Remark on Jacobson 1999: Crossover as a local constraint [*]

Chris Barker
University of California, San Diego

August 2002

Keywords: variable-free semantics, possessives, flexible types, crossover

1. Crossover in a variable-free system

In a series of papers beginning in the late 1980's, Jacobson develops a novel and provocative theory of binding that does entirely without movement or variables. She provides many theoretical and empirical arguments in favor of a variable-free approach. However, she does not discuss quantifier scope in any detail, and therefore does not provide an account of binding out of DP.

(1) a. [Everyone$_i$'s mother]$_{\text{DP}}$ loves him$_i$.
b. [A man from every city$_i$]$_{\text{DP}}$ loves it$_i$.

Binding out of DP includes binding by quantificational possessors as in (1a), and so-called inverse linking as in (1b). What these cases have in common is a quantificational DP embedded within a larger DP in such a way that the quantificational DP can take scope over and bind a pronoun that it does not c-command.

This remark evaluates some of the strengths and the weaknesses of the variable-free program in the course of extending the fragment in Jacobson (1999) to handle a more complete range of binding constructions, including those in (1). In particular, I will show in detail how to incorporate Hendriks' Flexible Types approach to quantifier scope with excellent results, making good on a major promisory note in Jacobson's variable-free program.

One of the main points of interest below concerns the status of weak crossover in a variable-free framework.

(2) *His$_i$ mother loves everyone$_i$.

In a brief discussion of weak crossover, Jacobson (1999:135) speculates that the ungrammaticality of sentences like (2) may fall out from her variable-free system without extra stipulation. This would be a spectacular result, needless to say. However, I will suggest that the same

[*] Thanks to Daniel Büring, Greg Carlson, David Dowty, Pauline Jacobson, and Chung-Chieh Shan.

mechanisms needed to deal with binding out of DP also incorrectly generate (2), so that once the variable-free approach is extended to deal with quantifier scope and binding out of DP, ruling out crossover requires stipulation, just as in other frameworks.

Nevertheless, the variable-free approach has something new and insightful to say about crossover. The characteristic property of Jacobson's variable-free framework is that all binding relationships must be strictly local: since there is no LF movement and (obviously) no variables, the dependence of one value on another can only be established through a chain of local function/argument relationships. It follows that all constraints on binding must also be strictly local, in the sense that they can be sensitive only to syntactic information associated with immediate subconstituents. Given the standard crossover explanation in terms of long-distance LF movement and co-indexation, the fact that it is possible to formulate a local crossover constraint is quite interesting in its own right and a large plus in favor of Jacobson's program.

Another intriguing result is that the crossover constraint proposed below makes crucial reference to linear order. Although many older treatments of crossover depended on linear order, ever since Reinhart's (1983) landmark c-command account, most modern analyses have been expressed purely in terms of hierarchical dominance.

The picture that emerges is not as simple or as elegant as Jacobson's (understandably) optimistic projections. In fact, the solution developed here at least doubles the complexity of the system in Jacobson (1999), measuring complexity by the number of distinct operators. Yet even so, the complexity of the result is by no means out of line with that of other theories that develop a semantic account of binding out of DP to a comparable level of explicitness, e.g., Barker (1995) or Büring (2001).

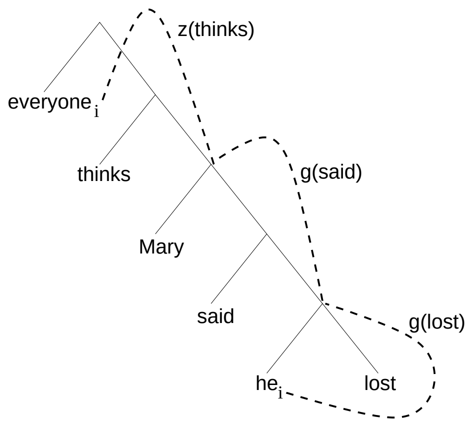
Setting aside the issue of simplicity, what makes Jacobson's variable-free framework well worth careful study is its radical but appealing notion of compositionality and semantic locality, as well as the unexpected insights it provides into crossover and other empirical phenomena.

2. Why binding out of DP is problem for Jacobson 1999

For the sake of concreteness, I will concentrate on the specific variable-free analysis presented in Jacobson (1999) (henceforth J99), though my comments will be relevant for variable-free research programs in general, notably Szabolcsi (1987, 1992), as well as for certain Type-Logical approaches discussed in Dowty (1993).

The reason that binding out of DP requires complicating the system in J99 is easy to grasp on an intuitive level by comparing the diagrams in (3) and (4). Binding in J99 depends on the interaction of two operators, **z** and **g**:

(3) Everyone$_i$ thinks Mary said he$_i$ lost.

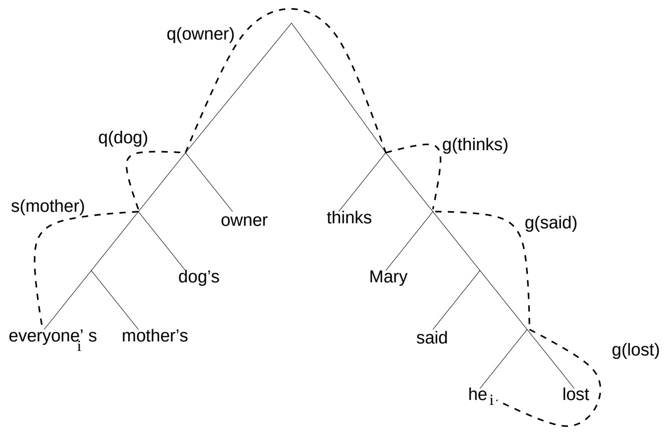


Here is how binding works in J99, starting at the bottom and working up: applying **g** to *lost* allows it to combine with the pronoun *he* in order to form a constituent whose syntactic category indicates the presence of a bindable pronoun. Applying **g** to *said* transmits the binding information one level higher, so that the syntactic category *Mary said he lost* reflects the presence of the embedded pronoun. Finally, applying **z** to *thinks* allows the quantifier denoted by *everyone* to bind the pronoun embedded in the sentential complement, resulting in a clause with no bindable pronouns. More specifically, what **z** does is “bind” the pronoun contained by its complement to the value of the matrix subject. Crucially, **z** only allows a less oblique argument position (here, the subject of *thinks*) to bind a pronoun in a more oblique argument (the sentential complement). (These operators are described in technical detail in the next section.)

This simple picture is possible only because J99 does not consider quantifier scope displacement. What I mean by ‘scope displacement’ is the ability of a quantifier to take semantic scope over material it does not c-command syntactically, i.e., what would normally be handled by Quantifier Raising in a theory that allows movement. As a result, quantifiers in J99 can bind only what they c-command, and therefore the path between a binder and a bound pronoun need only travel downwards from mother to daughter. As illustrated above in (3), this requires only one binding operator, **z**, and one binding propagation operator, **g**.

Now consider a binding out of DP example involving a quantificational possessor:

(4) Everyone$_i$'s mother's dog's owner thinks Mary said he$_i$ lost.



