Direct Compositionality and Variable-Free Semantics: The Case of Binding into Heads

Pauline Jacobson
Brown University

to appear in B. Jackson (ed.), *Proceedings of the Twelfth Conference on Semantics and Linguistic Theory,* Cornell University: CLC Publications (Dept. of Lingusitics), 2002.

1. Theoretical Background

The point of departure for this paper is the hypothesis of "direct compositionality" (see, e.g., Montague 1974): the syntax and the model-theoretic semantics work in tandem. Thus the syntax "builds" (i.e. proves the well-formedness of) expressions, where each syntactic rule supplies the proof of the well-formedness of an output expression in terms of the well-formedness of one or more input expressions. (These rules might, of course, be stated in highly general and schematic terms as in, e.g., Categorial Grammar.) The semantics works in tandem with this - each output expression is directly assigned a meaning (a model-theoretic interpretation) in terms of the meaning(s) of the input expressions(s). There is thus no need to for any kind of abstract level like LF mediating between the surface syntax and the model-theoretic interpretation, and hence no need for an additional set of rules mapping one "level" of syntactic representation into another.

Within this general program, I will be advocating a sub-program: that of Variable-Free semantics (see, e.g., Jacobson, 1994, 1999; 2000). The claim here is that the semantics makes no use of variables and hence no use of assignment functions. The relevance of this to the hypothesis of direct compositionality stems from the observation that many of the arguments for LF are based on the notion that "binding" is a relationship between an actual "binder" (an NP[1]) and a pronoun, trace, and/or variable at some level of representation. It is further assumed that the "binder" and the "bindee" have to be in a particular syntactic configuration with respect to each other at the relevant level. But because we frequently find "binders" and "bindees" which are not in the right configurational relationship on the surface (or at least not at the "pronounced"part of surface structure), it is common to posit an abstract level (or unpronounced pieces) at which they *are* in the right configurational relationship. The variable-free program maintains that this entire view is incorrect. The claim here is that there is no such thing as "binders" and "bindees" - the semantic effect that we think of as "binding" is the result of a rule which merges *argument slots.* Hence "binding" is not a relationship between two actual NPs in some syntactic representation.

This paper is an in-depth study of one such case - the case of the binding of a pronoun in head position from a "binder" within a relative clause. . On the one hand, we will see (Sec. 4) that positing an abstract level to account for the "binding" here is actually of no help. Among other problems, it does not get the semantics right (at least not in any straightforward way). On the other hand, we will see that assigning the correct meaning is straightforward under the variable-free program. To do this, I will propose one new type shift rule above and beyond the apparatus developed in Jacobson (1999), but I will argue that this particular rule is extremely simple and natural.

There is an additional point of interest emerging from this domain: it provides an illuminating comparison between a theory with and without variables. Thus something akin to the proposal here could be implemented in a theory with varibles, and in fact my proposed solution is arguably just a variant of the solution in Engdahl (1986) for this type of problem. Engdahl too adopts a version of direct compositionality, but implements this using variables. Because the two solutions vary essentially in whether or not they adopt variables and assignment functions, this domain highlights a key difference between variable-free and variable-ful. Thus I will argue that Engdahl's version lacks the "naturalness" and simplicity allowed in the variable-free account - this is precisely because of the clutter engendered by assignment functions. As soon as one recasts her basic insight into the variable-free way of thinking, it becomes perfectly natural.

2. Background Pieces

Let us first lay a few background assumptions. Consider the semantic composition of relative clauses in general, as in:

(1) the cat that Mitka chased

I make some minimal and fairly standard assumptions here. One is that the meaning of *(that) Mitka chased* is <u>λx[chased'(x)(m)]</u>. I further assume that the grammar contains some rule(s) which allow *cat* to combine with *that Mitka chased* to give a complex noun (which then occurs with the determiner), whereby the meaning of *cat that Mitka chased* is the intersection of <u>cat'</u> with <u>that-Mitka-chased'</u>. (I use the prime notation here to notate model-theoretic objects - not objects in some intermediate representation. I will, moreover, be referring to complex material *cat that Mitka chased* as a "noun" and will assign it the category N. One can substitute one's favorite category label here..) Further, take a case with two relative clauses like:

(2) the cat that Mitka chased that lives next door

I will assume a standard "stacking" analysis here - whereby *cat that Mitka chased* first combines to give a complex head which then combines with the second relative clause. (Hence <u>cat'</u> intersects with <u>that-Mitka-chased'</u> which result in turn intersects with <u>that-lives-next-door'</u>. See Sec. xx for an alternative possibility.

The next background piece is less standard, and concerns the question of how binding takes place into the post-copular constituent in (3) (such cases were first noted in Geach (19xx) and have played a prominent role in recent literature):

(3) a. The woman who every man_i loves (the most) is his_i mother.
b. The (only) woman that no man_i would forget to invite to his wedding is his_i mother.

Incidentally, the use of indices here and throughout is merely for convenience to indicate the intended reading - in the variable-free view the grammar knows nothing about indices. (I also will omit them when the intended reading is clear). Here I assume the general line taken in von Stechow (1990), Jacobson (1994) and Sharvit (1998) - who all observe that these are a straightforward extension of functional questions as analyzed in Groenedijk and Stokhof (1983) and Engdahl (1986). This basic observation is neutral between the standard and the variable-free views (note, for example, that von Stechow and Sharvit's versions of the analysis are embedded in a standard framework). I have, however, argued elsewhere that the analysis functional phenomena (question and NPs) in general is much simpler and more natural in a variable-free framework. In fact, the existence of functional readings for questions and NPs is completely unsurprising in the variable-free view and so these readings come virtually for free (see Jacobson (1999) for discussion). I will, therefore, elucidate the functional analysis of (3) using the variable-free apparatus.

To save space, I will have to assume some basic familiarity with the apparatus discussed in Jacobson (1999), but will briefly review its applicability to the case at hand. (This is detailed further in Jacobson, 1994.) Take first a relative clause such as *(who) every man loves*. In an ordinary case, this is of type <e,t> - and denotes the set of individuals that every man loves. This result can be derived rather straightforwardly (and in a direct compositional fashion) under, e.g., the analysis of extraction in Steedman (1987) or under any of the other extraction analyses which have been put forth in the Categorial Grammar and related literature. But given the variable-free apparatus, this relative clause also automatically has an additional "functional" meaning, by which I mean a meaning of type <<e,e>,t> (i.e., a set of functions from individuals to individuals). We need nothing new in the system to derive this: the very apparatus used for pronominal "binding" in general automatically gives us this meaning. Thus *loves* can undergo the **z** rule to be of type <<e,e>,<e,t>> with meaning λf[λx[loves'(f(x))(x)]]. The normal conventions will apply to put together the full relative clause, and - under the derivation where *loves* has undergone **z** - the meaning will be λf[every-man'(λy[loves'(f(y))(y)])]. (In other words, this is the set of functions that every man **z**-loves.) This, then, is what happens in (3a). The same basic point holds for the case of functional questions. Notice that functional readings for questions are somewhat unexpected under the standard view of things - the usual line is to assume that a "trace" or "gap" can correspond either to an individual variable or to a variable over functions of type <e,e> applied to a variable over individuals (see, e.g., Groenendijk and Stokhof (1983), Engdahl (1986)). But it follows from nothing else that a gap should be able to have either meaning. In the variable-free system, a "gap" is nothing more than an argument slot which hasn't been filled in the "normal" way. Moreover, the **z** rule is the type-shift rule which accomplishes binding *in general*. (Notice that any theory needs *some* rule to accomplish binding.) Thus the very mechanism for binding in general can apply here so as to give a functional "gap" rather than an individual gap.

