

Binding without variables: Solving the under-generation problems*

Yimei Xiang
Rutgers University

Abstract The variable-free semantics of Jacobson (1999, 2000, 2014) derives binding relations by the local application of the z-rule. This rule, however, under-generates binding. This paper makes two contributions: (i) replacing the z-rule with a more flexible rule called *i* (a la the *W*-combinator of Szabolcsi 1992), which allows for more binding relations; (ii) enriching Jacobson’s variable-free system and proposing a two-dimensional analysis to account for the interactions between scoping and binding. Issues to be covered include binding into adjuncts, possessor binding, scope ambiguity, inverse linking, weak crossover, and ‘paycheck pronouns’.

Keywords: Pronouns, binding, variable-free, direct compositionality, non-c-commanding binding, scoping, inverse linking, weak crossover, paycheck pronouns, categorial grammar

1 Introduction

1.1 Jacobson’s variable-free approach

Jacobson’s (1999; 2000; 2014) variable-free semantics assumes no index/variable in the grammar.¹ In contrast to ‘variable-full’ approaches, which translate pronouns as variables and interpret these variables via assignment functions (as in Kratzer & Heim 1998), ‘variable-free’ approaches analyze pronouns as instances of the category NP^{NP} that denote identity functions over entities, as in (1).

$$(1) \quad \text{CAT}(it) = \text{NP}^{\text{NP}}, \text{ and } \llbracket it \rrbracket = \lambda x_e . x$$

More generally, pronouns and expressions that contain an unbound pronoun are treated as categories with a superscript NP abstraction, called ‘ X^{NP} -categories’, as exemplified in (2-ii). In meaning, they are the same as the functional category X|NP , which selects an NP argument and returns an X.² However, in syntax, they don’t

* For helpful discussions, I thank Simon Charlow, Michael Glanzberg, Pauline Jacobson, Haoze Li, Ken Safir, and the abstract reviewers and the audiences of SALT 31. All errors are mine.

¹ The variable-free hypothesis is widely assumed in works on Categorial Grammar. The system presented in this paper is developed from a series work by Jacobson beginning in the late 1980s.

² I use ‘ $A|B$ ’ as the joint representation for A/B and $A \setminus B$, which select a B on the right and on the left,

actively merge with an NP. For example, *their mom* has the same semantics as *__'s mom*, but it doesn't take an NP argument.

	Category	Meaning
(i) <i>their mom</i>	NP^{NP}	$\lambda x_e.mom(x)$
(ii) <i>__'s mom</i>	$\text{NP} \backslash \text{NP}$	$\lambda x_e.mom(x)$
(i) <i>invited him</i>	$(\text{S} \backslash \text{NP})^{\text{NP}}$	$\lambda y_e \lambda x_e.ivt(x, y)$
(ii) <i>invited __</i>	$(\text{S} \backslash \text{NP}) / \text{NP}$	$\lambda y_e \lambda x_e.ivt(x, y)$

To tackle composition with pronouns, Jacobson assumes a type-shifting rule $\text{g-sup}(\text{script})$, defined as in (3). Applying g-sup to a functional expression α that selects an X argument allows this α to apply to an X^{NP} argument, and it passes up the information about the unbound pronoun contained in this X^{NP} .

(3) **The g-sup -rule** (for passing up info about unbound pronouns)

For any expression α s.t. $\text{CAT}(\alpha) = A|B$, we have:

$$\text{CAT}(\text{g-sup}(\alpha)) = A^C|B^C, \text{ and } \llbracket \text{g-sup}(\alpha) \rrbracket = \lambda V_{cb} \lambda C_c. \llbracket \alpha \rrbracket (V(C))$$

Consider (4) for example. The verb *invited* selects an NP object, and thus it cannot directly apply to the pronominal expression *his mom*, which is an NP^{NP} ; however, the type-shifted verb $\text{g-sup}(\text{invited})$ can apply to an NP^{NP} -category. The resulting VP inherits the superscript NP, which tells that the pronoun remains unbound in this VP.

(4) Composition of *invited his mom* (without binding)³

$$\frac{\frac{\text{invited}}{(\text{S} \backslash \text{NP}) / \text{NP}} \quad \lambda y_e \lambda x_e.ivt(x, y)}{(\text{S} \backslash \text{NP})^{\text{NP}} / \text{NP}^{\text{NP}}} \quad \lambda f_{ee} \lambda y_e \lambda x_e.ivt(x, f(y)) \quad \text{g-sup} \quad \frac{\text{his mom}}{\text{NP}^{\text{NP}}} \quad \lambda y_e.mom(y)}{(\text{S} \backslash \text{NP})^{\text{NP}} \quad \lambda y_e \lambda x_e.ivt(x, mom(y))} >_{\text{FA}}$$

The g-sup rule is analogous to the so-called ‘Geach rule’ (Geach 1970), referred to as ‘ $\text{g-sl}(\text{ash})$ ’. These two g -rules have the same effect in semantics, but g-sl is used to pass up the information about an unsaturated syntactic argument. For example, in *didn't come*, applying g-sl to the auxiliary *didn't* shifts it from an S/S to an $(\text{S} \backslash \text{NP}) / (\text{S} \backslash \text{NP})$, which can apply to the intransitive verb *come*. The result of composition is an $\text{S} \backslash \text{NP}$, where the backslash-NP tells that the subject argument remains unsaturated.

(5) **The g-sl -rule** (for passing up info about unsaturated syntactic arguments)

For any expression α s.t. $\text{CAT}(\alpha) = A|B$, we have:

$$\text{CAT}(\text{g-sl}(\alpha)) = (A/C)|(B/C), \text{ and } \llbracket \text{g-sl}(\alpha) \rrbracket = \lambda V_{cb} \lambda C_c. \llbracket \alpha \rrbracket (V(C))$$

respectively. (Jacobson writes them as ‘ $A/_R B$ ’ and ‘ $A/_L B$ ’.) I also use ‘ $A \ddagger B$ ’ as the joint representation for categories corresponding to the type $\langle \text{TYPE}(B), \text{TYPE}(A) \rangle$, including A/B , $A \backslash B$, A^B , and $A|B$.

³ In the derivation graphs, dotted lines are for unary rules, such as g and z ; full lines are for binary rules, such as forward Functional Application ($>_{\text{FA}}$) and backward Functional Application ($<_{\text{FA}}$).