The reason that **g** and **z** alone are not sufficient to establish the desired binding relation is that **z** only allows binding relations between co-arguments of a single predicate (in (3), between the subject of *thinks* and its sentential complement). In (4), we can't apply **z** to *mother*, since *mother* has no oblique complement to bind into. Furthermore, even if we could somehow apply **z**, we would still be stuck, because we would need to transmit the binding relationship from the most deeply embedded possessor through the predicates that follow it (namely, *dog's* and *owner*). We couldn't use **g**, because **g** only transmits the presence of a bindable pronoun, and not the presence of a quantificational binder.

Therefore we need to extend the J99 account with two additional operators, **s** and **q**: applying **s** to *mother* allows the possessor to bind one level up, and applying **q** once to *dog* and once to *owner* transmits the binding upward to the point where **g** can take over.

3. Sketch of the J99 fragment

The J99 system consists of two formal devices for building complex expressions: combination, which makes a larger expression out of two smaller expressions; and operators, which shift the syntactic category and the meaning of an individual expression.

Semantically, combination corresponds to functional application. Syntactically, combination is divided into two operations that differ only in the linear order of functor and argument:

(5)
$$\begin{array}{llllll}
\text{combine forward:} & \text{syntax:} & \mathrm{A/B} & + \;\mathrm{B} & \Longrightarrow & \mathrm{A}\\
 & \text{semantics:} & f & \;\;\; a & \Longrightarrow & f(a)\\
 & & & & & \\
\text{combine backward:} & \text{syntax:} & \mathrm{B} & + \;\mathrm{A}\backslash\mathrm{B} & \Longrightarrow & \mathrm{A}\\
 & \text{semantics:} & a & \;\;\; f & \Longrightarrow & f(a)
\end{array}$$

Here '+' indicates the syntactic operation of string concatenation. In the schemata here and below, A and B are placeholders that can be instantiated by arbitrary categories. It is these two familiar modes of combination that make the grammar in J99 a *categorial* grammar. A simple example will illustrate both of these operations:

(6) John saw Mary.

$$\boxed{\cfrac{\begin{array}{l}\textit{John}\\ \mathrm{NP}\\ \mathbf{j}\end{array} \quad \cfrac{\begin{array}{l}\textit{saw}\\ (\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}\\ \mathbf{saw}\end{array}\quad\begin{array}{l}\textit{Mary}\\ \mathrm{NP}\\ \mathbf{m}\end{array}}{\begin{array}{l}\text{combine-forward}\\ \mathrm{S}\backslash\mathrm{NP}\\ \mathbf{saw}(\mathbf{m})\end{array}}}{\begin{array}{l}\text{combine-backward}\\ \mathrm{S}\\ \mathbf{saw}(\mathbf{m})(\mathbf{j})\end{array}}}$$

In this example, the lexical syntactic category of the transitive verb *saw* is (S\NP)/NP, which encodes the information that the first argument (the direct object) occurs to the right of the verb, and the second argument (the subject) occurs to the left of the verb phrase.

Operators, the other main formal device, take an expression and return a new expression whose syntactic category is a function of the original category, and whose semantic value is a function of the original value. What makes J99 a *combinatory* categorial grammar is that the semantic part of each operation is equivalent to a combinator. Technically, a combinator is nothing more or less than a function whose meaning can be expressed in the pure lambda calculus (i.e., with nothing but λ, variables, and parentheses) and that contains no free variables. For instance, the meaning of $\lambda xy.y$ is a combinator, but $\lambda xy.z$ is not, since z remains free.

As indicated above, the two main operators from J99 that concern us here are **z** and **g**, since they constitute the binding mechanism.

(7) **z**: syntax: $(\mathrm{A}\backslash\mathrm{NP})/\mathrm{B} \Longrightarrow (\mathrm{A}\backslash\mathrm{NP})/(\mathrm{B}^{\mathrm{NP}})$
semantics: $f \Longrightarrow z(f)$

g: syntax: $\mathrm{A}/\mathrm{B} \Longrightarrow (\mathrm{A}^{\mathrm{NP}})/(\mathrm{B}^{\mathrm{NP}})$
syntax: $\mathrm{A}\backslash\mathrm{B} \Longrightarrow (\mathrm{A}^{\mathrm{NP}})\backslash(\mathrm{B}^{\mathrm{NP}})$
semantics: $f \Longrightarrow g(f)$

where $z = \lambda xyw.(x(y(w)))(w)$ and $g = \lambda xyz.x(y(z))$. When an operator has more than one syntactic schema, as **g** does here, it means that the operator is valid for either pattern. I will follow J99 and call **g** the GEACH rule.

In the J99 system, binding is always initiated by an application of **z**, and propagated by applications of **g**:

(8) Everyone$_i$ thinks he$_i$ lost.

```
everyone      thinks             he      lost
S/(S\NP)      (S\NP)/S           NP^NP   S\NP
everyone      thinks             I       lost
              ----------------           ----------
              z                          g
              (S\NP)/(S^NP)              S^NP\NP^NP
              z(thinks)                  g(lost)
                                 -------------------
                                 combine-backward
                                 S^NP
                                 λx.lost x
              -------------------------------------
              combine-forward
              S\NP
              λx.((thinks(lost x)) x)
---------------------------------------------------
combine-forward
S
everyone(λx.thinks(lost x) x)
```

The denotation of the pronoun *he* is the identity operator **I**, where $\mathbf{I} = \lambda x.x$. The operator **z** takes the basic lexical category of *thinks*, $(\mathrm{S}\backslash\mathrm{NP})/\mathrm{S}$, and shifts it to $(\mathrm{S}\backslash\mathrm{NP})/(\mathrm{S}^{\mathrm{NP}})$. The effect can be paraphrased as saying 'I expect my first argument to contain a pronoun, and I hereby cause my second argument to bind that pronoun'. In the case at hand, the first argument of *thinks* is the embedded clause, and the second argument is the matrix subject; thus **z** enables the pronoun contained in the embedded clause to be bound by the subject.

As mentioned above, the only mechanism for binding pronouns in J99 is the **z** operator. Why won't **z** handle *Everyone$_i$'s mother loves him$_i$* (= (1a))? The reason is that **z** only allows more oblique arguments to bind (into) less oblique arguments, in accord with the notion from Bach and Partee (1980) that binders must 'F-command' their bindees. In (1a), *everyone*, the binder, is the first and only direct argument of *mother*, so the syntactic schema for **z** doesn't apply, which is equivalent to the observation that *everyone* does not F-command *him*.

4. Adding scope to J99 with Hendriks' Flexible Types

Steedman (2000) develops a combinatory categorial grammar that complements the J99 system in the sense that it treats quantifier scope in depth, but ignores quantificational binding. One might hope that a combination of the two theories could be found that accounted both for quantifier scope and for quantificational binding. Unfortunately for present purposes, Steedman relies heavily on the geach operator (i.e., functional composition) for dealing with scope. It is not clear to me how to reconcile the use of the geach rule in the two systems within a single grammar.

Fortunately, Hendriks' (1988, 1993) Flexible Types provides a theory of quantifier scope that is easy to convert to an equivalent combinatory categorial grammar, and that can be added to the J99 system without disturbing the treatment of binding. Thus one result of this remark is to show in some detail how the J99 system can easily be extended to handle quantifier scope.