The next step in the semantic composition of (3a) is the only piece which does not come for free in the variable-free program. Thus, we need a type-shift rule to map the meaning of the head noun *woman* (whose lexical meaning is a set of individuals) into a set of functions whose range is woman'. This is given in (4);, I leave it open as to the effect this rule has on the syntactic category of the expression:

(4) Let α be an expression of the form: <[α]; N; α'>. Then there is an expression β of the form <[α]; N (??); $\lambda f[\forall x_{\text{in domain of } f}]\, \alpha'(f(x))]$> for f a (partial) function of type <e,a>, where α'is of type <a,t>.

Two comments are in order . First, in most cases of concern f is of type <e,e>, as the input to the rule is a noun with meaning of type <e,t>. However, I have formulated (4) more generally and hence this can apply recursively. This will, for example, allow a noun to shift to have a meaning of type <<e,e>,t>, and then to shift again into a meaning of type <<e,<e,e>,t> etc. (It should perhaps be given even more generally so that the output can be of any type <b,a>, but we will not pursue this here.) Second, one does not want a proliferation of type-shift rules, and it would be nice if (4) followed from something else. As of yet, though, I do not see how to eliminate this (but see Winter, 2001, this volume). But in defense of (4), it seems like a rather natural and unpernicious rule. And note that the non-variable free analogue to the analysis here *also* makes use of this shift (see, e.g., von Stechow (1990), Sharvit (1998)) and so this is not extra to the variable-free analysis.

The final piece in the analysis does comes "for free" in the variable-free program: the apparatus is such that, in general, an expression like *his mother* denotes a function from individuals to individuals (rather than the standard view under which it is a function from assignment functions from individuals). We now have all the pieces to give a smooth, simple, direct compositional analysis for (3a) without reconstruction and without use of any additional apparatus except for the rule in (4). *woman* denotes a set of functions with range woman' and *every man loves* denotes the set of functions f such that every-man **z**-loves f. These two intersect - as in the case of combining any head with a relative clause - and this then occurs as argument of *the.* Hence the pre-copular constituent denotes the unique (contextually salient) function f with range woman' such that every man **z**-loves f. The meaning of the post-copular constituent *his mother* automatically is a function of the right type and in particular is the-mother-of function, and *be* equates these.

The moral, then, is that the apparent "binding" of *his* is an illusion. There is no need to posit any kind of extra level/structure at which *his* is c-commanded by *every man,* nor is there any need for any kind of co-indexation between *every man* and the pronoun. Devices like "binding" and indices are artefacts: the semantics just smoothly works to get the appropriate meaning here.

3. The problem: "binding" into the head in functional questions and functional NPs

That's the good news, but unfortunately not the end of the story. For the appearance of binding is possible not only in the post-copular constituent but also in the head (both in a functional question and in a functional NP) and the story told above says nothing about these:

(5) Which of his_i relatives does every man_i admire the most? (His mother)

(6) a. The relative of his_i that every man_i admires loves the most is his_i mother.
b. The relative of his_i that no man_i would forget to invite to his wedding is his_i mother.

Cases like (5) are discussed extensively in Engdahl (1986), whose analysis I return to later. Notice that in the above examples we get apparent "binding" into the complement of a head noun. Unsurprisingly, there is also binding into a relative clause within a complex head or fronted *wh* phrase:

(7) Which woman that he_i loves did every man_i invite?

(8) a. The woman that he_i loves that every man_i invited was his_i mother.
b. The woman that he_i loves that no man_i will forget to invite is his_i mother.

Before continuing, let us consider just where we get this kind of apparent binding. Jacobson (1999) argues that NPs can have functional readings in general (not just in copular constituents), and that allowing for a functional reading on the object NP in, e.g., (9a) explains the presence of the unexpected - or "sloppy"- inference shown in (9c) and discussed in, among others, Reinhart (1991):

(9) a. John always buys whatever Bill buys.
b. $Bill_i$ buys his_i favorite car.
c. Therefore, $John_j$ buys his_j favorite car.

Leaving aside the full details, (9a) means that John **z**-buys whatever function f Bill **z**-buys. (9b) says that Bill **z**-buys the function mapping each individual into that person's favorite car. Again all of this is automatic: it follows immediately that *buy* can undergo **z** in all of its occurrences in (a) and (b),and it follows immediately that *his favorite car* denotes a function of type <e,e>. From this (c) follows - John **z**-buys the favorite car function also.

It is therefore unsurprising that the unexpected "binding" of pronouns in the head occurs in these functional NPs as well:

(10) John forgot to invite the very relative of his that no other man would have ever forgotten to invite (namely, his mother).

The account to be developed below will straightforwardly extend to this case, as the interested reader will easily be able to verify. As will become clear, my account is tied in to the fact that we are dealing in all of these cases with *functional* NPs.

But there is one complication which I will not address: surprising binding effects are occasionally found in other places. Cases along the lines of (11) have been noted from time to time in the literature; here the NP in question does not obviously have a functional reading (or, if it does, then one needs further conventions to put the meanings all together):

(11) The relative of his_i that every man_i loves the most took his_i picture at the family reunion.

These seem to be a special case of a more general phenomenon discussed in Sharvit (1998), who notes further that the phenomena here is far more restricted than the apparent "binding" into functional NPs. (Sharvit deals primarily with Hebrew and concentrates on the apparent "binding" of the pronoun in object position rather than the pronoun within the head, but her observations extend directly tho this case.) For example, these are not terribly happy with downward entailing binders:

(12) ?*The (very) relative of his_i that no man_i would forget to invite always pays for his_i wedding (namely, his_i mother).

.
(I am not confident that (12) is completely bad, but it is certainly far worse than (8),) Sharvit therefore analyzes this type of case as involving a "pair/list" relative - I will not explore whether the account of binding into heads that I propose here will extend to these cases.

Returning to the better understood functional cases, our task is thus to account for apparent "binding" into the head position in (6) and (8). Before developing my proposal, let me mention three other accounts. The first is that in Engdahl (1986), to which I will return in Sec. xx. The second is an account developed in Winter (2001, this volume). This turns out to be in the same spirit as the account here and our accounts (developed independently) have a striking affinity: they are, in fact simply inverses of each other. I will comment on this below. But the approach which is probably best known is a "reconstruction" analysis - according to which there is a level of representation at which the *wh* phrase in questions and the head of a relative clause is in the position of the gap. Since this is so widely assumed to provide a solution to the puzzle, let me digress to look more closely at this approach.

4. Interlude: Deconstructing reconstruction

4.1. The incorrect semantics

Under a "reconstruction" story, the idea regarding (5)and (7) (the question cases) is to posit a level of representation at which the fronted *wh* material is in the position of the gap, and to assume that the constraints on binding (and/or the interpretation of the sentence) holds for that level. Since these are *wh* questions, this goes hand-in-hand with the standard assumption that the fronted *wh* constituent starts out in the gap position and then moves. Hence the level relevant for binding could be the pre-movement level (the hypothesis that interpretation happens *before* movement was standard in the classical transformational and Generative Semantics literature of the 60' s and 70's); it could instead be a "reconstruction" level at which material is moved and then later put back into the position from which it is moved (as in standard GB); or it could instead be thought of as the actual surface structure under the copy theory of movement (whereby moved material remains in its pre-movement position, albeit in "unpronounced" form). I will henceforth use the term *reconstruction solution* to refer to any of these three views. The explanation for the appearance of binding in a relative clause head as in (6) or (8) is somewhat trickier - the analogous story can be maintained only with the additional assumption that the *head* in a relative clause (as well as the *wh* - word) starts out in the gap position. This view - the Head Raising analysis - is considerably more controversial. But it has been proposed for this general kind of problem off and on since at least as early as Vergnaud (19xx). Thus in (6) the head is *relative of his* and so - under a Head Raising analysis - is in the gap position at the relevant level. In (8x) (assuming a stacking analysis) the head is *woman who he loves* and so this is object of *invite* at the relevant level. (Of course in this case *woman* is also raised from the gap position following *loves*.)