Further, to tackle sentences with multiple unbound pronouns, Jacobson (1999) proposes to define $g\text{-sup}$ recursively. Accordingly, applying $g\text{-sup}_n$ to α has the effect of applying $g\text{-sup}$ to α while skipping the n -many abstractions in α . The following definition also extends this assumption to $g\text{-sl}$:

- (6) **The recursive g -rules** (after Jacobson 1999)
- a. For any meaning F of type $\langle a, b \rangle$, $g_0(F) = \lambda V_{cb} \lambda C_c. F(V(C))$.
 - b. For any expression α s.t. $\text{CAT}(\alpha) = (A|B)\ddagger\vec{X}^n$, we have:
 - i. $\text{CAT}(g\text{-sup}_n(\alpha)) = (A^C|B^C)\ddagger\vec{X}^n$
 - ii. $\text{CAT}(g\text{-sl}_n(\alpha)) = ((A/C)|(B/C))\ddagger\vec{X}^n$
 - iii. $\llbracket g\text{-sup}_n(\alpha) \rrbracket = \llbracket g\text{-sl}_n(\alpha) \rrbracket = \lambda D. g_{n-1}(\llbracket \alpha \rrbracket(D))$

Consider (7) for an example of recursive $g\text{-sup}$. Obviously, a VP (of category $S \setminus \text{NP}$) must be shifted by $g\text{-sup}$ when it merges with a pronominal subject. In (7), however, the $g\text{-sup}$ rule defined in (3) doesn't apply to the VP because this VP has an extra superscript NP abstraction from the possessive pronoun *his*. Defining $g\text{-sup}$ recursively overcomes this problem: applying $g\text{-sup}_1$ means saturating this NP abstraction temporarily, applying $g\text{-sup}$, and finally abstracting the NP back.

(7) *She_i invited his_j mom.* (Unbound interpretation)

$$\frac{\frac{\text{NP}^{\text{NP}} \quad \text{she} \quad \lambda x_e . x}{\text{NP}^{\text{NP}} \quad \lambda x_e . x} \quad \frac{\frac{\frac{[g\text{-sup}(\text{invited}) \quad [his \text{ mom}]]}{(S \setminus \text{NP})^{\text{NP}} \quad \lambda y_e \lambda x_e . \text{ivt}(x, \text{mom}(y))} \quad \text{g-sup}_1}{(S^{\text{NP}} \setminus \text{NP}^{\text{NP}})^{\text{NP}} \quad \lambda y_e \lambda g_{ee} \lambda x_e . \text{ivt}(g(x), \text{mom}(y))} \quad <_{\text{FA}_1}}}{(\text{NP}^{\text{NP}})^{\text{NP}} \quad \lambda y_e \lambda x_e . \text{ivt}(g(x), \text{mom}(y))}}{\text{NP}^{\text{NP}} \quad \lambda x_e . x} \quad \text{(see steps in (4))}$$

For the same consideration, following a suggestion from Pauline Jacobson (p.c.), I assume that the $g\text{-sup}_1$ -shifted VP composes with the subject *she* via *generalized backward FA* ($<_{\text{FA}_1}$), which means applying the type-shifted VP to the subject while ignoring the superscript NP abstraction in the VP stemming from *his*.⁴

⁴ Jacobson 1999 doesn't generalize binary composition rules or define them recursively. For sentence (7), type-lifting the subject *she* via generalized Montague lift allows it to take the VP as an argument. However, in light of the idea that type-shifting rules are recursive, we can pursue a similar treatment for composition rules (Pauline Jacobson p.c.). I define generalized backward FA as in (i). Here $\lambda \vec{x}^n$ is a sequence of abstractions stems from the n -many unbound pronouns in α , and $\lambda \vec{y}^m$ is a sequence of abstractions corresponding to the m -many unsaturated syntactic arguments of α . The application of $<_{\text{FA}_{n+m}}$ to $[\beta - \alpha]$ says: apply α to β while skipping the n -many pronominal abstractions in α as well as the m -many abstractions of α that should be filled by strings appearing on the right of α . Other binary composition rules can be generalized analogously.

- (i) For any expressions α and β s.t. $\text{CAT}(\alpha) = ((A \setminus B) / \vec{Y}^m) \vec{X}^n$ and $\text{CAT}(\beta) = B$, we have:
 $\text{CAT}([\beta \alpha]) = (A / \vec{Y}^m) \vec{X}^n$, and $\llbracket \beta \alpha \rrbracket \xrightarrow{<_{\text{FA}_{n+m}}} \lambda \vec{x}^n \lambda \vec{y}^m . \llbracket \alpha \rrbracket(\vec{x}^n)(\vec{y}^m)(\llbracket \beta \rrbracket)$.

For binding, Jacobson assumes that pronoun binding is achieved by local application of the z-rule, defined in (8). Applying the z-rule to a transitive verb closes off the dependency between the two arguments of this verb. Then binding arises when the z-shifted verb merges with the binder and a string that contains the bindee.

(8) **The z-rule** (to create binding relations)

For any expression α s.t. $\text{CAT}(\alpha) = (A \setminus \text{NP})/B$, we have:

$$\text{CAT}(z(\alpha)) = (A \setminus \text{NP})/B^{\text{NP}}, \text{ and } \llbracket z(\alpha) \rrbracket = \lambda f_{eb} \lambda x_e. \llbracket \alpha \rrbracket (f(x))(x).$$

In (9), before merging with the bindee *his mom* and the binder *John*, the type-shifted verb $z(\textit{invited})$ is already encoded with a co-argument dependency. It composes with the arguments by FA, yielding an S with the meaning that John invited John's mom.

(9) *John_i invited his_i mom.* (Bound interpretation)

$$\frac{\frac{\textit{John}}{\text{NP}} \quad \frac{\frac{\frac{\textit{invited}}{(\text{S} \setminus \text{NP})/\text{NP}} \quad \lambda y_e \lambda x_e. \textit{ivt}(x, y)}{(\text{S} \setminus \text{NP})/\text{NP}^{\text{NP}} \quad \lambda f_{ee} \lambda x_e. \textit{ivt}(x, f(x))} \quad z \quad \frac{\textit{his mom}}{\text{NP}^{\text{NP}} \quad \lambda y_e. \textit{mom}(y)}}}{\text{S} \setminus \text{NP} \quad \lambda x_e. \textit{ivt}(x, \textit{mom}(x))}} >_{\text{FA}}}{\text{S} \quad \textit{ivt}(j, \textit{mom}(j))} <_{\text{FA}}$$

1.2 Problems of under-generation

Example (9) is an instance of co-argument binding. In co-argument binding, the bindee needs to be contained inside a lower argument of the z-shifted expression, and the binder needs to serve as a higher argument of this expression by itself.

