(x)]) \wedge \text{no-man}'(\lambda y[\text{marries}'(f(y))(y)])]$
his mother; *the-mother-of*(x) --> $\lambda x[\text{the-mother-of}(x)]$
 (this constituent can then serve as argument of the conjoined material)

So far, this requires nothing new above and beyond what is needed for the case of functional questions and their answers under the Groenendijk and Stokhof / Engdahl analysis. Under this kind of analysis of questions, we in any case need traces which translate as $\underline{f}(x)$ (or a type-shift rule with the same effect), and we also need a way to type-shift constituents like *his mother* from "open individuals" to functions of type $\langle e, e \rangle$.

But the interesting point to note about this approach is that it has exactly the same problem as the reconstruction analysis: it too does not account for the generalization in (27) without further stipulations. We will show this in detail momentarily, but in order to arrive at the intuition here one can note that the analysis of the two conjuncts under this approach is isomorphic to the post-reconstruction representation of the two conjuncts under the reconstruction theory. The reconstruction theory posits a constituent such as *his_i mother* in each conjunct and the problem is that one occurrence of *his* can be bound within one conjunct while the other can be free within the second conjunct. Here each conjunct contains as part of its meaning the sequence $\underline{f}(x)$ and the same problem emerges: \underline{x} can be bound within one conjunct and not within the other. (Note, then, that the \underline{f} variable in this analysis is analogous to *mother* in the reconstruction analysis, while \underline{x} is analogous to *his*.)

To flesh this out in greater detail, consider first the case in (22b) where one variable is bound by one subject and the other is free. That is, there is no obvious way to block the derivation shown in (30) (this assumes the VP-level binding rule in (9); similar remarks would apply to the Binders Out approach shown in (10)):

- (30) *loves* $t_f(t_x)$; *loves'*($f(x)$) --> (binding rule in (9)) $\lambda x[\text{loves}'(f(x))(x)]$
every man loves $t_f(t_x)$; *every-man'*($\lambda x[\text{loves}'(f(x))(x)]$)
marries $t_f(t_x)$; *marries'*($f(x)$)
no man marries $t_f(t_x)$; *no-man'*($\text{marries}'(f(x))$)
every man loves $t_f(t_x)$ and *no man marries* $t_f(t_x)$;
 $\text{every-man}'(\lambda x[\text{loves}'(f(x))(x)]) \wedge \text{no-man}'(\text{marries}'(f(x)))$

The derivation will be completed by λ -abstraction over \underline{f} and by application of the result to the argument $\lambda x[\text{the-mother-of}(x)]$. This yields the meaning in (22b) where the \underline{x} variable within the first conjunct is bound by its subject, but where the \underline{x} variable within the second conjunct remains free. Note again that it will not do to simply treat free pronouns in a different manner, since the same problem extends to the case of (24b) where both pronouns are bound, but where one is bound from

higher up. To demonstrate this, we can note that there is obvious way to block the following derivation:

- (31) *every man loves* $t_f(t_x)$ *and no man marries* $t_f(t_x)$;
 every-man'(λx [loves'(f(x))(x)]) \wedge no-man'(marries'(f(x))) \rightarrow
 (λ -abstraction over f):
 λf [every-man'(λx [loves'(f(x))(x)]) \wedge no-man'(marries'(f(x)))]
every man loves $t_f(t_x)$ *and no man marries* $t_f(t_x)$ *his mother*;
 λf [every-man'(λx [loves'(f(x))(x)]) \wedge no-man'(marries'(f(x)))]
 (the-mother-of) =
 every-man'(λx [loves'(the-mother-of(x))(x)]) \wedge no-man'(marries'
 (the-mother-of(x)))
believes that every man loves $t_f(t_x)$ *and no man marries* $t_f(t_x)$ *his mother*;
 believes'(every-man'(λx [loves'(the-mother-of(x))(x)]) \wedge
 no-man'(marries'(the-mother-of(x)))) \rightarrow (binding rule in (9))
 λx [believes'(every-man'(λx [loves'(the-mother-of(x))(x)]) \wedge
 no-man'(marries'(the-mother-of(x))))]

The final line here occurs as argument of the matrix subject *no boy*; the result is that this subject binds the x variable in the second conjunct, while the x variable in the first conjunct is bound by the embedded subject *every man*. The problem here is exactly analogous to the problem under the reconstruction view: the interpretation of each of the two "gaps" should be independent of each other, but in fact they are not - if one contains a variable bound within one conjunct, so must the other. Notice that the two gaps themselves have the same meaning - both mean $f(x)$, and so there is no obvious reason why RNR should care that one x variable is bound within one conjunct while the other is not.