To the extent that there are specific proposals for what it means to interpret the head noun in the position of the gap, I believe that the usual current view is that the lexical content of the head restricts the variable to range over members of the set denoted by the head. In other words, imagine that the "reconstruction" structure for an ordinary relative clause like (13) is as shown in (14a) or (14b). (The choice here depends on whether or not one adopts a copy theory of movement - in which case *man* is in two places - or whether the head is actually moved into (or, out of) the gap position. In either case, though, it will not (or need not) be interpreted in head position, so in (14b) I will put the head in italics, adopting a convention to italicize material which is "invisible" to the semantics).

(13) every man who Mary invited

(14) a. every [who Mary invited $t_{i\,[man]}$] b. every *man* [who Mary invited $t_{i\,[man]}$]

Assume further the following convention for interpreting the trace:

(15) $[[t_{i[N]}]]^g = g(x_i)$ if $g(x_i) \in [[N]]^g$ and undefined otherwise

Given this. *man who Mary invited* in (13) ends up with the same meaning as in the standard view (according to which the semantic composition intersects the man-set with the set of Mary invitees). That is, $[[\text{Mary invited } t_{i\,[man]}]]^g$ is true if Mary invited $g(x_i)$ and $g(x_i)$ is a man. λ-abstracting over the variable in object position will yield something which has the same value on all assignment functions, and in particular will yield (for every assignment function) a function mapping an individual a to true iff Mary invited a and a is a man.

But for the case of an NP like *the relative of his that every man invited* this procedure yields the wrong meaning. The interested reader can work through the details, but in this case *relative of his that every man invited* will denote the set {x|every-man'(λy[x is a relative of y & y invited x]}. *the* maps this into the unique (contextually salient) member of this set and so we get the unique (contextually salient) individual who is a relative of every man and was invited by every man. This, of course, is of no use in (6a) - what we need here is a functional meaning. In other words, this analysis misses the point that this unexpected binding goes hand-in-hand with the fact that these NPs have functional interpretations (modulo the open cases such as (11) which have a far more limited set of binders).

Now one might try to save the recsontruction analysis by adopting reconstruction <u>and</u> combining it with a functional NP analysis. Consider the ordinary functional case like (3a) *The woman who every man loves is his mother*. Combining head reconstruction with a straightforward extension of the Groenendijk and Stokhof/Engdahl analyses of functional questions, one would simply say the following: (i) the trace here is complex (let us index it as $t_{f(i)}$; (ii) this translates as a variable over functions of type <e,e> applied to a variable over individuals - hence, as <u>$f(x_i)$</u> (note: the <u>f</u> variable should also be indexed but I ignore that here); (iii) *woman* is in the gap position, and in this case (iv) the role of the head noun is to restrict the range of the function. Thus the structure for the functional reading of this NP is (16), and the interpretation of the functional trace is as in (17):

(16) every man loves $t_{f(x)\ woman}$

(17) $[[t_{f(i)\ [N]}]^g = g(f)(g(x_i))$ iff $y[g(f(y)) \varepsilon [[N]]^g$ and undefined otherwise

However, this still does not give us the right meaning in the case that the head contains a pronoun which we try to "bind" to a quantified NP in another relative clause. Consider the interpretation of *the relative of his that every man invited*. The reconstructionists hope is that *his* can be an ordinary variable bound in the reconstruction structure by *every man*: the structure is thus:

(xx) the *relative of his* [every man_i invited $t_{f(j)\ [relative\ of\ his(i)]}$]

The interpretation of the trace on an assignment function g is $g(f)(g(x_j)$ provided that the range of the relevant function is the set of relatives of $g(x_i)$. In this case x_i is ultimately bound by *every man.* Thus the entire NP has the following meaning: it denotes the unique (contextually salient) function f such that every man is an x who invited f(x) and the range of f is relatives of x. This means that for all y, f maps y into a <u>relative of x</u> (not, as one might hope, into a relative of y). The only situation in which any function could satisfy that is one where everyone was related to everyone else - clearly not a requirement for truth in (xx). Incidentally, even if we could find some other way to get the functional meaning to come out right, a reconstruction analysis has no obvious way to <u>block</u> the incorrect individual meaning discussed above. We would incorrectly predict, then, that these can occur in ordinary (non-functional) contexts, and that an ordinary subject NP as in (xx) would refer to some unique (contextually salient) individual who is such that every man invited her/him and she/he is a relative of everyone. But (xx) does not have this meaning:

(xx) The relative of his that every man invited walked into the room.

4.2. The questionable logic of the "c-command" restriction

We should, moreover, ask *why* anyone might think about a reconstruction analysis for binding in the first place. (That is, in a theory which does not respect direct compsositionality, one could instead try to raise the binder at LF to scope over the pronoun. The obvious ways to do this actually give the wrong meaning - but, as we saw above, so does reconstruction.) The desire to reconstruct here seems to come from an assumption about the mapping between syntax and semantics: in order for a “binder” to “bind” a pronoun the former must at some relevant level c-command the latter, where this assumption in turn is motivated by Weak Crossover (herearter, WCO) phenomena. In other words, any theory must account for the empirical fact that (xx) is a standard WCO violation like (xx) is not:

(xx) His$_i$ mother loves every man$_i$.

Thus a standard view of WCO assumes a constraint to the effect that a “binder” must c-command a “bindee” at some relevant level of representation (cf., Reinhart, 1983). Of course we need here a definition of “binder” and “bindee”: assume that a binder/bindee pair is any pair of full NP (or, DP - i.e., something whose meaning can be of type <<e,t>,t>) and pronoun which are co-indexed. Assume further that the role of the indices is that there is some level of representation at which the binder is out of the main sentence (e.g., a level derived by QR), but where its surface position is instead occupied by a pronoun or trace with the index that the binder bears at surface structure. The semantic work of co-indexation is done by the assumption that all pronouns with index $\underline{i}$ and all traces with index $\underline{i}$ translate as the variable $\underline{x_i}$. (Notice then that the terms “binder” and “bindee” are a bit misleading - both the position occupied by the “binder” at surface structure and the pronoun(s) it “binds” simply correspond to samely-indexed variables; the actual “binding” in the semantics is accomplished by the fact that these variables are λ-abstracted over and the resulting meaning is taken as argument of the generalized quantifier which we are calling the “binder”.)

Consider now though the following question: what is the level of representation at which the “binder” must c-command the “bindee”? Obviously this cannot be a requirement on LF - the fact that the “binder” must c-command the “bindee” at LF is true by definition (and - given standard assumptions about how LF trees input the semantics - there is no coherent way to give a semantics of “binding” without this.) Because of this fact an LF condition is of no use in accounting for a typical WCO case like (xx) - notice that of course the binder c-commands the bindee in the LF in (xx). The usual assumption, then, is that the requirement is on surface structure.