(10) Schema of co-argument binding: $[[\text{BINDER}] z(\text{verb}) [\dots \text{BINDEE} \dots]]$

However, there are many cases where the binder and the bindee do not stand in a configuration like (10). First, as in (11), the bindee may occur in an adjunct, not in an argument. (Jacobson herself has an analysis for such cases, see fn. 7.)

- (11) a. Mary_i went home [with her_i mom].
b. We will sell no wine_i [before its_i time].

Second, the binder may not be an argument of a verb by itself (Barker 2005, 2012; a.o.). For example in (12a), the binder is the possessor *John*, not the entire subject *John's mother*. In the Government-and-Binding (GB) theory, such cases of binding are called 'non-c-commanding binding', including binding out of DPs, as in (12a,b), and binding out of adjuncts, as in (12c).⁵

(12) a. [John_i's/ His_i/ Every boy_i's mother] loves him_i.

⁵ I use '(non-)c-commanding' only for description. To be exact, Jacobson's system doesn't have the concept of 'c-commanding', because this system doesn't model syntax in terms of tree structures.

- b. [Someone from every city_i] despises it_i.
- c. [In everyone_i's own mind], they_i are the most important person.

This paper primarily aims for solving these binding under-generation problems in Jacobson's variable-free system. Moreover, I will enrich this system to account for the interactions between scoping and binding. Topics to be covered include scope ambiguity, inverse linking, weak crossover, and 'paycheck pronouns'.

2 Proposal: binding by the i-rule

As a general assumption on binding, I propose to replace Jacobson's z-rule with the following i-rule (a la Szabolcsi's (1992) W-combinator).⁶ The application of the i-rule has an 'identity' effect as follows: for any multi-ary predicate α that contains a bindable pronoun, whatever saturates the superscript NP abstraction from the bindable pronoun also saturates the next NP abstraction of α .

- (13) **The i-rule** (for creating dependencies)
 For any expression α s.t. $CAT(\alpha) = (A\ddagger NP)^{NP}$, we have:
 $CAT(i(\alpha)) = A\ddagger NP$, and $\llbracket i(\alpha) \rrbracket = \lambda x_e. \llbracket \alpha \rrbracket(x)(x)$.

The proposed i-rule stands in the following relation with Jacobson's z-rule:

- (14) For any transitive verb F and NP^{NP} -category f , $i(\llbracket g\text{-sup}(F) \rrbracket(f))$ is identical to $\llbracket z(F) \rrbracket(f)$ in both syntax and semantics.

Consider (15b) for illustration. First, the type-shifted verb $g\text{-sup}(invited)$ applies to *his mom* and returns a VP with two NP abstractions, one from the pronoun *his* (λy), and one from the unsaturated subject argument (λx). Second, applying i to this VP makes the two NPs be bound by the same abstractor, changing the meaning from a two-place predicate to a one-place predicate that expresses a dependency.

- (15) Ways to derive the bound interpretation of *invited his mom*:

- a. By the z-rule (the same as in (9)):

$$\frac{\frac{(S \setminus NP) / NP \quad invited}{(S \setminus NP) / NP^{NP} \quad \lambda f_{ee} \lambda x_e. ivt(x, f(x))} \quad z \quad \frac{his \ mom}{NP^{NP} \quad \lambda y_e. mom(y)}}{S \setminus NP \quad \lambda x_e. ivt(x, mom(x))} >_{FA}$$

- b. By the i-rule:

$$\frac{g\text{-sup}(invited) \ his \ mom}{(S \setminus NP)^{NP} \quad \lambda y_e \lambda x_e. ivt(x, mom(y))} \quad \text{(see steps in (4))} \quad i \quad \frac{S \setminus NP \quad \lambda x_e. ivt(x, mom(x))}{}$$

⁶ The i-rule takes insights from Szabolcsi's (1992) W-combinator. Szabolcsi builds the W-combinator into the lexicon of pronouns. Shan & Barker (2006) and Charlow (To appear) build this combinator into the quantificational antecedent.

As seen above, the z-rule is locally applied to the transitive verb *invited*, used *before* this verb merges with its arguments; in contrast, the i-rule is used *after* the verb has merged with an expression that contains the bindee.

Example (15) has shown how the i-rule accounts for **co-argument binding** — due to the equivalence in (14), the i-rule, together with the existing rule g-sup, can account for any binding relations that can be derived by the z-rule. Moreover, the i-rule can be used to account for other binding relations.

The i-rule allows for **binding into adjuncts**, because it is applied after the verb has merged with the bindee-containing expression, no matter whether this expression is an argument or an adjunct. In (16), option (16a) is deviant because the type-shifted verb $z(\textit{invited})$ selects a NP^{NP} -argument — it cannot merge with a VP-modifier (of category $(S \setminus \text{NP}) \setminus (S \setminus \text{NP})$).⁷ In contrast, in (16b), applying the i-rule to the modified VP *left with his mom* derives a dependency between the subject argument and the pronoun contained within the PP-adjunct.

- (16) John_i left [with his_i mom].
- a. * John z(left) [with his mom]
 - b. John i [left [with his mom]]
 - c. i [LIFT(John) left [with his mom]]

Or alternatively, as in (16c), when the binder *John* is type-lifted into a quantifier, which will be analyzed as a two-dimensional expression with a ‘trace’-like component (section 3.1), the dependency can be derived by applying *i* to the full sentence. This option is especially advantageous in cases where the adjuncts are sentential modifiers. For more details, see section 3.3.

The i-rule also sufficiently accounts for **pronominal binding**, regardless of ‘c-commanding’ — all we need is to apply the i-rule to an expression that contains both

⁷ Jacobson has also offered a way to derive binding into adjuncts by applying generalized Montague lift to the VP so that it can take the adjunct as an argument.

- (i) [John z(*et*-LIFT(left)) [with his mom]]

The analysis precedes as follows. First, applying generalized Montague lift *et*-LIFT to the VP to shifts this VP from $S \setminus \text{NP}$ to $(S \setminus \text{NP}) / ((S \setminus \text{NP}) \setminus (S \setminus \text{NP}))$. This operation allows the VP to take the PP-adjunct (of category $(S \setminus \text{NP}) \setminus (S \setminus \text{NP})$) as an argument. Second, applying *z* to the type-lifted VP closes off the dependency between the argument slots that will be saturated by the subject and the adjunct. Last, letting the resulting VP merge with the bindee-containing adjunct and the subject binder yields a full sentence. (See Jacobson 1999 for a related discussion tackling unbound readings.) Despite the distinct analytical moves, the analyses in (i) and (16b) converge on the idea that binding is independent of grammatical distinctions between arguments and adjuncts.