Flexible Types relies on four operators, only two of which will interest us here, namely, Argument Raising and Value Raising. I will discuss two subtypes of Argument Raising first:

(9) Argument Raising 1 ($\mathbf{AR}_1$):
syntax: (S\A)/NP $\Longrightarrow$(S\A)/(S/(S\NP))
semantics: f $\Longrightarrow ar_1(f)$

Argument Raising 2 ($\mathbf{AR}_2$):
syntax: (S\NP)/A $\Longrightarrow$(S\(S/(S\NP)))/A
semantics: f $\Longrightarrow ar_2(f)$

where the combinators are given by $ar_1 = \lambda fxy.x(\lambda z.(f(z))(y))$, and $ar_2 = \lambda fxy.y(\lambda z.(f(x))z)$. $\mathbf{AR}_1$ raises the category of the first argument from an individual-denoting expression (category NP) to one denoting a generalized quantifier (category S/(S\NP)), and semantically gives that quantifier scope over the other argument position; $\mathbf{AR}_2$ does the same, except that it targets the second argument as taking wide scope. As pointed out in J99:135, this operation is also familiar from, for instance, Partee and Rooth (1983). (Hendriks gives a schema that subsumes both $\mathbf{AR}_1$ and $\mathbf{AR}_2$ within a single operator, but treating them separately here simplifies exposition.)

By applying the Argument Raising schemata to the two arguments of *see* in different orders, we get two logically distinct translations corresponding to the two scopings for the arguments.

(10) Everyone saw someone.

everyone	*saw*	*someone*
S/(S\NP)	(S\NP)/NP	S/(S\NP)
everyone	**saw**	**someone**
	$\mathbf{AR}_1$ (S\NP)/(S/(S\NP)) $ar_1(\mathbf{saw})$	
	$\mathbf{AR}_2$ (S\(S/(S\NP)))/(S/(S\NP)) $ar_2(ar_1\,\mathbf{saw})$	
	combine-forward S\(S/(S\NP)) $\lambda\mathcal{P}.\mathcal{P}(\lambda x(\mathbf{someone}(\lambda y.\mathbf{saw}\,y\,x)))$	
combine-backward S $\mathbf{everyone}(\lambda x.\mathbf{someone}(\lambda y.\mathbf{saw}\,y\,x))$		

The constants **everyone** and **someone** are the usual generalized quantifiers of type $\langle\langle e,t\rangle,t\rangle$. Applying $\mathbf{AR}_1$ to *saw*, then applying $\mathbf{AR}_2$ gives the subject wide scope, as shown. Applying $\mathbf{AR}_2$ first, then $\mathbf{AR}_1$, however, results in a different denotation for the verb phrase, namely $\lambda\mathcal{P}.\mathbf{someone}(\lambda y.\mathcal{P}(\lambda x.\mathbf{saw}\,y\,x))$, which ultimately gives the object wide scope.

Obviously, Argument Raising on its own is capable only of determining quantifier scope relations among arguments of a single local predicate. Since binding out of DP requires a quantificational DP to be able to take scope over an unbounded series of embedding predicates (see (4)), it is necessary to supplement Argument Raising with a second operator that Hendriks calls Value Raising. One familiar instance of Value Raising lifts individual-denoting expressions to generalized quantifier type, e.g., $vr(\mathbf{j}) = \lambda P.P(\mathbf{j})$. Indeed, the **lift** operator in J99 is a special case of value raising, which serves to illustrate the compatibility of the Flexible Types system with mechanisms already in place in J99.

But the kind of Value Raising that is most relevant here is one that goes beyond **lift**, and that allows a quantifier to take scope outside of the semantic domain of a higher predicate:

(11) Value Raising (**VR**):

syntax:	A\B	$\Longrightarrow$	(C/(C\A))\B **or** (C\(C/A))\B
syntax:	A/B	$\Longrightarrow$	(C/(C\A))/B **or** (C\(C/A))/B
semantics:	f	$\Longrightarrow$	$vr(f)$

where $vr = \lambda fxk.k(f(x))$, and A, B and C stand for arbitrary categories.

(12) Everyone's mother left.

everyone's	*mother*	*left*
S/(S\NP)	NP\NP	S\NP
everyone	**mother**	**left**
	VR (S/(S\NP))\NP vr(**mother**)	
	AR$_1$ (S/(S\NP))\(S/(S\NP)) ar_1(vr(**mother**))	
combine-backward S/(S\NP) λP.**everyone**($\lambda x.P$(**mother** x))		
combine-forward S **everyone**(λx.**left**(**mother** x))		

Since the possessed noun ends up as a DP modifier, that is, something of category (S/(S\NP))\(S/(S\NP)), this analysis iterates nicely: for *Everyone's mother's dog barked*, in which the quantificational possessor is two levels down from the subject position instead of one, we merely perform Value Raising and then Argument Raising first on *mother*, then on *dog*.

5. Considering a paycheck strategy

Now we are ready to address binding out of DP:

(13) Every boy$_i$'s mother loves him$_i$.

My starting point for a solution will be to consider using paycheck pronouns. As Jacobson (1999, 2000) points out, paycheck pronouns arise naturally in the J99 system merely by applying (generalized versions of) the geach operator to a pronoun, and Jacobson explains in detail how they account elegantly for a variety of empirical phenomena including paycheck sentences, functional questions, and Bach-Peters sentences.

In the J99 system, an ordinary pronoun denotes the identity function on individuals (type $\langle e,e\rangle$). A paycheck pronoun, in contrast, denotes a function from a pronoun meaning to another pronoun meaning (type $\langle\langle e,e\rangle,\langle e,e\rangle\rangle$). The obvious first stab at a paycheck analysis of binding out of DP appears in (14), but the resulting truth conditions are adequate only for a restricted range of contexts.

(14) Everyone$_i$'s mother loves him$_i$.

everyone's mother $\mathrm{S/(S\backslash NP)}$ $\lambda P.\mathbf{eo}(\lambda x.P(\mathbf{mom}\,x))$	*loves* $\mathrm{(S\backslash NP)/NP}$ **love**	*him* $\mathrm{NP^{NP}}$ **I**
g $\mathrm{(S^{NP^{NP}})/((S\backslash NP)^{NP^{NP}})}$ $g(\lambda P(\mathbf{eo}(\lambda x.P(\mathbf{mom}\,x))))$	**z** $\mathrm{(S\backslash NP)/(NP^{NP})}$ $z(\mathbf{love})$	**g** $\mathrm{(NP^{NP})^{NP^{NP}}}$ $g(\mathbf{I})$
	g $\mathrm{((S\backslash NP)^{NP^{NP}})/((NP^{NP})^{NP^{NP}})}$ $g(z(\mathbf{love}))$	
	combine-forward $\mathrm{(S\backslash NP)^{NP^{NP}}}$ $\lambda fx.(\mathbf{love}(fx))x$	
combine-forward $\mathrm{S^{NP^{NP}}}$ $\lambda f.\mathbf{everyone}(\lambda x.\mathbf{love}(f(\mathbf{mother}\,x))\,(\mathbf{mother}\,x))$		

The required generalization of **g** allows it to apply to exponents (e.g., $\mathrm{A^B} \Rightarrow \mathrm{(A^{NP})^{(B^{NP})}}$), and allows the new exponent to be a function instead of an NP (e.g., $\mathrm{A/B} \Rightarrow \mathrm{A^{NP^{NP}}/B^{NP^{NP}}}$).