But of course now we must ask whether (xx) satisfies the WCO requiement under a reconstruction solution. Here it will depend on exactly what version of the latter we adopt. Let us leave aside for the moment the copy theory of movment, and take a standard GB approach whereby the reconstruction level results from moving material back into the position which it occupied before it moved in the syntax. Well then obviously WCO as a constraint on surface structure is still violated in (xx). After all, on the *surface* the head is <u>not</u> in its pre-movement position and so on the surface the binder is not c-commanded by the bindee. The c-command requirement is met only for the “reconstruction level” - but what is that level? We can say it is LF (and say that it is the level that inputs the semantic interpretation), but we have just seen that WCO can’t be a constraint on LF. We would thus have to conclude that WCO is a constraint on some other level - call it level A - a level which is neither surface structure nor LF. Material is put back into the gap position at level A for the purpose of checking WCO violation. But since A is not LF it would appear to do no other work and have no other motivation. In other words, reconstruction is really not helping.

The situation is slightly improved under the copy theory of movement but there still remain open questions. The idea here is that moved material remains in its pre-movement position in surface structure - but is unpronounced there. The "moved" material is actually a copy of the other material. Hence the pronoun which remains in the pre-movement position and is unpronounced is, indeed, c-commanded by its binder at surface structure. But wait - there is another pronoun (the one in the "raised" copy) which still violates the WCO requirement. Surely we can solve this by defining "bindees" as only the unpronounced ones, but why should this be? (I am certain that a defender of the copy theory of movement is thinking here that the reason is obvious: the semantics will be set up in such a way that the one which actually feeds the interpretation is the unpronounced one. But this merely raises two other questions. First, why is this so? Why does the semantics interpret the original and not the copy? Second, we have already seen that LF - i.e., the level which is interpreted - is really irrelevant in the statement of WCO. Under this general view of things, WCO is not a requirement about semantic composition - it is a purely syntactic requirement about co-indexation. The fact that one of the pronouns is "visible" to the interpretation while the other is not is th us really irrelevant.) Thus we can certainly gerry mander things to work, but the contrast between (xx) and (xx) does not follow from any obvious natural and otherwise motivated set of assumptions.

4.3. A new empirical problem for "reconstruction"

There is, moreover, an additional interesting problem with reconstruction. To develop this, note first that the binding works just as well if the two clauses are reversed (I thank Chris Barker for this observation):

(xx) a. The (only) woman that he loves that no man would fail to invite is his mother.
b. The (only) woman that no man would fail to invite that he loves is his mother.

The reconstruction solution discussed above accounts for (a) by positing that a level of representation in which *that he loves* is in the position of gap following *invite.* The result of this is that *he* is c-commanded by *no man* and can hence be unproblematically bound by *no man*. But what about (b)? Notice that - assuming a stacking structure for (xxb) on the surface, *no man* does not c-command *he.* But it does not do so at the reconstruction level either: here *woman that no man loves* would be in the gap position, and would still fail to c-command *he.* (In fact, it would be c-commanded by *he* - suggesting that one might actually expect a Strong Crossover violation, but there is none.)

The problem is a bit more complex, however, for there is a solution available under the general reconstruction approach. This derives from noticing that any time we have two relative clauses - such as a simple case like (xx) - there are conceivably two different structures:

(xx) every man who Bill likes who Mary invited

The first is the stacking structure which we have been dealing with throughout. But - at least given a certain set of assumptions - it may be that (xx) also has an extraposition structure - whereby *who Mary invited* is extraposed modifies the first relative pronoun *who* and is extraposed from this or, similarly, is extraposed from the position of the gap. Whether or not this structure is possible depends on a variety of other assumptions, but it is certainly not out of the question to think that (xx) does have one such analysis. (For detailed discussion, see Jacobson (1982) who in fact proposes that the extraposition analysis is the *only* available analysis for

(xx) and that there is no such thing as stacking. If this is correct then the basic point in this section will continue to hold.) This means that we can account for the apparent binding of the pronoun by *no man* in (xxb) quite simply. If (xxb) can have an extraposition structure - whereby *that he loves* is extraposed from the gap position (the position which is the object of *invite)* then it is c-commanded by its binder at least at the reconstruction level. In other words, since (xx) conceivably has both a stacking and an extraposition structure, we can account for both binding patterns. (xxa) is an instance of stacking, while (xxb) is an instance of extraposition.

But there is a rub. By way of background, Engdahl (1986) notices that - with respect to the analogous case of funcitonal questions - there can be any number of pronouns with any number of binders. The same holds true for the relative clause cases. Thus consider a context in which all of the phonology professors are women, all of the students are men, and each student takes a phonology course from each professor. In this context, we can have a good sentence like (xx) with two binders and two pronouns:

(xx) The assignment that every student most wanted every phonology professor to like was the last one he handed in to her.

Here we simply have two binders in the pre-copular constituent and two pronouns in the post-copular constituent. But we can expand on this example to create the situation of binding into a head. Such a case would be (xx), which is analogous to our original case in (xx) but where we have two bound pronouns within the head. Similarly, we can have the reverse binding pattern, where the relative clause containing the pronouns follows the one with the “binders” as in (xx):

(xx) The assignment that he gave her that every phonology professor most praised every student for was the last one he handed in to her.

(xx) The assignment that every student gave every phonology professor that she most praised him for was the last one he handed in to her.

So far these cases tell us nothing new. Under the reconstruction approach, we can posit that (xx) is an instance of stacking (hence both pronouns are c-commanded by the binders at the reconstruction - assuming, as is independently needed - that a direct object c-commands into a PP). (xx), on the other hand, would involve an extraposition structure; at the reconstruction level *that she most praised him for* is in the position of the gap phollowing *professor* and again all is well.

The trick, though, is that we can also mix and match the binding patterns, as in (xx):[2]

(xx) The assignment that every student gave her that every phonology professor most praised him for was the last one he handed in to her.

To account for the binding of *her* using a reconstruction solution, we need to assume that this is a case of stacking so that *assignment that every student gave her* is in the position of the gap following *for*. This way, *her* is c-commanded by *every phonology professor* at the relevant level. But this leaves *him* unbound. In order to successfully bind *him,* we need to assume that instead this is an extraposition case - whereby the second relative clause is extraposed from the gap position in the first relative clause (and hence *him* is c-commanded by *every student).* But then *her* is not in the right place to be bound. In other words, there is no obvious way to construct some representation in which both pronouns in (xx) are c-commanded by their would-be binders.

5. The proposed solution under Direct Compositionality and Variable-Free Semantics

Recall how the apparent "binding" of the pronoun worked in a simple case like (xx) under direct compositionality and variable-free semantics:

(xx) The (only) woman who no man invited was his mother.

Here the head *woman* shifts to denote a set of functions with range woman'. The meaning of the relative clause *who no man invited* can be put together as follows: every-man' o **z**(invite'), which meaning can be represented equivalently as: λf[every-man'(**z**(invite'(f)]] (hence this is also a set of functions). These two sets can then intersect, just as in the case of the semantic composition of a normal relative clause where the set denoted by the head intersects with the set denoted by the relative clause. But recall further the problem for a case like (xxa) or (xxb):

(xx) a. The (only) relative of his that no man invited was his mother.
b. The (only) woman who he loves that no man invited was his mother.