However, as pointed out by Simon Charlow (p.c.), the generalized-lifting-based analysis in (i) cannot account for cases where the binder is part of the VP, such as in *We will sell no wine_i before its_i time*. For my explanation, see section 3.3.

the bindee and the pronominal binder. (17) illustrates a case of ‘non-c-commanding’ pronominal binding, where the pronominal possessor within the subject binds the object pronoun. The z-rule cannot derive this binding relation: in (17a), z-shifting the verb *loves* yields an unwanted meaning where the entire subject binds the object.

- (17) [His_i mom] loves him_i.
 a. # [[His mom] z(loves) him] (Meaning: ‘His mom loves his mom.’ ×)
 b. [i (His mom loves him)] (Meaning: ‘His mom loves him.’ ✓)

More details of (17b) are given below. The sentence is first analyzed as an $(S^{NP})^{NP}$, in which the two NPs abstractions stem from the object pronoun *him* and the pronominal possessor *his*. This part of composition involves the applications of $g\text{-sup}_1$ and FA_1 , the same as in (7). Next, applying *i* reduces the sentence from $(S^{NP})^{NP}$ to S^{NP} and yields a dependency between *his* and *him*.

(18)

$$\frac{\frac{\text{His mom}}{NP^{NP}} \quad \lambda x_e.mom(x) \quad \frac{\text{loves him}}{(S \setminus NP)^{NP}} \quad \lambda y_e \lambda x_e.lv(x,y)}{(S^{NP} \setminus NP^{NP})^{NP}} \quad \lambda y_e \lambda f_{ee} \lambda x_e.lv(f(x),y)} \quad \begin{matrix} g\text{-sup}_1 \\ <FA_1 \end{matrix}}{\frac{(S^{NP})^{NP}}{S^{NP}} \quad \lambda y_e \lambda x_e.lv(mom(x),y)}{S^{NP}} \quad \lambda x_e.lv(mom(x),x)} \quad i$$

3 Extension: quantificational binding

Barker (2012) observes a broad range of cases where scoping feeds ‘non-c-commanding’ binding. These cases argue that quantificational binding is available only if the quantificational binder scopes above the bindee. Since Jacobson’s system doesn’t deal with scoping, the relation between scoping and binding remains a mystery.

- (19) a. [Every boy_i’s mother] loves him_i. (Possessor binding)
 b. [Someone from every city_i] despises it_i. (Inverse linking)

In GB-style compositional semantics, semantic binding is achieved by making the pronoun co-indexed with the trace of the quantifier at LF (Kratzer & Heim 1998). As in (20), the trace of the quantifier and the bound pronoun both carry the index *i*:

(20) [every-boy λi [*t_i* loves his_i mother]]

This section will propose a variable-free approach to quantificational binding that shares the wisdom in (20) but does not make use of quantifier raising or indexation.⁸ By applying the *i*-rule, the proposed approach has the effect of generating a dependency between a ‘trace’-like component of the quantifier and the pronoun.

⁸ In GB-style compositional semantics, the composition of semantics takes place at LF, an abstract level of representations in which the linear order of strings can be different from that in the surface

3.1 Quantificational DPs as two-dimensional expressions

I assume that quantificational DPs are two-dimensional expressions (cf. ‘Quantifier Storage’ in Cooper 1983), represented in the form of $\left[\begin{array}{c} \text{TOP} \\ \text{BOTTOM} \end{array} \right]$. For example, the category and meaning of *everyone* are as follows (*eo* abbreviates $\lambda P_{et}.eo(\lambda x_e.P(x))$):

$$(21) \quad \text{a. } \text{CAT}(\textit{everyone}) = \left[\frac{\text{S}[(\text{S}/\text{NP})]}{\text{NP}/\text{NP}} \right] \quad (\text{abbr.: } \left[\frac{\Pi}{T} \right]) \quad \text{b. } \llbracket \textit{everyone} \rrbracket = \left[\frac{eo}{\lambda x_e.x} \right]$$

The bottom component is ‘trace’-like.⁹ It is interpreted as an identity function over entities, just like pronouns; however, in syntax, it has a distinct category NP/NP (abbreviated as *T*), which applies to an NP argument on the right and returns an NP. The top component has the category S[(S/NP) (abbreviated as Π) and is interpreted as a generalized quantifier. I assume that the two components have distinct roles in composition. The bottom component participates in composition and binding, similar to the assumption in the GB-theory that it is the trace of the quantifier that composes with surrounding expressions and is co-indexed with the bindable pronouns. The top component deals with quantification and scoping; it remains unchanged till it can combine with the bottom component via inward FA.

The following defines the FA rules. (22a) says: when a two-dimensional expression α composes with a one-dimensional expression by FA, it is the bottom component of α that participates in the reduction by FA. (22b) is to tackle cases with multiple quantificational expressions: if both of the expressions that participate in FA are two-dimensional, the one that serves as the function takes narrow scope.

(22) **Forward FA** (backward FA is analogous)

$$\text{a. } \left(\text{A/B} \quad \left[\frac{\text{X}}{\text{B}} \right] \right) \text{ or } \left(\left[\frac{\text{X}}{\text{A/B}} \right] \quad \text{B} \right) \xrightarrow{>_{\text{FA}}} \left[\frac{\text{X}}{\text{A}} \right]$$

$$\text{b. } \left(\left[\frac{\text{X1}}{\text{A/B}} \right] \quad \left[\frac{\text{X2}}{\text{B}} \right] \right) \xrightarrow{>_{\text{FA}}} \left[\frac{\text{X2}}{\left(\left[\frac{\text{X1}}{\text{A/B}} \right] \quad \text{B} \right) >_{\text{FA}}} \right] = \left[\frac{\text{X2}}{\left[\frac{\text{X1}}{\text{A}} \right]} \right]$$

structure. In contrast, Jacobson assumes ‘Direct Compositionality’, a hypothesis independent of the variable-free hypothesis. She argues that syntax and semantics work in tandem: each linguistic expression that is proven well-formed in the syntax is assigned a meaning by the semantics, and the syntactic rules or principles which prove an expression as well-formed are paired with the semantics which assign the expression a meaning. As a result, this system has no intermediate level like LF and no transformation (e.g., LF movement) for mapping surface representations into LF representations.