The idea in (14) is that the denotation of the pronoun (and therefore of the sentence as a whole) depends on a contextually-specified Skolem function f of semantic type $\langle e,e\rangle$. If f turns out to be a function mapping mothers to their children, then the truth conditions of (14) will assert that every person x has the property that x's mother loves $f(\mathbf{mother}(x))$. As long as f is guaranteed to pick out a child of the mother of x, we get an approximation of the desired truth conditions.

There are at least two troubling aspects to this analysis. The first is that just the right Skolem function must be contextually salient. This leaves the truth conditions of the bound reading more dependent on pragmatic context than seems intuitively appropriate.

An even more serious problem is what Büring calls the uniqueness problem: in many cases, there is no function—salient or otherwise—that will provide the correct meaning. For instance, in (14), the needed function must be an inverse of the **mother** function; but the **mother** function does not have a unique inverse whenever there is a mother with more than one child, in which case there is no choice for f that will give the intuitively correct truth conditions.

Büring (2001) develops a paycheck-based analysis that overcomes (some portion of) the uniqueness problem through the use of situation semantics. On Büring's situation-semantics approach, the NP *everyone's mother* quantifies over situations containing a male person and his mother. If the pronoun denotes a function mapping situations onto boys, then as long as the situations quantified over are minimal situations in the technical sense of Kratzer (1989) or Heim (1990), then within each relevant situation there will be a unique boy, and every boy will get counted.

Unfortunately, like most attempts to use situation semantics to drive paycheck pronoun analyses, this solution suffers from a variant of the bishop problem discussed in Heim (1990). The problem is that even minimal situations can contain participants that are indistinguishable semantically, but which nevertheless need to be distinguished for the purposes of pronoun reference.

(15) a. Every male twin's twin brother$_i$ thinks he$_i$ is smart.
b. Every male twin$_i$'s twin brother thinks he$_i$ is smart.

The sentence in (15) is ambiguous, as indicated by subscripting: (15a) entails that each twin has a belief about himself, and (15b) entails that each twin has a belief about his brother. According to Büring's proposal, each minimal situation contains a twin and that twin's twin brother. But as far as the semantic information entailed to be present in the (minimal) situation, the twin brothers are indistinguishable. As a result, the paycheck function denoted by the pronoun will either fail to denote by virtue of a failure of uniqueness (which is not the correct result), or if it does denote, it will indiscriminately choose either brother, incorrectly predicting that (15a) and (15b) both mean the same thing. Büring points out (p.c.) that if we allow the pronoun to denote a function paraphrasable as 'his brother' rather than 'the brother', and bind the implicit variable corresponding to 'his' to the subject, then we can get the two readings of (15) as desired. But he goes on to observe that a slightly more complex bishop puzzle would then come into play: *Every male twin$_i$'s twin brother$_j$'s girlfriend likes him$_{i/j}$*, which admits of the same two kinds of readings, and which remains a problem for the situation/paycheck analysis.

Clearly what is needed in order to make a paycheck strategy work is a system in which the paycheck pronouns are bound directly by the controlling quantifier (in the examples above, the controlling quantifier is *every*), without the mediation of situations. We shall see below (especially in section 9) how to construct just this sort of paycheck analysis.

6. A variable-free solution

The basic ingredient in the variable-free solution I will develop here is the **s** operator as defined in J99:136:

(16) **s**: syntax: $(B/A)\backslash NP \Longrightarrow (B/(A^{NP}))\backslash NP$
syntax: $(B\backslash A)/NP \Longrightarrow (B\backslash(A^{NP}))/NP$
semantics: $f \Longrightarrow s(f)$

where the combinator $s = \lambda fxy.(f(x))(y(x))$. In contrast with **z**, **s** allows a more oblique argument to bind a less oblique argument. For instance, a quantificational possessor can now bind (into) the verb phrase:

(17) Everyone$_i$'s mother loves him$_i$

everyone's S/(S\NP) **everyone**	*mother* NP\NP **mother**	*loves* (S\NP)/NP **love**	*him* NP^{NP} **I**
	VR (S/(S\NP))\NP vr(**mother**)	**g** $((S\backslash NP)^{NP})/(NP^{NP})$ g(**love**)	
	s $(S/((S\backslash NP)^{NP}))\backslash NP$ $s(vr($**mother**$))$	combine-forward $(S\backslash NP)^{NP}$ $\lambda x.$**love** x	
	AR$_1$ $(S/((S\backslash NP)^{NP}))\backslash(S/(S\backslash NP))$ $ar_1(s(vr($**mother**$)))$		
combine-backward $S/((S\backslash NP)^{NP})$ $\lambda R.$**everyone**$(\lambda x.(Rx)($**mother** $x))$			
combine-forward S **everyone**$(\lambda x.($**love** $x)($**mother** $x))$			

This gives the desired reading, on which each person x has the property that x's mother loves x, without postulating pragmatically-controlled Skolem functions.

Unfortunately, as noted in J99:135, if left unconstrained, **s** can also allow a direct object to bind (into) a subject, resulting in a crossover violation:

(18) *His$_i$ mother loves everyone$_i$.

$$\begin{array}{|llll|}
\hline
\textit{his} & \textit{mother} & \textit{loves} & \textit{everyone}\\
\mathrm{NP}^{\mathrm{NP}} & \mathrm{NP}\backslash\mathrm{NP} & (\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP} & \mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})\\
\mathbf{I} & \mathbf{mother} & \mathbf{love} & \mathbf{everyone}\\
\cline{2-3}
 & \mathbf{g} & \mathbf{s} & \\
 & (\mathrm{NP}^{\mathrm{NP}})\backslash(\mathrm{NP}^{\mathrm{NP}}) & (\mathrm{S}\backslash(\mathrm{NP}^{\mathrm{NP}}))/\mathrm{NP} & \\
 & g(\mathbf{mother}) & s(\mathbf{love}) & \\
\cline{1-3}
\text{combine-backward} & & \mathbf{AR}_1 & \\
\mathrm{NP}^{\mathrm{NP}} & & (\mathrm{S}\backslash(\mathrm{NP}^{\mathrm{NP}}))/(\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})) & \\
\lambda x.\mathbf{mother}\,x & & ar_1(s(\mathbf{love})) & \\
\cline{3-4}
 & & \text{combine-forward} & \\
 & & \mathrm{S}\backslash(\mathrm{NP}^{\mathrm{NP}}) & \\
 & & \lambda P.\mathbf{everyone}(\lambda x.(\mathbf{love}\,x)(P\,x)) & \\
\hline
\text{combine-backward} & & & \\
\mathrm{S} & & & \\
\mathbf{everyone}(\lambda x.(\mathbf{love}\,x)(\mathbf{mother}\,x)) & & & \\
\hline
\end{array}$$

Because of this kind of overgeneration, J99 rejects **s** as a legitimate operator. But we need **s** to derive grammatical sentences like (17). Apparently, we must provide some way of constraining derivations involving **s** to prevent crossover. But first, one more operator needs to be motivated and defined before we have a full enough picture to develop a general crossover constraint.