In the case of (xxa), for example, *relative of his* does not denote a set of functions, but - under the variable-free view taken here - a two-place relation. (Thus *relative of Bill's* is of type <e,t> and so *relative of his* must be of type <e,<e,t>>.) The same is true for *who he loves*. Given that a normal relative clause is of type <e,t>, a relative clause like this one which contains a pronoun is of type <e,<e,t>> under the assumptions laid out in Sec. (xx).

Before continuing, a brief comment is in order concerning the actual meaning of *who he loves* under the variable-free view. The particular 2-place relation that this denotes depends on just what fine-grained assumptions one makes about how the meaning of relative clauses are put together. Under one view, the meaning for this turns out to be λx[λy[loves'(x)(y)]] or, more simply, loves'. This is essentially the result that one gets using a Steedman-style approach to extraction combined with the conventions for pronouns developed in Jacobson (1999; see that paper for discussion of this point). To clarify, let A|B be any expression A with a B-type gap. (Of course in Steedman (1987) there is no distinguished kind of "slash' for an extraction gap, and this would simply be an expression of category $A/_R B$. I will, however, continue to use A|B as a notation for something with an extraction gap of category B so as to be able to phrase the discussion in more general terms - since there are a number of proposals for extraction within CG which *do* distinguish extraction "slashes" from the ordinary "slash".) Now if the conventions for extraction and for the passing of the pronoun-containing information are such that the final category ends up being S^{NP}|NP then the meaning shown above will be the meaning for *who he loves.* One can convince oneself of this by noting that under this view the extraction gap is the expected first argument of the function and the slot occupied by the pronoun is the expected second argument. Thus the argument structure for gap in object - pronoun in subject is exactly the same as for the ordinary transitive verb *love.* There are, however, other accounts of extraction with in CG (see, e.g., Jacobson 1987, Moortgat, 1987, Oehrle 1991), which are such that interacting these with the pronoun conventions of Jacobson (1999) y ields a final category $(S|NP)^{NP}$. Here the argument slot occupied by the pronoun (the subject slot) is the expected first argument in while the extraction gap here (and in all other cases) is *the innermost argument slot.* Under this view, then, the meaning of *who he loves* will be the reverse - i.e., <u>$\lambda x[\lambda y[\text{loves'}(y)(x)]]$</u>. As will be documented below, this is exactly what we need the meaning to be (the reason will emerge below) - thus we would hope that the second class of views on extraction are the right one rather than exactly a Steedman-style view. But this turns out to be an extremely fortuitous result. The interaction of extraction and pronouns is discussed in detail in Jacobson (1999, fn.19) and - for totally independent reasons - I came to the same conclusion there. (and also in Jacobson, 1989 - aborted appendix)

Returning to the main theme, we are trying to come up with a story regarding the semantic composition of "the woman who he loves that every man invited". We have seen that *who he loves* denotes a 2-place relation, while *woman* and *that every man invited* can both denote sets uf functions, where these two sets will intersect. But how can we fold a 2-place relation into the semantic composition? It appears that we need a new combinatory rule or a new type-shift rule in order to do this.

The solution that I wish to propose does in fact make use of a new type-shift rule above and beyond the other mechanisms proposed in Jacobson (1999, 2000) and discussed in Sec. xx (the **g** rule and the **z** ru le). But, it will turn out that this type shift rule is both simple and extremely natural. Consider the case in (xx):

(xx) The relative of his that no man invited was his mother.

Here we want *relative of his* to denote the set of functions <u>$\lambda f[\ \ x[\text{relative-of'}(x)(f(x))]]$</u> (this is the set of functions which map each individual into a relative of theirs). *relative of his* denotes the 2-place relation mapping any x and y to true iff x is a relative of y. But note that if we were to shift this into a set of functions in the most simplistic and minimal way , the above is exactly what we would get. To see this, take any function R of t ype <e,<e,t.>> and consider the "no-brainer" way to map this to a set of functions. This will be to first de-Curry it so that it is a set of ordered pairs of individuals (there are two ways to do this depending on which argument we chose to be the first member and which the second; while I know of no particular reason to choose one over the other it will turn o ut that we want the "subject" to be the second member of the ordered pair and the object to be the first member). Now collect up all the subsets of this set which are functions. What we have is exactly the necessary set of functions to give the meaning above.

Formally , then, assume that we have an operation which we will call **m** which is defined as follows:

(xx) Let F be a function of type <b,<a,t>>. Then **m**(F) is a function of type <<b,a>,t> such that **m**(F) = $\lambda h_{<b,a>}[\ \ x_{\text{in the domain of h}}\ [F(x)(h(x))]]$, where h is a partial function from b to a.

Note that (xx) may look perfectly arbitrary, but it is nothing more than the "natural" mapping from 2-place relations to sets of functions defined above.
As mentioned above, we could as well have de-Curry'ed in the reverse way, which is to say that the **m** operation could just as naturally have been stated as in (xx):

(xx) Let F be a function of type <b,<a,t>>. Then **m**(F) is a function of type <<b,a>,t> such that **n**(F) = $\lambda h_{<a.b>}[\ \ x_b[F(h(x))(x)]]$

The evidence that (xx) is the correct version and not (xx) comes primarily from the pre-shifted meaning of *relative of his* - we know that this denotes the 2-place relation λx[λy[relative-of'(x)(y)]] which is why would need to choose the version in (xx) rather than (xx). It is this fact in turn which leads to the conclusion discussed above that *who he loves* must have the meaning λx[λy[loves'(y)(x)]] and not loves', which in turn means that its syntactic category is $(S|NP)^{NP}$ and not $S^{NP}|NP$. As noted above, this is a very happy result in view of the independent evidence for this discussed in Jacobson (1999). On the other hand, I confess to having no explanation for why it is that (xx) is the correct rule and not (xx) - the two are, as far as I can see, equally natural. To complete the account, this means that we have a ty pe-shift rule as given in (xx); note that I am oversimplifying in that I ignore the syntax connected to this rule. The reason is that there are some vexing issues concerning the syntactic categories vis-a-vis the head noun; since these are actually somewhat vexing I leave them for another day :

(xx) Let α be an expression of the form <[α]; ... ; α'>, for α' a function of type <a,<b,t>>. Then there is an expression β of the form <[α]; ... ; **m**(α')> (syntax is ignored here)

With this apparatus, a head noun such as *relative of his* can shift to denote the set of functions λf[x[relative-o f'(x)(f(x)]] while a relative clause like *who he loves* will shift from λx[λy[loves'(y)(x)]] to the set of functions λf[x[loves'(f(x))(x)]]. There is, therefore, now no problem with giving the semantic composition for an NP like *the woman who he loves that every man invites. woman* denotes the set of functions whose range is the woman set; *who he loves* denotes the set of functions mapping each member of its domain into someone who they love, and *that eveyr man inv*ites is the set of functions f such that every man z-invites f. These three sets can happily intersect. Notice too that exactly the same thing will hold if the relative clauses are reversed as in *the woman that every man invited that he loves.* Whether or not there is an extraposition structure for this is really irrelevant here - as long as stacking is possible this surely can have a stacking structure. And now there is no problem in giving a meaning for the stacking structure - again the three sets intersect. The order in which we do the stacking makes no difference. Thus the real "moral" here is that there is actually no "binding" relationship between *every man* and *he.* (Of course this is true in general in the variable-free semantics; "binding" really plays no rule in the grammar.)

Precedents/ simi larities: Engdahl, Winter

xx. x Interaction with Weak Crossover

Notice that Weak Crossover pattern shows up in one of the relative clauses but not in the other:

(xx) a. The (only) woman who he loves that no man invited was his mother.
b. *The (only) woman who he loves that invited no man was his mother.