It appears that the problem here illustrates a fundamental mistake in the standard motivation for variables. The intuition which one would like to grab hold of to account for the generalization in (27) is as follows: If one conjunct contains a variable as part of the "gap" meaning which is bound within that conjunct while the corresponding variable in the other conjunct is not bound within the its conjunct, the two conjuncts should be of different semantic types - and therefore being unconjoinable. But this is exactly the intuition which the standard theory of variables denies. Recall that one of the central motivations for variables is to allow their presence not to be felt until they are just ready to be bound: an as-yet unbound pronoun/gap/variable/trace should be just like an ordinary NP or an already bound pronoun/gap/variable/trace. Hence, the standard account with variables must quite crucially treat the two conjoined Ss here as being of the same semantic type, even though one contains only bound (individual level) variables and the other contains an as-yet unbound individual variable. The intuition that the two conjuncts are of different semantic types (and hence unconjoinable) is, however, exactly the intuition which emerges immediately under the variable-free account (although in slightly different terms), and it is to this that we now turn.

7. The ATB Generalization in a Variable-Free Semantics

As noted above, the generalization in (27) emerges as an immediate consequence of the particular variable-free semantics developed here and has to do with the fact that the two conjuncts are not of the same semantic type (nor of the same syntactic

category). In a nutshell, the reason is that one of the conjuncts is "looking to bind" while the other is not.

To clarify, we return to the Steedman/Dowty analysis of RNR and thus dispense with both reconstruction and with functional gaps. Consider the analysis of the second conjunct (*that no man marries*) in a sentence like (24b) under the intended (bad) reading. Recall that what is trying to happen under this reading is that this conjunct ultimately would get a pronoun-containing constituent as argument (the argument being the Right Node Raised constituent), but in this case the pronoun will not be bound by *no man*. Rather, this expression wants to "pass the binding job" on up - hence this expression has undergone the Geach operation *g*. What this means, then, is that its semantic type is $\langle\langle e, e \rangle, \langle e, t \rangle\rangle$ and its syntactic category is SNP/RNP^{NP} . But now consider the first conjunct (*that every man loves*). This conjunct is also looking for a pronoun-containing constituent as argument (and hence, an argument of syntactic category NP^{NP} and of semantic type $\langle e, e \rangle$), but here the pronoun will be bound by *every man* - in this case, *loves* has undergone *z*. Thus the semantic type of this conjunct is just $\langle\langle e, e \rangle, t \rangle$ and the syntactic category is, correspondingly, just S/RNP^{NP} . The full derivations of each conjunct are spelled out in (32):

- (32) *loves*; $(S/LNP)/RNP$; *loves'* \rightarrow_z *loves*; $(S/LNP)/RNP^{NP}$; $z(\textit{loves}')$ =
 $\lambda f[\lambda x[\textit{loves}'(f(x))(x)]]$
every man; $S/R(S/LNP)$; *every-man'*
every man loves; S/RNP^{NP} ; *every-man' o z(\textit{loves}') =
 $\lambda f[\textit{every-man}'(\lambda x[\textit{loves}'(f(x))(x)])]$ (type $\langle\langle e, e \rangle, t \rangle$)
no man; $S/R(S/LNP)$; *no-man'* *marries*; $(S/LNP)/RNP$; *marries'*
no man marries; S/RNP ; *no-man' o marry'* = $\lambda x[\textit{no-man}'(\textit{marry}'(x))]$
 \rightarrow_g
no man marries; SNP/RNP^{NP} ; $\lambda f[\lambda y[\lambda x[\textit{no-man}'(\textit{marry}'(x))](f(y))]]$ =
 $\lambda f[\lambda y[\textit{no-man}'(\textit{marry}'(f(y)))]]$ (type $\langle\langle e, e \rangle, \langle e, t \rangle\rangle$)
 (Note: this meaning for *no man marries* can also be derived by Geaching first on both *no man* and on *marries* and function-composing the results.)*

The two expressions, therefore, cannot be conjoined and there is no way to get the impossible reading shown in (24b). These remarks hold equally well for both (22b) and (24b). The fact that in one case the pronoun is bound from higher up while in the other case the pronoun remains free is irrelevant since one cannot even put together the two conjuncts in the necessary way.