7. Binding propagation upward: q

As illustrated above in the diagram in (4), somehow we need to transmit the binding provided by the application of **s** upwards through a sequence of possessed nouns. Therefore let **q** be a new operator designed to serve this purpose.

(19) **q**:

$$\begin{array}{lll}
\text{syntax:} & (\mathrm{B}/\mathrm{A})\backslash\mathrm{NP} & \Longrightarrow (\mathrm{B}/(\mathrm{A}^{\mathrm{NP}}))\backslash(\mathrm{C}/((\mathrm{C}\backslash\mathrm{NP})^{\mathrm{NP}}))\\
\text{syntax:} & (\mathrm{B}/\mathrm{A})/\mathrm{NP} & \Longrightarrow (\mathrm{B}/(\mathrm{A}^{\mathrm{NP}}))/(\mathrm{C}\backslash((\mathrm{C}/\mathrm{NP})^{\mathrm{NP}}))\\
\text{semantics:} & f & \Longrightarrow q(f)
\end{array}$$

Here q is the combinator $\lambda f \mathcal{P} p.\mathcal{P}(\lambda ab.(f(b))(p(a)))$. Jacobson named **z** because it was a dual of **s**, and 'z' graphically is an 's' reversed; in the same spirit, **q** is the dual of the geach rule **g**. Note that **q** raises the most oblique argument from an individual to a generalized quantifier, very much like Argument Raising does. This is forced by the fact that a quantifier can only bind a pronoun if it takes scope over it.

(20) Everyone$_i$'s mother's dog loves him$_i$

eo's mom's	*dog*	*loves him*
$\mathrm{S/((S\backslash NP)^{NP})}$	$\mathrm{NP\backslash NP}$	$\mathrm{(S\backslash NP)^{NP}}$
$\lambda R.\mathbf{eo}(\lambda x.(R\,x)(\mathbf{mom}\,x))$	$\mathbf{dog}$	$\lambda x.\mathbf{love}\,x$
	VR	
	$\mathrm{(S/(S\backslash NP))\backslash NP}$	
	$vr(\mathbf{dog})$	
	q	
	$\mathrm{(S/((S\backslash NP)^{NP}))\backslash(S/((S\backslash NP)^{NP}))}$	
	$q(vr(\mathbf{dog}))$	

combine-backward
$\mathrm{S/((S\backslash NP)^{NP})}$
$\lambda R.\mathbf{everyone}(\lambda x.(Rx)(\mathbf{dog}(\mathbf{mother}\,x)))$

combine-forward
S
$\mathbf{everyone}(\lambda x.(\mathbf{love}\,x)(\mathbf{dog}(\mathbf{mother}\,x)))$

Notice that after applying Value Raising and **q** to *dog*, the denotation of *dog* shifts to become a modifier on expressions of type $\mathrm{S/((S\backslash NP)^{NP})}$. Since this is the type of *everyone's mother*, we can iterate possessed nouns, inserting as many as we like between the quantifier and the verb phrase (*everyone$_i$'s mother's friend's lawyers' date's ... dog loves him$_i$*).

The new operator **q** also automatically accounts for inverse linking cases including the example given above in (1b), repeated here:

(21) A man from every city$_i$ loves it$_i$.

a	*man*	*from*	*every city*
$(S/(S\backslash NP))/N$	N	$(N\backslash N)/NP$	A
a	**man**	**from**	**ec**
q		**VR**	
$(S/((S\backslash NP)^{NP}))/(A\backslash(A/N)^{NP})$		$((A\backslash(A/N))\backslash N)/NP$	
$q(\mathbf{a})$		$vr(\mathbf{from})$	
		s	
		$(A\backslash((A/N)^{NP})\backslash N)/NP$	
		$s(vr(\mathbf{from}))$	
		$\mathbf{AR}_1$	
		$((A\backslash((A/N)^{NP})\backslash N))/A$	
		$ar_1(s(vr(\mathbf{from})))$	
		combine-forward	
		$(A\backslash(A/N)^{NP})\backslash N$	
		$\lambda PD.\mathbf{ec}(\lambda x.(Dx)(\mathbf{from}\,x\,P))$	
	combine-backward		
	$A\backslash(A/N)^{NP}$		
	$\lambda D.\mathbf{ec}(\lambda x.(Dx)(\mathbf{from}\,x\,\mathbf{man}))$		
combine-forward			
$S/((S\backslash NP)^{NP})$			
$\lambda P.\mathbf{every{-}city}(\lambda x.(\mathbf{a}(\mathbf{from}\,x\,\mathbf{man}))(Px))$			

Where A is the category of a generalized-quantifier-denoting NP, that is, $(S/(S\backslash NP))$.

The fact that examples like (21) are handled in a relatively straightforward way is to the credit of the (extended) J99 system, since this type of sentence creates considerable difficulty in some Quantifier Raising analyses (see, e.g., Heim and Kratzer (1998:234) for discussion).

The derivation sketched in (21) and some of the derivations below use minor variations on Value Raising, Argument Raising, and **s** that enable them to skip over an intervening argument; for instance, in (21), **s** must connect *every city* with the determiner *a* despite the fact that the next argument is the nominal *man*. The variant syntactic schema is $((B\backslash A)\backslash N)/NP \Rightarrow ((B\backslash(A^{NP}))\backslash N)/NP)$, and the variant combinator is $\lambda fxny.(fxn)(yx)$. See J99:134 for the motivation and the details for exactly analogous variations on the **z** operator in J99:134. The need to add multiple cases to the syntactic schemata of the operators seems to be an annoying but characteristic property of the J99 system.

But there is a potential payoff from the ability to fine-tune the syntactic details of the operator variants, since this allows us to rule various binding configurations in or out. In fact, this technique comes frustratingly close to providing an elegant account of crossover. As Chung-Chieh Shan points out (personal communication), the grammatical (17) uses one syntactic pattern for **s** (namely, (B/A)\NP), and the ungrammatical (18) uses the other pattern ((B\A)/NP). And indeed, the second pattern is highly suspicious-looking: it seems to permit a quantifier on the right (the NP) to bind into a constituent to its left (the A)—which sounds exactly like crossover. If we keep the first pattern and eliminate the second, we might hope to generate all the good binding cases and none of the crossover examples. This is not quite as simple as Jacobson's original idea of eliminating **s** entirely, but it is in the same spirit.

Unfortunately, the derivation in (21) requires (a two-argument version of) the second type of pattern, so both types of **s** combinator are needed after all. And in fact, the existence of inverse linking means that no solution based on restricting the application of **s** can work. The reason is that binding operators in the J99 system must apply to the first predicate that combines with the quantificational NP. Therefore any attempt to restrict the application of **s** must be based on the syntactic category of *from*. But in general, DPs can occur as subjects, direct objects, or in some other syntactic position independently of its internal syntactic structure; as a result, it is not possible to tell by examining the syntactic category of *from* whether the pronoun in question will precede or follow the DP containing *from* and its quantificational complement. In other words, there simply isn't enough information present to detect crossover violations locally at the position at which the binding operator (**z** or **s**) applies, and we must look higher in the tree.