⁹ To be exact, Jacobson’s framework doesn’t assume empty categories like traces in syntax. However, assuming traces doesn’t affect composition; for example, assume that a trace *t* is an NP/NP category denoting an identity function, then the VP [g-sl(*invite*) *t*], which contains an object trace, is equivalent to the transitive verb *invite*, in both syntax and semantics. In this paper, saying that an expression has a ‘trace’-like component means that this expression contains an unfilled non-subject argument, a two-dimensional quantifier, or some other NP/NP-category.

(23) defines inward FA, which applies to a single two-dimensional expression. (23a) says: in a two-dimensional expression, if the bottom component can serve as the argument of the top component, applying inward FA has the effect of applying the top to the bottom, reducing this two-dimensional expression into a one-dimensional expression. Further, (23b) says: for a recursion of top–bottom pairs, the effect of inward FA applies to the innermost top–bottom pair.

(23) **Inward FA** (a la LOWER in Shan & Barker 2006 and Barker & Shan 2014)

$$\text{a. } \left[\frac{A [B]}{B} \right] \xrightarrow{\text{VFA}} A \quad \text{b. } \left[\frac{C [A]}{\left[\frac{A [B]}{B} \right]} \right] \xrightarrow{\text{VFA}} \left[\frac{C [A]}{\left(\left[\frac{A [B]}{B} \right] \right)_{\text{VFA}}} \right] = \left[\frac{C [A]}{A} \right] \xrightarrow{\text{VFA}} C$$

When a unary rule that is defined for one-dimensional expressions (e.g., *g*, *i*, and LIFT) is applied to a two-dimensional expression, the effect of this rule applies to the bottom component of this expression, as schematized in (24). In a recursion of top–bottom pairs, (24) applies recursively; as a result, the effect of a unary rule like *g* and *i* applies to the bottom component of the innermost top–bottom pair.

(24) If \mathbb{R} is a unary rule defined for one-dimensional expressions:

$$\left[\frac{A [B]}{C} \right] \xrightarrow{\mathbb{R}} \left[\frac{A [B]}{\mathbb{R}(C)} \right]$$

3.2 (Non-)quantificational possessor binding

With the assumptions made for quantificational DPs, the analysis of pronominal possessor binding easily extends to quantificational possessor binding. In (25b), the role of the pronominal possessor in (25a) (of category NP^{NP}) is now taken by the ‘trace’-like component of the quantificational possessor (of category NP/NP).

- (25) a. His_{*i*} mom loves him_{*i*}. (Pronominal possessor binding)
 b. Every boy_{*i*}’s mom loves him_{*i*}. (Quantificational possessor binding)
 c. John_{*i*}’s mother loves him_{*i*}. (Non-quantificational possessor binding)

The composition of (25b) precedes as in (26). First, applying *g-sl* and *g-sl*₁ to *__*’s *mom* and *loves him* respectively allows them to select a *T*. Second, the ‘trace’-part of *every boy* composes with the remnant DP *__*’s *mom* and then with the VP *loves him* via backward FA, returning an (S/NP)^{NP}-category denoting a two-place predicate. In this output, the superscript NP comes from the bindee *him*, and the right-slash NP comes from the ‘trace’-part of the quantificational binder *every boy*. Third, applying *i* yields a dependency between these two NPs and shifts the bottom component of the sentence into an S/NP denoting a one-place predicate. This S/NP may serve as an argument of the quantificational part at the top. (Recall that Π abbreviates

S[(S/NP).] Finally, applying inward FA reduces the two-dimensional sentence into a simple S expressing quantificational binding: ‘For every boy x , x ’s mom loves x .’

(26) Every boy $_i$ ’s mom loves him $_i$.

$$\begin{array}{c}
 \begin{array}{c} \text{every boy} \\ \hline \left[\frac{\Pi}{T} \right] \left[\frac{eb}{\lambda_{x_e}.x} \right] \end{array} \quad \begin{array}{c} \text{'s mom} \\ \hline \text{NP}\backslash\text{NP} \quad \lambda_{x_e}.mom(x) \\ \hline T\backslash T \quad \lambda_{f_{ee}}\lambda_{x_e}.mom(f(x)) \end{array} \quad \text{g-sl} \\
 \hline
 \begin{array}{c} \left[\frac{\Pi}{T} \right] \left[\frac{eb}{\lambda_{x_e}.mom(x)} \right] \end{array} \quad \begin{array}{c} \text{loves him} \\ \hline (\text{S}\backslash\text{NP})^{\text{NP}} \quad \lambda_{y_e}\lambda_{x_e}.lv(x,y) \\ \hline ((\text{S}/\text{NP})\backslash T)^{\text{NP}} \quad \lambda_{y_e}\lambda_{g_{ee}}\lambda_{x_e}.lv(g(x),y) \end{array} \quad \text{g-sl}_1 \\
 \hline
 \begin{array}{c} \left[\frac{\Pi}{(\text{S}/\text{NP})^{\text{NP}}} \right] \left[\frac{eb}{\lambda_{y_e}\lambda_{x_e}.lv(mom(x),y)} \right] \\ \hline \left[\frac{\Pi}{\text{S}/\text{NP}} \right] \left[\frac{eb}{\lambda_{x_e}.lv(mom(x),x)} \right] \\ \hline \text{S} \quad eb(\lambda_{x_e}.lv(mom(x),x)) \end{array} \quad \begin{array}{l} \text{i} \\ \text{vFA} \end{array} \\
 \hline
 \text{S} \quad eb(\lambda_{x_e}.lv(mom(x),x)) \quad \text{vFA}
 \end{array}$$

This analysis also extends to (25c), where the binder is a simple NP. All we need is to lift the NP binder into a two-dimensional quantifier. The rule is given in (27).

(27) **Lift** (a la Montague lift)

For any expression α s.t. $\text{CAT}(\alpha) = \text{NP}$, we have:

$$\text{CAT}(\text{LIFT}(\alpha)) = \left[\frac{\text{S}[(\text{S}/\text{NP})]}{\text{NP}/\text{NP}} \right] \text{ (abbr.: } \left[\frac{\Pi}{T} \right] \text{)}, \text{ and } \llbracket \text{LIFT}(\alpha) \rrbracket = \left[\frac{\lambda_{P_{et}.P}(\llbracket \alpha \rrbracket)}{\lambda_{x_e}.x} \right].$$

3.3 More cases of binding into adjuncts

The analysis of binding into adjuncts given in (16b) works only when the binder serves as the subject and the bindee-containing adjunct serves as a VP-modifier. Hence, this analysis doesn’t apply to (28): in (28a), the binder *no wine* is inside the VP; in (28b), the *if*-clause modifies a sentence, not a VP.