This is not to say that expressions with different semantic types can never be conjoined; the reader may have noticed an obvious apparent problem for the account here. This is that while binding out of just one conjunct is bad, binding into just one conjunct is of course fine:

- (33) Every man_i thinks (that) he_i lost and (that) Mary won.

In fact, though, sentences of this sort pose no problem for the account here as long as *and* is listed in the lexicon as being of category $(X/LX)/RX$ with meaning $\lambda A[\lambda B[B \text{oe} A]]$. In this case, binding into just one conjunct is just like binding into any other argument of a function. Here *and (that) Mary won* will combine in the normal way to give an S/LS , and this expression can then undergo the Geach rule; this is shown in (34):

- (34) *and*; (S/LS)/RS; $\lambda p[\lambda q[q \wedge p]$ *Mary won*; S; *won'(m)*
and Mary won; S/LS; $\lambda q[q \wedge \text{won}'(m)]$ --->g
and Mary won; $\text{SNP}/\text{L}\text{SNP}$; $\lambda P[\lambda x[\lambda q[q \wedge \text{won}'(m)](P(x))]] =$
 $\lambda P[\lambda x[P(x) \wedge \text{won}'(m)]]$
(that) he lost; SNP ; *lost'*
that he lost and Mary won; SNP ; $\lambda x[\text{lost}'(x) \wedge \text{won}'(m)]$
thinks undergoes *z* and takes the above as argument

(If the pronoun had been in the second conjunct matters would be slightly more complicated - but only because in this case the pronoun is passed up from the would-be function rather than the would-be argument. The first conjunct, then, would have to type-lift over the second and then undergo the Geach rule; we leave it to the interested reader to work out the details. The important point, however, is that although the derivation may look complex, no new apparatus is needed for this case.)

Thus it is an oversimplification to say that two expressions can conjoin only if they are of the same semantic type (and syntactic category), as witnessed by (33). Here the fact that binding is possible into just one conjunct is a special case of the fact that binding is possible into arguments in general - (33) simply involves binding into the argument of *and*. But the ATB effects of concern here is the generalization in (27) - which says that it is impossible to have binding out of just one conjunct. And in this case, this follows because a conjunct which is looking to bind is not of the same category and type as one which is not looking to bind, and there is no way to play with the type-shift rules and combinatorics in such a way as to get conjunction here. (Since we have various type-shift rules at our disposal, I unfortunately cannot give an actual proof of this assertion. I can, however, note the following fact which supports this. The essential difference between (24b) (involving binding out of just one conjunct) and cases like (33) (involving binding into just one conjunct) is that in the former case the two conjuncts denote functions which disagree on their range - in the latter case, the two denote functions which disagree on their domain. The Geach rule, *z*, and other familiar type-shift rules, however, manipulate the domains and not the ranges, and so no amount of "playing with" these processes will allow conjoinability of two functions which disagree on their range.)

The conclusion, then, is that not only does the variable-free approach account for the existence of ATB binding with no extra machinery, but it also accounts immediately for the generalization in (27). The relevant expressions will be unconjoinable because something which is looking to bind is not of the same type as something which is not looking to bind. Moreover, none of this needs to be accounted for by positing extra material (either in the syntax or in the semantics) in the position of the gap, and hence this follows completely smoothly under the hypothesis of local interpretation.

8. Nested and Crossed Binding: A Final Problem for a "Parallelism" Constraint

Before concluding, there is one other interesting interaction of binding and conjunction worth considering. By way of background, note first that any theory of binding must account for the fact that both nested and crossed patterns of binding are possible, as in (35a) and (35b):

- (35) a. Every man_i said that no boy_j thinks that he_j likes him_i.
 b. Every man_i said that no boy_j thinks that he_i likes him_j.