8. A local crossover constraint

The key to arriving at a suitable crossover constraint is the following theorem of the J99 system: in any well-formed derivation, there is a one-to-one correspondence between each application of a binding operator and a single specific bound pronoun. In the J99 system, there is only one binding operator, **z**, so there will be exactly as many applications of **z** as there are bound pronouns; but the analogous theorem continues to hold for the extended system developed here, except that the set of binding operators contains **s** in addition to **z**.

In fact, we can trace each unique binding path through the syntactic part of the derivation as a series of syntactic exponents, where each element in the path is connected to the one before it or the one after it by an application of one of the binding propagators, either **g** or **q**. (The series of dotted links in the diagrams in (3) and (4) illustrate what I mean by a "binding path".) Furthermore, there will always be a unique place in the binding path at which the path ceases to travel upwards, and begins to descend; I will informally call this the "peak".

It is at the peak that crossover violations can be detected. Crossover violations typically contain a binding path peak that has the following syntactic form:

(22) combine-backward(B^{NP}, $A\backslash B^{NP}$) $\Longrightarrow$ A

The category on the left (B^{NP}) contains a pronoun that can be bound, and the category on the right provides a value for that pronoun (i.e., binds it). Since the bindee is to the left of the binder, crossover has occurred.

Unfortunately, we cannot simply prohibit derivational steps that resemble (22). Once again, the flexibility of syntactic categories that is the hallmark of combinatory categorial grammars creates problems. For one thing, crossover violations can also be arrived at via forward combination, for instance, if we were to lift (i.e., apply Value Raising to) the leftmost category in (22).

Although recognizing when a pronoun has been bound is intuitively obvious, characterizing a robust, explicit version of this notion in the J99 system is not trivial. I will only sketch one way this could be formalized. The idea is to associate each exponent in the syntax with a feature declaring whether that exponent corresponds to a bindee ('$-$') (something that is seeking to be bound) or else a binder ('+') (something that is seeking to bind).

- Exponents introduced by pronouns are bindees, and are marked '$-$'.

- Exponents introduced by the binding rules (**s** and **z**) are binders, and are marked '+'.

- Exponents introduced by **g** are all bindees ('$-$').

- Exponents introduced by **q** are all binders ('+').

- Argument Raising and Value Raising preserve the binder/bindee status of any exponents in their inputs.

- New exponents introduced by Argument Raising or Value Raising are neutral (unspecified), and must take on the status of any exponent they match during combination.

The last clause requires elaboration in order to be fully explicit, but since it does not figure in any of the examples in this remark, I will leave it at that.

Given an annotated derivation (e.g., (29) below), we can predict the presence of a crossover violation as follows:

(23) **A variable-free crossover constraint**: if syntactic combination (either forward or backward) matches some $\mathrm{NP}^{\mathrm{NP}-}$ on the left with $\mathrm{NP}^{\mathrm{NP}+}$ on the right, crossover has occurred.

To see how this constraint makes predictions, here are all the instances of syntactic combination corresponding to binding path peaks in the derivations given in (8), (17), (18), (20), and (21):

(24)	(8)	Everyone$_i$	[thinks] $(\mathrm{S}\backslash\mathrm{NP})/(\mathrm{S}^{\mathrm{NP}+})$	[he$_i$ lost]. $\mathrm{S}^{\mathrm{NP}-}$
	(17)		[Everyone$_i$'s mother] $\mathrm{S}/((\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}+})$	[loves him$_i$]. $(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}-}$
	(18)		*[His$_i$ mother] $\mathrm{NP}^{\mathrm{NP}-}$	[loves everyone$_i$] $\mathrm{S}\backslash(\mathrm{NP}^{\mathrm{NP}+})$
	(20)		[Everyone$_i$'s mother's dog] $\mathrm{S}/((\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}+})$	[loves him$_i$] $(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}-}$
	(21)		[A man from every city$_i$] $\mathrm{S}/((\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}+})$	[loves it$_i$] $(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}-}$

Of these examples, only in (18) is there a negative exponent on the left, and only in (18) is there a crossover violation. This is an instance of backward combination, and in the course of instantiating the combination schema, the category on the left, $\mathrm{NP}^{\mathrm{NP}-}$, matches the subpart $\mathrm{NP}^{\mathrm{NP}+}$ of the category on the right. Since the $\mathrm{NP}^{\mathrm{NP}-}$ is on the left, a crossover violation is predicted. Thus the proposed constraint detects crossover perfectly in the examples considered so far.

Here are some additional examples that all have the same categories at their peaks as in (18), i.e., $\mathrm{NP}^{\mathrm{NP}-}$ on the left and $\mathrm{S}\backslash(\mathrm{NP}^{\mathrm{NP}+})$ on the right (given the derivations as sketched beneath them) and therefore are all correctly predicted to be crossover violations:

(25) *He$_i$ loves everyone$_i$.
[he] [**s**(loves) everyone]

*He$_i$ loves everyone$_i$'s mother.
[he] [**q**(loves) (everyone's **s**(**vr**(mother)))]

*His$_i$ father loves everyone$_i$'s mother.
[his **g**(father)] [**q**(loves) (everyone's **s**(**vr**(mother)))]

And, for good measure, an example of crossover correctly predicted within NP:

(26) *A	[man from it$_i$]	[in every city$_i$]	became mayor.
	[**g**(man) **g**(from) it]	[**s**(**vr**(in)) every city]	
	$\mathrm{N}^{\mathrm{NP}-}$	$\mathrm{N}\backslash\mathrm{N}^{\mathrm{NP}+}$	

So annotating exponents with pluses and minuses (whose only motivation is to provide a means for dealing with crossover) is tedious, but permits a crossover constraint that gives promising results.

Why should this theory of binding make it so awkward to define explicitly when a particular argument position has been bound? The reason is precisely because the J99 system rejects coindexation. If there were essential use of variables in this system, each pronoun would have a syntactic index that would translate as a (potentially) different variable symbol, and we could detect a binding relation just by comparing indices or variable symbols. So one of the costs of going variable-free is an increased difficulty in defining when a quantifier and a pronoun stand in a binding relation.

Although it may be easier in other theories to define a binding relation, it would be a mistake to assume that such theories would necessarily be in a better position to state a local crossover constraint. The problem is that, as we have seen, the correct place to detect a crossover violation is at the binding path peak. Because of this, theories that use assignment functions cannot hope to provide a local characterization of crossover. The reason is that detecting crossover depends on knowing whether combining two expressions would cause a pronoun to become bound, and it is not in general possible to determine this reliably based only on the denotation of the expressions, even with access to the relevant assignment functions (particularly difficult cases involve pronouns embedded within tautologies).

In contrast, the prohibition against variables in the J99 system forces each expression to encode a list of its unbound pronouns on its syntactic category in the form of a syntactic exponent. Because each expression wears its bound pronouns on its sleeve, so to speak, the grammar provides exactly the information needed to tell whether a crossover violation has occurred merely by examining syntactic categories at the peak. Thus one advantage of the J99 approach is that it provides a way of stating a crossover constraint that is strictly compositional and strictly local both syntactically and semantically.