- (28) a. We will sell no wine $_i$ [before its $_i$ time].
 b. [John $_i$ will be disappointed] [if his $_i$ mom is late].

The two-dimensional analysis allows us to derive dependencies in those cases by the *i*-rule. Basically, the analysis in (16b) allows the *i*-rule to be applied to a string that contains the bindee but not the binder; here, by analyzing the binder as a two-dimensional expression, we can apply the *i*-rule to a string that contains also the binder, no matter whether this string is a VP or a full sentence. For example, in (28a), applying *g*-sup to the strings that ‘*c*-command’ *its* passes up the information about this bindable pronoun; likewise, applying *g*-sl to the strings that ‘*c*-command’ the object *no wine* passes up the information about the ‘trace’-part of this quantificational

binder. Composition yields a two-dimensional sentence with an $(S/NP)^{NP}$ -category at the bottom. Further, applying i to this sentence closes off the dependency between *its* and the ‘trace’-part of *no wine*, yielding the wanted binding-into-adjunct effect.

3.4 Scope ambiguity

In GB-style compositional semantics, scope ambiguity of quantifiers is derived by quantifier raising. In the presented variable-free system, as seen in the definition of FA in (22b), the scope of a quantifier is determined by how it participates in FA:

- (29) For a sentence S of the form “Q1 V Q2”, where Q1 and Q2 are quantificational DPs, we have: (i) S has a surface scope reading if the bottom component of Q1 serves as an argument of [V Q2]; (ii) S has an inverse scope reading if Q1 as a whole serves as an argument of [V Q2].

3.4.1 Surface scoping

As in Figure 1, composing *everyone* with *loves someone* by backward FA derives the surface scope reading, where *eo* scopes above *so* in (a). This stack is reduced into a simple S by applying inward FA twice.

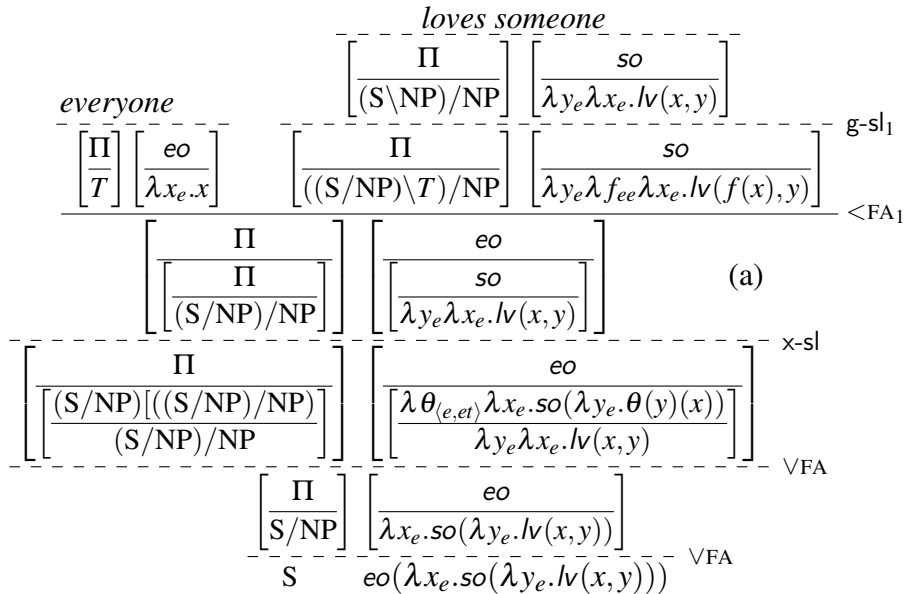


Figure 1 Surface scope reading of *Everyone loves someone*

One caveat here is that the innermost top–bottom pair in (a) cannot be reduced by inward FA: the quantificational part of *someone* should combine with an S/NP, but here the bottom component has the category (S/NP)/NP, which contains an extra NP abstraction (corresponding to λx) stemming from the ‘trace’-part of *everyone*. To solve this type-mismatch, I assume the following x-rules which apply to particular two-dimensional expressions (the subscript c in \mathbf{x}_c refers to the category C):

(30) **The x-rules**

For any two-dimensional expression $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ s.t.

$\text{CAT}(\alpha) = A[(B\ddagger X)$ and $\text{CAT}(\beta) = (B/C)\ddagger X$, we have:

$\text{x-sl}(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}) = \begin{bmatrix} \mathbf{x}_c\text{-sl}(\alpha) \\ \beta \end{bmatrix}$ and $\text{x-sup}(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}) = \begin{bmatrix} \mathbf{x}_c\text{-sup}(\alpha) \\ \beta \end{bmatrix}$, where

a. $\text{CAT}(\mathbf{x}_c\text{-sl}(\alpha)) = (A/C)[((B/C)\ddagger X)$

b. $\text{CAT}(\mathbf{x}_c\text{-sup}(\alpha)) = A^C[(B^C\ddagger X)$

c. $\llbracket \mathbf{x}_c\text{-sl}(\alpha) \rrbracket = \llbracket \mathbf{x}_c\text{-sup}(\alpha) \rrbracket = \lambda V_{\langle x,cb \rangle} \lambda C_c. \llbracket \alpha \rrbracket (\lambda X_x. V(X)(C))$

In Figure 1, applying x-sl after (a) converts the quantificational part of *someone* into a complex category that selects an (S/NP)/NP argument. It also transmits the information about the ‘trace’ of *everyone* to the entire existential quantification, which further forms the bottom component of the universal quantification. After applying x-sl, inward FA can proceed as normal.