As discussed in Jacobson (in preparation), this fact turns out to be no problem for the account proposed here: the different patterns will result from a different order of application of **g** and **z** on *thinks*. Space precludes a detailed demonstration of this fact, but we can informally elucidate the intuition as follows. First, assume that *he likes him* means simply *likes'* - note that its syntactic category is (SNP)^{NP}. (To actually put this sentence together one needs a slight generalization of the **g** operation defined in (11); this generalization is spelled out in Jacobson (in preparation).) In order for *thinks* to take this as argument and bind one (but only one) of the open slots in this complement, it will have to undergo both **g** and **z**. Suppose, then, we first apply **z** and then **g** - that is, the relevant verb here is **g**(**z**(*think'*)). Note that ordinary *think'* is of type $\langle t, \langle e, t \rangle \rangle$. Consider now the meaning **z**(*think'*). This is of type $\langle \langle e_1, t \rangle, \langle e_1, t \rangle \rangle$; I use the subscripts here merely for convenience to indicate the binding relations. Thus the effect of **z** is to introduce a new open slot in the meaning of the complement (syntactically this slot is filled by a pronoun), and to bind this slot to the subject position of *think*. This derived verb then undergoes **g** - this too introduces a new open slot which we will call *e*₂ and it expects this slot to remain open in the result and to be bound higher up. In other words, the result of applying **g** to **z**(*think'*) is to give a verb of type $\langle \langle e_2 \langle e_1, t \rangle, \langle e_2, \langle e_1, t \rangle \rangle \rangle$. Hence the first open argument of *likes* - which happens to be the object slot - will be bound from higher up, while the second open argument (the subject slot) will be bound by the subject of *thinks*. This then results in the meaning in (35a). Were the operations to occur in the reverse order we would get the meaning in (35b). Here **g**(*think'*) yields a function of type $\langle \langle e_1, t \rangle, \langle e_1, \langle e_2, t \rangle \rangle \rangle$. That is, we create a new open slot in the complement, and this slot remains open to be bound later. Application of **z** to this meaning opens up yet another slot in the complement, but it binds that slot to the subject of *think* - hence the resulting function is of type $\langle \langle e_2, \langle e_1, t \rangle \rangle, \langle e_1, \langle e_2, t \rangle \rangle \rangle$. (The reader may have noticed that the **z** rule in (12) will not actually apply in this way here since it requires binding "across" an argument slot; the generalization of **z** referred to directly under (12) will, however, have the desired effect. Moreover, this generalization is independently motivated for the case of 3-place verbs.)

This way of accounting for the difference between nested and crossed binding patterns makes a very interesting prediction with respect to RNR. Notice that **g**(**z**(*think'*)) and **z**(**g**(*think'*)) are of exactly the same semantic type, even though they have different meanings - both are of type $\langle \langle e, \langle e, t \rangle \rangle, \langle e, \langle e, t \rangle \rangle \rangle$. One verb wants to bind nestedly and the other wants to bind crossedly, but this is just a meaning difference rather than a difference in semantic type (or in syntactic category). Thus any two larger expressions which are built from these two verbs should also have the same semantic type, provided, of course, that the two expressions in question are built in a parallel way. The prediction, then, is that in the case of RNR it should be possible to conjoin two expressions - where one wants to bind in a crossed fashion and the other in a nested fashion.

Although the facts are perhaps a tad murky, all informants with whom I have checked (about 8) have reported that indeed the sentences in (36) and (37) are at best awkward and that these are unquestionably better than the cases in (22) and (24) of binding out of just one conjunct:

- (36) a. Every man_i told his son_j but no boy_k wanted to hear his father_m say that he_{i/m} would withhold his_{j/k} allowance.
 b. Every man_i told every boy_j but no boy_k wanted to hear any man_m say that he_{i/m} would withhold his_{j/k} allowance.
- (37) Every man_i thinks that every boy_j should know but no boy_k believes that any man_m would dare say that he_{i/m} will withhold his_{j/k} allowance.

The prediction thus seems to be borne out - although this is a rather striking and perhaps surprising result. But it is exactly as the variable-free account predicts: the fact that one conjunct wants to bind in a nested fashion while the other wants to bind in a crossed fashion has no consequences for their semantic type (nor syntactic category). What is especially striking about this result is that it would be extremely difficult to account for under any theory which tried to rule out (22) and (24) by some sort of parallelism constraint on binding.