(xx) a. The (only) woman who he loves that no man invited was his mother.
b. The (only) woman who loves him that no man invited was his mother.

This follows straightforwardly under the present analysis. Consider first the contrast in (xx). As noted in Jacobson (1994), this is simply analogous to the parallel case for functional questions, whose connection to WCO was first pointed out in Engdahl (1988):

(xx) a. Who does every man love? His mother.
b. *Who loves every man? His mother.

Engdahl (1988) observed that these reduce to WCO effects given her analysis of functional questions (see also Chierchia, 1991). Of course the precise details depend on the precise mechanics used for functional gaps in general, but Engdahl's idea was, roughly that in a functional question there is a gap translating roughly as <u>f(x)</u>. In (xxa) the variable <u>x</u> can be bound by the subject *every man* in the normal way that binding takes place, while in (xxb) the variable could not be bound by the object for whatever precise reason accounts for WCO effects in general. Of course under variable-free semantics the story needs to be translated into a different language, but the same basic story goes through without a hitch. As discussed in much greater detail in Jacobson (1994, 1999), the explanation for run-of-the-mill WCO effects is that the "binding" rule **z** is formulated so as to only "merge" a pronoun within one argument to a later (or, "higher") argument slot. Hence, for example, a pronoun within an object can be "merged" with the subject slot. But there is no backwards rule **s** whose effect would be to merge a pronoun within one argument to a lower (or, earlier) argument slot. (Hence there is no way to merge a pronoun in subject position with the object argument slot.) Since functional readings are simply the result of application of **z** on the verb, the contrast between (xxa) and (xxb) is expected. In (xxa) *invite* has undergone **z**, with the effect that the relative clause is the set of functions that every man z-invites. (xxb) does not allow for an analogous functional reading for the relative clause, since this would require **s** on invite (where the relative clause would then denote the set of functions f such that f s-invited every man).

But in the case of (xx) - where we do not find the WCO asymmetry - the relationship between *he* and the gap is not given by the **z** rule but rather by **m**, where **m** has no difficulty operating in either of these cases.

(xx) a. (=44a) *who he loves;* $(S/_R NP)^{NP}$; λx[λy[loves'(y)(x)]] $\text{--->}_{\mathbf{m}}$
λf[z[loves'(f(z))(z)]]
b. (=44b) *who loves him;* $(S/_L NP)^{NP}$; loves' $\text{--->}_{\mathbf{m}}$
λf[z[loves'(z)(f(z))]]

Apropos WCO, we saw earlier that it was not trivial under a reconstruction approach to account for the contrast between a standard bad WCO case like (xx) and the goodness of (xx). In neither case does the binder c-command the (pronounced) bindee at surface structure and in both cases it does in LF, so we need some way to distinguish these (either by positing an additional level of significance or by appeal to an unpronounced bindee in one case). But under the analysis here there are no new twists which need accounting for. A standard WCO violation like (xx) is bad because it would involve illegitimate use of a type-shift rule **s**. (xx) is fine because it involves **m**. In neither case is there any kind of notion of a binder or bindee as far as the grammar is concerned (nor any co-indexation), but the apparent "binding effect" results from two different type-shift processes.

xx. Comparison to Engdahl

As it turns out, the solution proposed here has some striking similarities to the proposal (for the analogous case of functional questions) developed in Engdahl (1986). This is not obvious at first glance because her exact implementation had wrapped into it certain facts which are particular to questions. Nonetheless, if we tease out the irrelevant differences between Engdahl's proposal and the proposal here, we can recast the essence of her proposal by positing the following rule:

(xx) $P \dashrightarrow \lambda f[\ x[P(f(x))]]$ (for P of type <e,t>)

The intent of x here is that this is not a variable over individuals but, rather, a variable over variables (any variable can be chosen). Indeed, the "trick" here will be that the variable name plays a crucial role, and the rule is deliberately set up in such a way as to "capture" an unbound variable within the input - and it is exactly this fact which will allow for an instructive comparison with the variable-free analogue. In other words, in the variable-free system a phrase like *relative of his* denotes a two-place relation between individuals and shifts by **m** to denote a set of functions. In a system with variables, *relative of his* is not a function of type <e,<e,t>>. Rather it is - relative to an assignment function - a function of type <e,t>. Let us suppose its meaning is what we will represent as relative-of'(x_i). Then if x_i is the variable chosen in an applicaton of P, this will map into $\lambda f[\ x_i[P(f(x_i))]]$. Relative to some assignment function, this is the set of functions f such that everyone is a relative of f(x). Since all the variables are bound here this will actually have the same value on all assignment functions

At first glance, this looks even nicer than the mechanisms which we have set up in the variable-free system - for (xx) kills two birds with one stone. Thus note that this looks just like the rule given in (xx) in the variable-free system - where (xx) is the rule that allows the meaning of a head noun like *woman* to shift in such a way that it denotes a set of functions whose range is the woman-set. Hence the Engdahl-inspired variant of the proposals here need only one rule where the variable-free version needed two; (xx) will do double-duty for both (xx) and for **m**.

But this collapsing is at a heavy cost. This becomes clear only once we consider just what is the actual status of (xx). Suppose, for example, we take this to be a rule which actually operates on representations which applies on the way to forming an LF which receives an interpretation. "P' here then would mean any chunk of representation of a certain form, and this would map into the output representation. In this case, though, there is surely no sense in which any mapping like this is particularly "natural" - if we are mapping uninterpreted representations into uninterpreted representations then any arbitrary string of symbols in the output would be as "natural" as any other.

Of course this is not what Engdahl meant by her analogous rule. (Again, recall that her rule is not exactly the rule given in (xx), but the differences are orthogonal to the points here.) Indeed Engdahl meant this is a mapping from model-theoretic objects to model-theoretic objects, exactly as in the case of the type-shift rules invoked here. But what sort of mapping is this? In the variable-free setting, a rule like (xx) actually maps a set of individuals into a set of functions. But here this is not the case. The input and output of the rule are both functions from G (the set of assignment functions) to something else. Given this, we need to ask whether the particular mapping is "natural" in the sense that (xx) and (xx) are?

Consider first the case in which we apply the mapping to *woman'*. xxx can ignore the assignment functions ... and think of

But this does the work we need for the case of (xx) only by noticing that the assignment functions are crucial - we cannot just "strip them away" and pretend that we are shifting sets of individuals into functons from individuals to individuals. To see the actual model-theoretic content of the rule represented in xx , take that subset of G which agrees on assignments to all variables except x and call that subset H. The input to the rule is a (possibly non-constant) function from H to sets of individuals, while the output is a constant function from h to sets of individuals. For any g in H, the input is some set of individuals, and the output is the set of fucntions f such that for all assignment functions g' in H, the set of individuals that the input expression assigns to g' includes the set of individuals that f assigns to the value of x on g'. I see no way to see this as any kind of "natural" shift xxx

xx. Generalizing to the Engdahl n-place cases

The apparent death knell for the reconstruction analysis discussed above was (xx) where there was a mix-and-match binding pattern. The variable-free solution proposed above cannot, therefore, claim total victory until it is shown that it has a way to account for this. This, then, will be the focus of this section. To give the bottom-line first: it is only fair to note that indeed a tiny bit will need to be added to the account above. In particular, we will (for rather subtle reasons) need to generalize **m** slightly. But once this is done there is no problem - hence the basic line extends in a reasonably straightforward way to account for (xx).