3.4.2 Inverse scoping

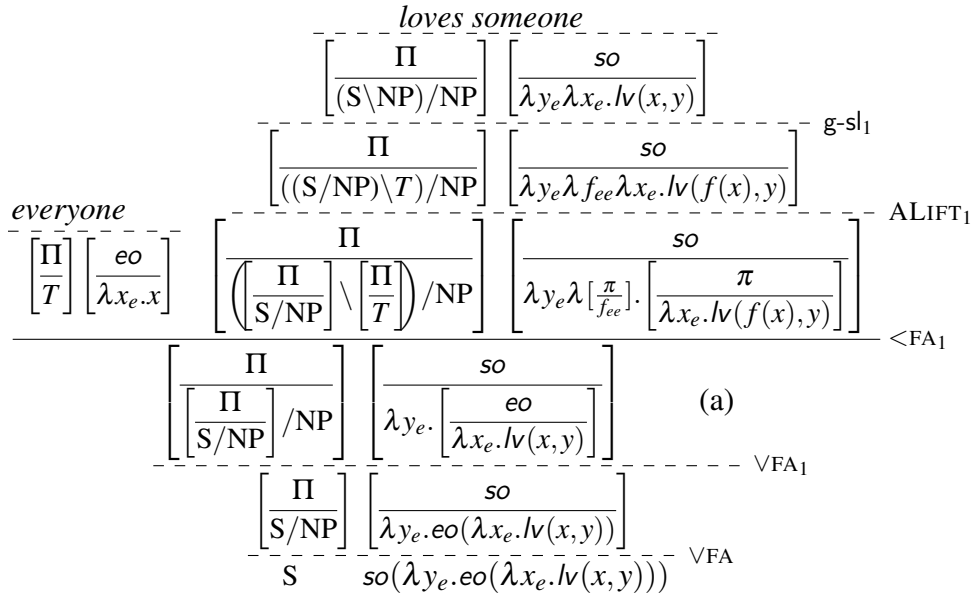
As mentioned, in “Q1 V Q2”, an inverse scope reading arises if it is the entire Q1 that participates in the composition as an argument of [V Q2]. A la Hendriks 1993, this condition can be achieved by *argument-lift*. In my system, argument-lift means lifting a T -category argument into a two-dimensional quantifier, defined as below:

(31) **Argument lift** (ALIFT; a la Hendriks 1993)

For any expression α s.t. $\text{CAT}(\alpha) = (X/NP)|T$, we have:

$\text{CAT}(\text{ALIFT}(\alpha)) = \left[\frac{\Pi}{X/NP} \right] | \left[\frac{\Pi}{T} \right]$ and $\llbracket \text{ALIFT}(\alpha) \rrbracket = \lambda \left[\frac{\pi}{f_{ee}} \right]. \left[\frac{\pi}{\lambda x_e. \llbracket \alpha \rrbracket (f(x))} \right]$

In Figure 2 (next page), first, by applying ALift_1 , the subject argument slot which used to call for a T now wants an argument that is a two-dimensional quantifier. Next, composing this argument-lifted VP with the subject *everyone* by backward FA yields an inverse scope reading — so scopes above eo in (a). Finally, applying inward FA twice reduces the stack into a simple S. All the rules, except the last inward FA, carry the subscript 1, since they need to skip the extra NP abstraction (corresponding to λy) stemming from the ‘trace’-part of the object quantifier *someone*.


Figure 2 Inverse scope reading of *Everyone loves someone*

3.5 Inverse linking

Compared to the basic cases of inverse scoping, further complications arise in sentences with inverse linking. For example, the composition of (32) should involve three operations: (i) restrictor modification, to let the PP *from every city* restrict the quantification domain of *someone*; (ii) inverse scoping: to make *every* scope above *some*, and (iii) binding: to make *every city* bind a non-c-commanding pronoun *it*. Previous sections have explained inverse scoping and non-c-commanding quantificational binding by using ALIFT and *i*, respectively. This subsection adds one more ingredient to derive restrictor modification.

(32) [Someone from every city]_{*i*} [despises it]_{*i*}].

Given that it is the ‘trace’ of *someone*, which is a *T* denoting an identity function, that participates in composition, I propose a restrictor modification rule as follows. This rule converts a predicative modifier into a $T \setminus T$ -category, which functions as a restrictor of *T*-categories. ($f' \leq f$ means that every pair in f' is a pair in f .)

(33) **The RES-rule** (for restrictor modification)

For any P s.t. $\text{CAT}(P) = S \setminus NP$, we have: $\text{CAT}(\text{RES}(P)) = T \setminus T$, and
 $\llbracket \text{RES}(P) \rrbracket = \lambda f_{ee} . \text{MAX}_{f' \leq f} [\forall x' \in \text{Dom}(f') [\llbracket P \rrbracket (f'(x'))]]$

(Read as: For any $f_{\langle e,e \rangle}$, return the maximal function f' that is a subpart of f' s.t. f' maps each element in its domain to something that P holds for.)

The complex subject is composed as in Figure 3. Applying the restriction rule to covert the PP *from every city* into a restrictor of skolem functions. This restrictor undergoes argument-lift, so that it can apply to the entire two-dimensional expression *someone* and derive an inverse scope reading (*every city* \gg *someone*).

Step (A): Compose the complex quantificational subject

$$\begin{array}{c}
 \text{from every city} \\
 \hline
 \left[\frac{\Pi}{(S \setminus NP) / NP} \right] \left[\frac{ec}{\lambda y_e \lambda x_e . fr(x, y)} \right] \\
 \hline
 \left[\frac{\Pi}{(T \setminus T) / NP} \right] \left[\frac{ec}{\lambda y_e \lambda f_{ee} . \text{MAX}_{f' \leq f} [\forall x' \in \text{Dom}(f') [fr(f'(x'), y)]]} \right] \text{RES}_1 \\
 \text{someone} \\
 \left[\frac{\Pi}{T} \right] \left[\frac{so}{\lambda x_e . x} \right] \left[\frac{\Pi}{\left(\left[\frac{\Pi}{T} \right] \setminus \left[\frac{\Pi}{T} \right] \right) / NP} \right] \left[\frac{ec}{\lambda y_e \lambda \left[\frac{\pi}{f} \right] . \left[\text{MAX}_{f' \leq f} [\forall x' \in \text{Dom}(f') [fr(f'(x'), y)]] \right]} \right] \text{ALIFT}_1 \\
 \hline
 \left[\frac{\Pi}{\left[\frac{\Pi}{T} \right] / NP} \right] \left[\frac{ec}{\lambda y_e . \left[\text{MAX}_{f' \leq (\lambda x_e . x)} [\forall x' \in \text{Dom}(f') [fr(f'(x'), y)]] \right]} \right] < \text{FA}_1
 \end{array}$$

Figure 3 Composition of *someone from every city* (inverse scope reading)

In the output of the above composition, the T -category part at the very bottom is read as ‘the maximal $\langle e, e \rangle$ -type identity function f' that maps each entity in its domain to an entity from y ’. This f' is extensionally equivalent to $\lambda x_e : fr(x, y) . x$, namely, the identity function that is only defined for entities from y .