9. Locality or variable-free?

I have focussed here on one aspect of a variable-free semantics: it allows for the effect of the pronoun meaning and the effect of binding to be located quite locally, which in turn provides a very smooth and natural analysis of both functional questions (and their answers) and ATB binding in RNR. The reader might wonder, however, whether the key is really tied in crucially with a semantics without variables. Could one adopt a semantics with variables, and still rethink binding in such a way as to mimic the locality predictions here?

The answer is yes, but only by undermining much of the initial motivation for variables. Thus, one could achieve all of the same results in the following way. Let *he* translate as \underline{x} (and hence its meaning is a non-constant function from assignment functions to individuals). Let any expression containing an as-yet unbound pronoun translate in the usual way; *he lost*, for example, has as its meaning $\underline{\text{lost}}(\underline{x})$. However, suppose that one generalizes the rule in (10) in such a way that one can at any point in the semantic composition λ -abstract over an unbound pronoun. Thus *he* can shift so that its meaning is $\lambda x[x]$. Further, assume that we maintain the Geach rule in (11), so that the combinatorics proceed exactly as in the system outlined here. Finally, assume that binding is not accomplished by either the rule in (9) or (10), but rather by the *z* rule given above in (13). As the interested reader can verify, this will allow the same predictions as the system outlined here.

But now we are left with a system in which the variables do no real work. As soon as they are introduced, they are be λ -abstracted over so as to "close" the expression - a pronoun is thus converted from denoting a (non-constant) function from assignment functions to individuals to a meaning which is a constant function from assignment functions to functions of type $\langle e, e \rangle$. But if one always closes off all of the unbound variables as soon as they are introduced, there is no point in maintaining the entire machinery of assignment functions - the meaning of any expression once it combines with others is a constant function from assignment functions. In such a system, in other words, the assignment functions and hence the variables do no work whatsoever; they merely get in the way and require the extra rule to λ -abstract over them. Obviously, such a system is more complex than the variable-free system here. The moral, it appears, is that the locality effects discussed here do indeed provide support for a variable-free semantics.

References

- Chierchia, G. (1988), "Aspects of a Categorical Theory of Binding", in R. Oehrle, E. Bach and D. Wheeler (eds.), *Categorical Grammar and Natural Language Structures*. Dordrecht: Reidel.
- Chierchia, G. (1990). "Anaphora and Attitudes De Se", in R. Bartsch et al. (eds.), *Language in Action*. Dordrecht: Foris.
- Curry, H. and R. Feys (1958). *Combinatory Logic*. Amsterdam: North Holland.
- Dahl, O. (1981). paper presented at the Stanford Sloan Workshop on Non-Transformational Grammars. Stanford University.
- Dowty, D. (1987). "Type Raising, Functional Composition, and Non-Constituent Conjunction", in R. Oehrle, E. Bach, and D. Wheeler (eds.), *Categorical Grammars and Natural Language Structures*, 153-198. Dordrecht: D. Reidel.
- Dowty, D. (1992). "'Variable-Free' Syntax, Variable-Binding Syntax, The Natural Deduction Lambek Calculus, and the Crossover Constraint", in *Proceedings of the 11th Meeting of the West Coast Conference on Formal Linguistics*. Stanford: CSLI Publications.
- Engdahl, E. (1986). *Constituent Questions*. Dordrecht: Reidel.
- Gawron, M. and S. Peters (1990). *Anaphora and Quantification in Situation Semantics*. Stanford: CSLI Publications.
- Gazdar, G. (1980). "A Cross-Categorical Semantics for Coordination", *Linguistics and Philosophy* 3, 407-409.
- Groenendijk, J. and M. Stokhof (1983). "Interrogative Quantifiers and Skolem Functions", in K. Ehlich and H. van Riemsdijk (eds.), *Connectedness in Sentence, Discourse and Text*, Tilburg Studies in Language and Literature 4. Tilburg: Tilburg University.
- Hendriks, H. (1987). "Type change in semantics: the scope of quantification and coordination" in E. Klein and J. van Benthem (eds.), *Categories, Polymorphism, and Unification*, Centre for Cognitive Science, University of Edinburgh / Institute for Language, Logic, and Information, University of Amsterdam.
- Hepple, M. (1990). *The Grammar and Processing of Order and Dependency: A Categorical Approach*. Ph.D. Dissertation, University of Edinburgh.
- Higginbotham, J. (1992). "Reference and Control", in R. Larson et al. (eds.), *Control and Grammatical Theory*. Dordrecht: Kluwer.
- Hohle, T. (1990). "On Reconstruction and Coordination". ms., University of Tübingen.
- Jacobson, P. (1977). *The Syntax of Crossing Coreference Sentences*. Ph.D. Dissertation, UC Berkeley. (published by Garland, Inc., 1981).
- Jacobson, P. (1991). "Functional Questions, Paycheck Pronouns, and Variable Binding", presented at the 1991 LSA/ASL Conference, Linguistic Society of America Summer Institute, Santa Cruz.
- Jacobson, P. (1992a). "Bach-Peters Sentences in a Variable-Free Semantics", in P. Dekker and M. Stokhof (eds.), *Proceedings of the Eighth Amsterdam Colloquium*. Amsterdam, University of Amsterdam ILLC.
- Jacobson, P. (1992b). "Antecedent Contained Deletion in a Variable-Free Semantics", in C. Barker and D. Dowty (eds.), *Proceedings of the Second Conference on Semantics and Linguistic Theory*. Columbus: Ohio State University.