Before turning to the full-blown mix and match case, let's first consider how the acco unt handles simpler cases where there are two (or more) pronouns within one of the relative clauses, and two or more quantifed NPs within the other, as in:

(xx) The assignment that he gave her that every student most hated every professor for was the last one he gave her.

We have three tasks here: (i) the head must shift into a set of functions from two individuals to individuals (i.e., to an object of type <<<e,<e,e>,>,t> ; (ii) the relative clause *every student most hated every professor for* must be able to compose in such a way as to yield an object whose meaning is of this type; and (iii) similarly for the relative clause *that he gave her.*

Beginning with (ii), this is shown already in Jacobson (1999). the asic idea is simply to let *hate (for)* undergo two successive applications of **z** - and where one is a version of **z** which allows the subject argument slot to "bind" while one allows the direct object argument slot to "bind". In an unshifted case, the gap following *for* expects to be an individual argument - an argument of type e. The first shift turns this into an expected function of type <e,e> , and one further shift by **z** will turn this into a gap o f type <e,<e,e>>. The final verb, then, is of type <<<e,<e,e>>,<e,<e,t>>> (i.e., first give me the fancy function from two individuals to another individual, then give me the object and then the subject, and I'll give a truth value). The general rules for relative clauses plus for quantifiers in object psotion will allow (xx) to compose up in s uch a way that the "gap" argument keeps getting held off, and so the final result if of type <<e,<e,e>>,t>.

There is one important point to note before leaving this case. I said above that the trick is to have two applications of **z** - and that when **z** operates on a 2-place verb, it can apply so as to allow binding either by a subject slot or an object slot. (This follows from a generalized version of **z** which is spelled out in Jacobson (1999) and which is needed for the case of 3-place verbs in general.) The two different versions of this operation can happen in either order - and this will in turn give the result that there are actually two different meanings which can be associated with this relative clause. To clarify, let me notate the type of the "gap" as <e_1<e_2,e>>. In one meaning, the subject slot (*every student)* binds the e_1

position while the object slot (*every professor)* binds the e_2 position, the other meaning has the reversed binding pattern. (Actually, there are two other possible meanings with this same type - both argument positions can be bound by the subject slot or both by the object slot. This is exactly what we need for cases where, for example, there is an actual object which contains two pronouns bound by the same thing. But cases of this kind do not concern us here.)

The next task is (i) above - to show that the head can shift to denote a set of functions from two individuals to an individual (in this case, the relevant function when all is put together ends up being a function from professors and students to assignments). But this simply follows from recursive application of the type shift rule given in (xx) [[[make sure this rule is given above]]:

(xx) assignment_1 of type <e,t> --->
assignment_2 of type <ee,t>: $\lambda f[\ \ x[\text{assignment}_1'(f(x))]]$ --->
assignment_3 of type <<e,ee>,t>:

$$\lambda F_{<e,ee>}[\ \ y[\lambda f[\ \ x[\text{assignment}_1'(f(x))]](F(y))]] =$$
$$\lambda F_{<e,ee>}[\ \ y[\ \ x[\text{assignment}_1'(F(y)(x))]]]$$

Thus tasks (i) and (ii) required no new apparatus in the system. In order to complete task (iii) (assigning the double functional reading to *that he gave to her)* we will indeed need to generalize the rule **m**, but in a way which is perfectly obvious and unsurprising. We simply need to reformulate **m** so that it can apply recursively, as given in (xx):

(xx) a. Let F be a function of type <b,<a,t>>. Then **m**(F) is a function of type <<b,a>,t> such that **m**(F) = $\lambda h_{<b,a>}[\ \ x_{\text{in the domain of } h}[F(x)(h(x))]]$
b. Let F be a function of type <c,<b,<a,t>>>. Then **m**(F) is a function of type <c, <<b,a>,t>> such that **m**(F) = $\lambda C_c[\mathbf{m}(F(C))]$

There is nothing unexpected about this generalization. First note that it is exactly analogous to the recursive definition of **g** which is needed independently (see Jacobson, 1999, p. 138 for the recursive formulation and discussion of **g**.) A generalization of this type simply allows us to "hold off" outermost argument slots and perform operations on the inerior portions of complex functions. Moreover, Barker (personal communication) notes that this is actually nothing more than "Geaching" the operations (thus (xxb) is actually **g(m)**.)

Given this generalization, we can obtain the appropriate meaning for *that he gave to her* as shown in (xx). Incidentally, the analysis shown here assumes that the argument structure for *give* is such that it first combines with the second object (which in this case is a "gap" which corresponds to the head noun *assignment)*, then with the direct object (*her)* and finally with the subject. In other words, it is assumed here that in a full version such as *The student gave the professor an assignment,* I assume that *an assignment* is introduced first, and that *the professor* then 'wraps" in (see Bach, 1979, 1980; Dowty, 1982; Jacobson (1986). This, however, is not crucial - everything will work out just as well if the post-verbal arguments are introduced in the opposite order. All that is crucial is the assumption which was discussed in Sec. xx - the conventions for forming relative clauses are such that the gap slot in (xx) ends up as the innermost argument position of the entire relative clause - this means that the ultimate meaning (of type <<e,e>,e> is happy when combining with the head - which is a function from two individuals to assignments:

(49) *that he gave her:* $((S/_R NP)^{NP})^{NP}$; $\lambda x[\lambda y[\lambda z[gave(z)(y)(x)]]]$

x: is ultimate subject slot (students)

$\text{--->}_{\textbf{generalized m}}$ $\lambda s[\mathbf{m}(\lambda x[\lambda y[\lambda z[gave(z)(y)(x)]]](s))] =$

$\lambda s[\mathbf{m}(\lambda y[\lambda z[gave(z)(y)(s)]])] =$

$\lambda s\,[\,\lambda f[\quad x[\lambda y[\lambda z[gave(z)(y)(s)]](x)(f(x))]]] =$

$\lambda s[\,\lambda f\,[\quad x[\,\lambda z\,[gave(z)(x)(s)](f(x))]]] =$

$\lambda s[\,\lambda f\,[\quad x[\,gave\,(f(x))(x)(s)]]]$

hence: first step here "holds off" on the subject (students) slot and performs **m** on the interior; maps $<e_{students}, <e_{professors}, <e_{assignments}, t>>>$ to $<e_{students}, <<e_{profs}, e_{assignments}>,t>>$
(note: subscripts here not part of any restriction on the functions; these are just put in for exposition)

$\text{--->}_{\mathbf{m}}$ will map the above into a set of functions from individuals (students) to functions from individuals (professors) to individuals (assignment)

i.e., to type $<<e_{students}, <e_{profs}, e_{assignments}>>,t>$

$\lambda G_{<e,ee>}[\quad y[above(y)(G(y))]] =$

$\lambda G[\quad y[\lambda s[\,\lambda f\,[\quad x[\,gave\,(f(x))(x)(s)]]]](y)(G(y))]] =$

$\lambda G[\quad y[\lambda f\,[\quad x[\,gave\,(f(x))(x)(y)]]\,(G(y))]] =$

$\lambda G[\quad y[\quad x[\,gave\,(G(y)(x))(x)(y)])\,]]$

this is the set of functions G from two individuals into individuals (assignments)
such that for all y and x, y gave x G(y)(x)

hence, now all of them intersect

Thus the composition of cases with n-place functions of the sort discussed in Engdahl (1986) for the case of functional questions presents no problem for the approach here. But our real mission is not just to show that (xx) can be derived, but to show that the analysis has no difficulty with the "mix and match" case in (xx), since that was the case which was especially problematic for reconstruction. Our new concern, then, is to show that the apparatus applies to a relative clause like *that every student gave her* so as to also map this into a function of type <<e,<e,e>,t>>. To give an overview before proceeding: initially this looks straightforward and requires no new apparatus. But there is a subtle catch: in merging everything together in the right way for a case like (xx), and to solve this problem we will need an additional generalization of the **m** rule. But again this generalization is fairly straightforward.