9. A complication that leads back to paychecks

There is a complication due to the possibility of having bound pronouns within nominals. Once again, the machinery forced upon us by avoiding essential use of variables is complex; but there is an interesting theoretical point, which is that the analysis would be more complicated still if we weren't able to take advantage of a variation on Büring's paycheck strategy.

Consider what happens when nominals contain bound pronouns:

(27) a. Everyone$_i$'s mother$_j$'s friend$_k$'s book about her$_j$ affair with him$_k$ surprised him$_i$.
b. A picture of a picture of a picture of every man's mother's friend's dog on her sofa on his bed in its doghouse sold for enough money to allow him to force her to feed it well.

In the sentences in (27), some binding paths must be discharged at the level of the nominal, but some others must continue to be transmitted upwards to the level where the subject combines with the VP.

The solution adopted here is straightforward but ugly: let $\mathbf{q}'$ be a variant of $\mathbf{q}$ that discharges locally one of the binders, passing the remainder upwards. As long as the discharged binder corresponds to the controlling quantifier, paycheck denotations can do the rest of the work.

(28) $\mathbf{q}'$: syntax: $((\mathrm{B}/\mathrm{A})\backslash\mathrm{NP})^{\mathrm{NP}-} \Rightarrow (\mathrm{B}/(\mathrm{A}^{\mathrm{NP}+}))\backslash(\mathrm{C}/((\mathrm{C}\backslash\mathrm{NP})^{\mathrm{NP}+})^{\mathrm{NP}+})$
syntax: $((\mathrm{B}\backslash\mathrm{A})/\mathrm{NP})^{\mathrm{NP}-} \Rightarrow (\mathrm{B}\backslash(\mathrm{A}^{\mathrm{NP}+})/(\mathrm{C}/((\mathrm{C}\backslash\mathrm{NP})^{\mathrm{NP}+})^{\mathrm{NP}+})$
semantics: $f \Rightarrow q'(f)$

Here q' is the combinator $\lambda f \mathcal{P} P.\mathcal{P}(\lambda zyx.((f(y))(x))(P(z)))$. This operator peels off the bottommost binder and feeds it to the predicate undergoing the operation, transmitting the other binder (in the general case, not shown here, all remaining binders) upwards.

(29) Everyone$_i$'s mother's picture of his$_i$ dog pleases him$_i$.

everyone's
$\mathrm{S/(S\backslash NP)}$
everyone

mother's
$\mathrm{NP\backslash NP}$
mother

Value Raise
$\mathrm{(S/(S\backslash NP))\backslash NP}$
$vr(\mathbf{mother})$

---

s
$\mathrm{(S/((S\backslash NP)^{NP+}))\backslash NP}$
$s(vr(\mathbf{mother}))$

---

s
$\mathrm{(S/(((S\backslash NP)^{NP+})^{NP+}))\backslash NP}$
$s(s(vr(\mathbf{mother})))$

Argument Raise 1
$\mathrm{(S/(((S\backslash NP)^{NP+})^{NP+}))\backslash(S/(S\backslash NP))}$
$ar_1(s(s(vr(\mathbf{mother}))))$

picture
$\mathrm{(NP\backslash NP)/NP}$
pix

VR
$\mathrm{((S/(S\backslash NP))\backslash NP)/NP}$
$vr(\mathbf{pix})$

---

g
$\mathrm{(((S/(S\backslash NP))\backslash NP)^{NP-})/(NP^{NP-})}$
$g(vr(\mathbf{pix}))$

of his dog
$\mathrm{NP^{NP-}}$
$\lambda x.\mathbf{dog}\,x$

---

combine-forward
$\mathrm{((S/(S\backslash NP))\backslash NP)^{NP-}}$
$\lambda xyG.G((\mathbf{pix}(\mathbf{dog}\,x))\,y)$

---

q′
$\mathrm{(S/((S\backslash NP)^{NP+}))\backslash(S/(((S\backslash NP)^{NP+})^{NP+}))}$
$q'(\lambda xyG.G((\mathbf{pix}(\mathbf{dog}\,x))\,y))$

---

combine-backward
$\mathrm{S/(((S\backslash NP)^{NP+})^{NP+})}$
$\lambda R.\mathbf{everyone}(\lambda x.((Rx)x)(\mathbf{mother}\,x))$

---

combine-backward
$\mathrm{S/((S\backslash NP)^{NP+})}$
$\lambda P.\mathbf{everyone}(\lambda x.((Px)((\mathbf{pix}(\mathbf{dog}\,x)))(\mathbf{mother}\,x)))$

pleases him
$\mathrm{(S\backslash NP)^{NP-}}$
$\lambda x.\mathbf{please}\,x$

---

combine-forward
S
$\mathbf{everyone}(\lambda x((\mathbf{please}\,x)((\mathbf{pix}(\mathbf{dog}\,x))(\mathbf{mother}\,x))))$

We apply **s** to *mother* twice, once for each bound pronoun. One binder is absorbed by the nominal *picture of his dog*, and the second travels upward to provide a binder for the pronoun in the verb phrase.

Within this basic strategy, paycheck denotations allow the pronouns to select as a binder any element in the possessive chain. For instance, to get *Everyone*$_i$*'s mother*$_j$*'s picture of her*$_j$ *dog pleases him*$_i$, the analysis is the same as in (29), except that the pronoun *her* gets a paycheck denotation (i.e., undergoes the geach rule—see example (14) above) and so depends on a contextually-supplied paycheck function (which in this example will need to be the **mother** function).

It will also be necessary to generalize **q** and **q′** to handle an arbitrary number of binders. Furthermore, variants of all of these operators will be needed for skipping over intervening argument positions. When all of these variations are considered, the full family of **q** operators becomes distressingly complex. Once again, having to constantly keep track of each individual binding path without the convenience of relying on variable names makes stating the grammar more complicated than it would be if we had assignment functions to work with.

10. Conclusions

So what makes a variable-free semantics worth pursuing? Should we be surprised or impressed to learn that it is possible to build a descriptively adequate semantics that does not make essential use of variables? Not very: it is well-known that variable-free systems exist that have an expressive power exactly equivalent to the pure untyped lambda calculus. The prime example is Combinatory Logic, due separately to Schönfinkel and Curry (see, e.g., Barendregt (1984:chapter 7) for a standard reference establishing this equivalence). It follows that any semantics that can be rendered in a lambda calculus has an equivalent variable-free version, so adopting a variable-free approach is no restriction at all from the point of view of expressive power.

What is not obvious is that it is possible to construct a variable-free semantics that satisfies various compositionality constraints. In fact, Jacobson advocates a radically strict compositional discipline that she calls **direct compositionality**: the idea is that each syntactic constituent has a well-formed and complete denotation that does not depend on any linguistic element external to that expression.