- Jacobson, P. (1994a). "i-within-i Effects in a Variable-Free Semantics and a Categorical Syntax", in P. Dekker et al. (eds.), *Proceedings of the 9th Amsterdam Colloquium*. Amsterdam: University of Amsterdam ILLC.
- Jacobson, P. (1994b). "On Copular Connectivity", in *Proceedings of the Forth Conference on Semantics and Linguistic Theory*. Cornell: Cornell Working Papers.
- Jacobson, P. (in preparation). *Towards a Variable-Free Semantics*.
- Keenan, E. and L. Faltz (1985). *Boolean Semantics for Natural Language*. Dordrecht: D. Reidel.
- Lakoff, G. (1986). "Frame Semantic Control of the Coordinate Structure Constraint", in A. Farkley et al. (eds.), *Papers from the Parasession on Pragmatics and Grammatical Theory*. Chicago: Chicago Linguistics Society.
- Landman, F. and I. Moerdijk (1983). "Compositionality and the Analysis of Anaphora", *Linguistics and Philosophy* 6, 89-114.
- Partee, B. and E. Bach (1981). "Quantification, Pronouns, and VP Anaphora", in J. Groenendijk, T. Janssen, and M. Stokhof (eds.), *Formal Methods in the Study of Language: Proceedings of the Third Amsterdam Colloquium*. Amsterdam: Mathematisch Centrum.
- Partee, B. and M. Rooth (1983). "Generalized Conjunction and Type Ambiguity", in R. Bauerle, C. Schwarze, and A. von Stechow (eds.), *Meaning, Use, and the Interpretation of Language*, 362-383. Berlin: Walter de Gruyter & Co.
- Quine, W.V. (1966). "Variables Explained Away", in W.V. Quine, *Selected Logic Papers*. New York: Random House.
- Reinhart, T. (1990). "Self Representation", presented at the Princeton Conference on Anaphora.
- Sag, I. (1976). *Deletion and Logical Form*. Ph.D. Dissertation, MIT. (Distributed by the Indiana University Linguistics Club, Bloomington).
- Sells, P. (1984). *Syntax and Semantics of Resumptive Pronouns*. Ph.D. Dissertation, University of Massachusetts. (Distributed by Graduate Linguistics Student Association, University of Massachusetts, Amherst.)
- Szabolcsi, A. (1987). "Bound Variables in Syntax: Are There Any?", in J. Groenendijk et al. (eds.), *Proceedings of the 6th Amsterdam Colloquium*. Amsterdam: University of Amsterdam ILLI.
- Szabolcsi, A. (1992). "Combinatory Categorical Grammar and Projection from the Lexicon", in I. Sag and A. Szabolcsi (eds.), *Lexical Matters*. Stanford: CSLI Publications.
- Stechow, A. von (1990). "Layered Traces", ms. University of Konstanz.
- Steedman, M. (1987). "Combinatory Grammars and Parasitic Gaps", *Natural Language and Linguistic Theory* 5, 403-440.
- Steedman, M. (1990). "Gapping as Constituent Coordination", *Linguistics and Philosophy* 13, 207-263.
- Williams, E. (1977). "Discourse and Logical Form", *Linguistic Inquiry* 8.