So lets first look at what happens in the composition of *that every student gave her*. We can compose this up in such a way as to have a meaning of the right type without invoking any additional apparatus. The basic idea is that *gave* undergoes **z** so that expected argument (the gap) is of type <e,e> . Moreover, this inovlves the generalized version of **z** discussed in Jacobson (1999) which allows the subject argument position of *give* to be merged with the argument slot of the gap function. This is just what happens in the case a simple functional relative clause. - such as the underlined part of *the assignment* *<u>that every student gave</u>*

<u>Professor Jones</u> (was the one he had worked on all night). This relative clause will be of type <<e,e>,t> - it denotes a set of functions. Now in the case at hand, the direct object slot is instead a pronoun and so its ultimate type is <e,<<e,e>,t>>, Once again it is worth recalling that the argument structure of a relative clause which also contains a pronoun is such that the gap corresponds to the innermost argument position and the pronoun slot to the outermost argument position - hence the type shown above. At this point, one final operation happens - which is that this meaning shifts by **m** so that the entire relative clause is of type <<e,<e,e>>,t>. This is no different than what we saw in the simple case of a relative clause like *who he loves* as in (xx) - we shifted there from a 2-place relation between individuals to a set of functions from individuals to idnividuals. Here we just shift from a 2-place relation between individuals and functions (of type <e,e>) to a set of functions from individuals to the fancier type (functions from individuals to individuals). The full details, then, are spelled out in (xx):

(xx) *that every student gave her __;* $(S/_R NP^{NP})^{NP}$; $\lambda x[\lambda f_{<e,e>}$[every-student'(**z***-gave(f)(x))]] =

$\lambda x[\lambda f_{<e,e>}$[every-student'($\lambda$y[gave(f(y))(x)(y)])]]

notes: x here fills the direct object slot

f is the "functional gap" - and is the result of **z** on <u>give'</u>

z* above is used to notate that this involves a generalization of the

z operation, to allow binding "across" an argument position - in this case, binding from the subject-argument slot into the argument slot of the function which is the "second object" slot

NOTE: this generalization discussed in, e.g., Jacobson (1999) - needed in general for the case of 3-place verbs

$\text{--->}_{\mathbf{m}}$ $\lambda G_{<e,ee>}$[z[above (z)(G(z))]] =

$\lambda G_{<e,ee>}$[z[λx[λf [every-student'(λy[gave(f(y))(x)(y)])]] (z)(G(z))]] =

$\lambda G_{<e,ee>}$[z[λf [every-student'(λy[gave(f(y))(z)(y)])](G(z))]] =

$\lambda G_{<e,ee>}$[z[every-student'(λy[gave(G(z)(y))(z)(y)])]]

Now let us consider the composition of the second relative clause *that every professor most praised him for.* One might think we are home free - we compose this in a fashion exactly analogous to that shown in (xx). Indeed, this is possible, but it will not give us what we want here. What we need the final meaning to have the expected arguments of the function in reverse order; that is the final meaning must come out as xzzz

sxxxx explain why -

one wants to bind in one order, and the other in the other

This is reminiscent of a case like xxx where the solution was a result of different orders of application of z and me. Hence, need same thing here ...

general8ize **m**

xx. Som e potential worries

is the generalize version still “natural”?
what will be the cost of the additiion of **m**? Need to fill in the syntax

why are these cases no good with two pronouns - too partial?

References

Cooper, R. (1979). "The Interpretation of Pronouns", in F. Heny and H. Schnelle (eds.), *Syntax and Semantics 10: Selections from the Third Groningen Round Table,* 61-92. Academic Press, New York.

Dowty, D. (1999). "Natural Language Anaphora and Type Logical Syntax", paper presented at the *Formal Grammar 1999* conference, Utrecht.

Engdahl, E. (1986). *Constituent Questions.* Reidel, Dordrecht.

Geach, P. (1972). "A Program for Syntax", in D. Davidson and G. Harman (eds.), *Semantics of Natural Language,* 483-497. Reidel, Dordrecht.

Hepple, M. (1990). *The Grammar of Order and Dependency: A Categorial Approach.* Ph.D. Dissertation, University of Edinburgh.

Heim, I. (1990). "E-type Pronouns and Donkey Anaphora", *Linguistics and Philosophy 13,* 137-178.

Jacobson, P. (1977). *The Syntax of Crossing Coreference Sentences.* Ph.D. Dissertation, University of California at Berkeley. Published by Garland Press, 1980, New York.

Jacobson, P. (1991). "Bach-Peters Sentences in a Variable-Free Semantics", in P. Dekker and M. Stokhof (eds.), *Proceedings of the Eighth Amsterdam Colloquium.* University of Amsterdam ILLC, Amsterdam.

Jacobson, P. (1999). "Towards a Variable-Free Semantics", *Linguistics and Philosophy* ***22***, 117-184.

Jacobson, P. (to appear). "Paycheck Pronouns, Bach-Peters Sentences, and Variable Free Semantics", *Natural Language Semantics.*

Karttunen, L. (1969). "Pronouns and Variables" in R. Binnick et al. (eds.), *Papers from the Fifth Regional Meeting of the Chicago Linguistic Society,* 108-116. Chicago Linguistic Society, Chicago.

Montague, R. (1974). "The Proper Treatment of Quantification in Ordinary English", in R. Thomason (ed.), *Formal Philosophy: Selected Papers of Richard Montague*, 247-278. Yale University Press, New Haven.

Partee, B. (1975). "Deletion and Variable Binding", in E. Keenan (ed.), *Formal Semantics of Natural Language.* Cambridge University Press, Cambridge.

Rooth, M. (1985). *Association with Focus*. Ph.D. Dissertation, University of Massachusetts, Amherst.

Ross, J. (1967). *Constraints on Variables in Syntax*. Ph.D. Dissertation, MIT, Cambridge.

Sauerland, U. (1998). *The Meaning of Chains.* Ph.D. Dissertation, MIT, Cambridge.

Sauerland, U. (1999). "Why Variables?", in *Proceedings of the North East Linguistic Society 29,* 323-337. Distributed by University of Massachusetts GLSA, Amherst.

Sauerland, U. (this volume). "On the Content of Pronouns: Evidence from Focus". *Proceedings of SALT 10,* Ithaca, Cornell Working Papers in Linguistics.

Schwarzschild, R. (1999). "GIVENness, AVOIDF and Other Constraints on the Placement of Accent", *Natural Language Semantics* ***7****,* 141-177.

Szabolcsi, A. (1992). "Combinatory Categorial Grammar and Projection from the Lexicon", in I. Sag and A. Szabolcsi (eds.), *Lexical Matters.* CSLI Publications, Stanford.

[1] Or, if you prefer, DP. I will not distinguish between NP and DP here.

[2] better with this kind of crossing binding pattern - for reasons I don't understand the following is less happy:

The assignment that every student gave her that he most wanted everty professor to like is the last one he handed in to her.

check the examples - make sure they are consistent
and check for Englishmen throughout

if space, put in a footnote about meaning of who in CG

somewhere: [[N]] in reconstruction theory means complex as well as simple stuff