On Jacobson's view, taking direct compositionality seriously has consequences for expressions that contain pronouns. In the standard treatment, pronouns translate as variables, and the denotation of an expression that contains a pronoun depends on an external assignment function. But if variables are prohibited, then of course assignment functions cannot be used (since they are functions from variables to individuals), and the denotation of the expression itself must provide explicit semantic access to the argument position occupied by the pronoun. For instance, we saw above that if *loves Mary* denotes a function from individuals to truth values, then *loves him* denotes something more complicated, namely, a function from individuals to a function from individuals to truth values.

Consequently, variable-free direct compositionality entails that an expression that contains a bindable pronoun will have a syntactic category that is different from an expression that is otherwise identical but that contains a proper name in the place of the pronoun. The implicit prediction is that expressions that contain bindable pronouns could have a restricted syntactic distribution, and the existence of weak crossover confirms this prediction. In other words, even though weak crossover does not "fall out" for free in the variable-free approach, the possibility that a constraint like weak crossover might exist does fall out. In most other theories I am aware of, no matter how easy it might be to state a constraint prohibiting weak crossover, the existence of such a constraint always comes as an unpleasant surprise.

One of the unusual aspects of the account of weak crossover given above is that it crucially relies on linear order. Is an order-sensitive theory of weak crossover viable? After all, it is well-known that anaphors can sometimes precede their antecedents, as in *Near him$_i$, John$_i$ saw a snake*. But this remark considers only quantificational binding, not anaphora in general. Note that an attempt at quantificational binding in the same configuration results in ungrammaticality: **Near him$_i$, every boy$_i$ saw a snake*. Thus weak crossover is a constraint on quantificational binding, not anaphora in general.

However, weak crossover is usually assumed to apply also to certain configurations involving WH-extraction.

(30) a. Which of his$_i$ relatives does every boy$_i$ love ___?
b. *Which of his$_i$ relatives ___ loves every boy$_i$?

The generalization, roughly, is that a quantifier is able to bind (into) a WH-expression only if it would have been able to bind (into) an expression in the position of the WH-trace. In particular, note that in (30a), the quantifier denoted by *every* is able to bind the pronoun *his* even though the pronoun is to the left. In a derivational treatment, this is no mystery if binding relations are established before WH-movement occurs; but in a framework that takes direct compositionality seriously, movement is not available, and it would be necessary to build the binding relation into the syntactic relationship between the quantifier and the expression containing the WH-trace. As far as I know, no one has yet worked out in detail the syntax and the semantics of WH-movement in a variable-free framework to anywhere near the requisite level of completeness, though I am guardedly optimistic that doing so would allow a reasonably natural generalization of the approach to crossover suggested here.

Does the account of weak crossover proposed here translate to other approaches? The obvious candidates are other categorial grammars. Dowty (1993) establishes a close parallel between an earlier version of the J99 system and a certain Type-Logical approach to binding based on Hepple (1990). Except for Dowty (1993) and an inconclusive discussion in chapter 9 of Carpenter (1996), most discussions of binding in the Type-Logical tradition concentrate on binding of reflexives rather than on quantificational binding, including e.g., Moortgat (1990) and Hepple (1990) (incidentally, Szabolcsi's (1987, 1993) discussions of binding in a variable-free categorial grammar also concentrate mainly on reflexives). Dowty (1993) does discuss quantificational binding and crossover, but, like J99, does not consider binding out of DP. It is not yet clear how to extend the ideas in Dowty (1993) to handle the full range of binding out of DP, especially some of the more difficult cases discussed here, such as (1b) and (27).

I have suggested that binding out of DP poses a severe challenge to the simplicity and elegance of the J99 system, since it requires adding so many new operators. However, it may be possible to factor the principles embodied in the various operators proposed here into a smaller set of more general operators, although doing so may cause the resemblance to the specific system in J99 to become fairly remote; see Shan and Barker (ms.) for one attempt along these lines.

In the meantime, Jacobson's variable-free system stands as a viable, robust example of a radically compositional theory that rewards those who study it with unexpected insights into empirical phenomena.

11. References

Bach, Emmon and Barbara Partee. 1980. Anaphora and Semantic Structure. Papers from the Parasession on Pronouns and Anaphora, J. Kreiman and A. Ojeda, eds. CLS:1–28.

Barker, Chris. 1995. *Possessive Descriptions.* CSLI, Stanford.

Barendregt, Hendrik. 1984. *The Lambda Calculus—Its Syntax and Semantics.* North-Holland, Amsterdam.

Büring, Daniel. 2001. A situation semantics for binding out of DP. To appear in the proceedings of SALT 11.

Carpenter, Bob. 1996. *Type-Logical Semantics.* MIT Press, Boston.

Dowty, David. 1993. 'Variable-Free' Syntax, Variable-Binding Syntax, the Natural Deduction Lambek Calculus, and the Crossover Constraint. In *Proceedings of the 1992 West Coast Conference on Formal Linguistics.*

Jacobson, Pauline. 1999. Towards a variable free semantics. *Linguistics and Philosophy* **22**:117–184.

Jacobson, Pauline. 2000. Paycheck pronouns, Bach-Peters sentences, and variable-free semantics. *Natural Language Semantics* **8**:77–155.

Heim, Irene. 1990. E-Type Pronouns and Donkey Anaphora. *Linguistics and Philosophy* **13**: 137–177.

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*, Blackwell.

Hendriks, Herman. 1988. Type Change in Semantics: the Scope of Quantification and Coordination, in E. Klein and J. van Benthem, eds., *Categories, Polymorphism and Unification*, ITLI, Amsterdam, 96–119.

Hendriks, Herman. 1993. *Studied Flexibility*, ILLC Dissertation Series, Amsterdam.

Hepple, Mark. 1990. The Grammar and Processing of Order and Dependency: a Categorial Approach. PhD dissertation, University of Edinburgh.

Kratzer, Angelika. 1989. An Investigation of the Lumps of Thought. *Linguistics and Philosophy* **12.5**:607–653.

Moortgat, Michael. 1990. The quantification calculus. In H. Hendriks and M. Moortgat, eds. *Theory of Flexible Interpretation* Esprit DYANA Deliverable R1.2.A, Institute of Language, Logic, and Information, University of Amsterdam.

Partee, Barbara Hall and Mats Rooth. 1983. Generalized conjunction and type ambiguity. In Rainer Bäuerle, Christoph Schwarze, and

Arnim von Stechow, eds., *Meaning, Use, and Interpretation of Language*, 361–383.

Reinhart, Tanya. 1983. *Anaphora and Semantic Interpretation.* University of Chicago Press.

Shan, Chung-Chieh and Chris Barker. (ms.) A unified explanation for crossover and superiority in an abstraction-by-movement theory of binding.

Steedman, Mark. 2000. *The Syntactic Process.* MIT Press.

Szabolcsi, Anna. 1987. Bound Variables in Syntax (Are There Any?). In Jeroen Groenendijk, Dijk de Jongh, and Martin J. B. Stokhof, eds. *Proceedings of the 6th Amsterdam Colloquium.* ILLI, Amsterdam. Revised version in Renate Bartsch et al., eds. 1989. *Semantics and Contextual Expression.* Foris, Dordrecht.

Szabolcsi, Anna. 1992. Combinatory Grammar and Projection from the Lexicon. In Anna Szabolcsi and Ivan Sag, eds. *Lexical Matters*, CSLI, Stanford. 241–268.