The modified quantificational subject has a complex category. As illustrated in Figure 4: Step (B), to compose with this subject, the VP has to undertake a few of type-shifting operations till it selects an argument of category $\left[\frac{\Pi}{T} \right] / NP$. In particular, the first $g\text{-sl}_1$ and ALIFT are applied to allow the VP to select a two-dimensional quantificational argument, and the second $g\text{-sl}_1$ is used to pass up the information about the ‘trace’ of *every city*.

The final steps of composition are given in Figure 4: Step (C). First, the complex subject composes with the type-shifted VP via backward FA. Next, applying the i -rule yields a dependency between the ‘trace’ of *every city* and the object pronoun *it* (corresponding to λy and λz , respectively). Last, applying inward FA twice reduces the stack into a simple S expressing inverse linking.

Step (B): Type-shift the VP

$$\begin{array}{c}
 \text{despises it} \\
 \hline
 (S \setminus NP)^{NP} \quad \lambda_{z_e} \lambda_{x_e} . \text{desp}(x, z) \\
 \hline
 ((S/NP) \setminus T)^{NP} \quad \lambda_{z_e} \lambda_{f_{ee}} \lambda_{x_e} . \text{desp}(f(x), z) \quad \text{g-sl}_1 \\
 \hline
 \left(\left[\frac{\Pi}{S/NP} \right] \setminus \left[\frac{\Pi}{T} \right] \right)^{NP} \quad \lambda_{z_e} \lambda_{\left[\frac{\pi}{f} \right]} . \left[\frac{\pi}{\lambda_{x_e} . \text{desp}(f(x), z)} \right] \quad \text{ALIFT}_1 \\
 \hline
 \left(\left(\left[\frac{\Pi}{S/NP} \right] / NP \right) \setminus \left(\left[\frac{\Pi}{T} \right] / NP \right) \right)^{NP} \quad \lambda_{z_e} \lambda_{\rho_{\langle e, \left[\frac{et, t \right]}{ee} \rangle}} \lambda_{y_e} . \left[\lambda_{\left[\frac{\pi}{f} \right]} . \left[\frac{\pi}{\lambda_{x_e} . \text{desp}(f(x), z)} \right] \right] (\rho(y)) \quad \text{g-sl}_1
 \end{array}$$

Step (C): Combine the complex subject with the VP

$$\begin{array}{c}
 \text{(A) someone from every city} \quad \text{(B) despises it} \\
 \hline
 \left[\frac{\Pi}{\left(\left[\frac{\Pi}{S/NP} \right] / NP \right)^{NP}} \right] \left[\frac{ec}{\lambda_{z_e} \lambda_{y_e} . \left[\frac{so}{\lambda_{x_e} . \text{desp}((\text{MAX}_{f' \leq (\lambda_{x_e, x})} [\forall x' \in \text{Dom}(f') [fr(f'(x'), y)])]) (x), z)} \right]} \right] \quad \text{<FA}_1 \\
 \hline
 \left[\frac{\Pi}{\left(\left[\frac{\Pi}{S/NP} \right] / NP \right)^{NP}} \right] \left[\frac{ec}{\lambda_{y_e} . \left[\frac{so}{\lambda_{x_e} . \text{desp}((\text{MAX}_{f' \leq (\lambda_{x_e, x})} [\forall x' \in \text{Dom}(f') [fr(f'(x'), y)])]) (x), y)} \right]} \right] \quad i \\
 \hline
 \left[\frac{\Pi}{S/NP} \right] \left[\frac{ec}{\lambda_{y_e} . so(\lambda_{x_e} . \text{desp}((\text{MAX}_{f' \leq (\lambda_{x_e, x})} [\forall x' \in \text{Dom}(f') [fr(f'(x'), y)])]) (x), y))} \right] \quad \text{VFA}_1 \\
 \hline
 \left[\frac{\Pi}{S/NP} \right] \left[\frac{ec}{\lambda_{y_e} . so(\lambda_{x_e} . fr(x, y) \wedge \text{desp}(x, y))} \right] \quad \text{equivalent} \\
 \hline
 S \quad ec(\lambda_{y_e} . so(\lambda_{x_e} . fr(x, y) \wedge \text{desp}(x, y))) \quad \text{VFA}
 \end{array}$$

Figure 4 Composition of *Someone from every city despises it* (inverse linking)

4 Weak crossover

Scoping doesn't feed binding in crossover constructions. (34) illustrates a case of weak crossover: the pronominal possessor *his* in the subject cannot be bound by the quantificational object *everyone*.

- (34) His_{j/*i} mom invited everyone_i.
 (Unavailable reading: 'For everyone x , x 's mom invited x .')

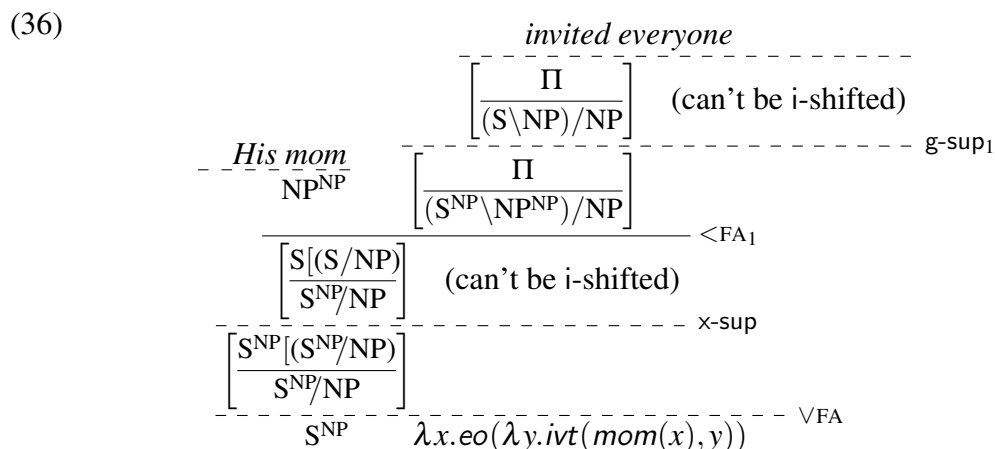
In my analysis, the deviance of weak crossover is straightforwardly predicted by the definedness constraint on the syntax of the *i*-rule. As defined in (13), the *i*-rule only applies to categories of the form ' $(X \ddagger NP)^{NP}$ '. According to this constraint, the

superscript NP abstraction provided by a bindable pronoun has to be reduced first, before the other NP abstraction saved for the binder of this pronoun is reduced.

For instance, the *i*-rule can be applied to $(S \setminus NP)^{NP}$, $(S/NP)^{NP}$, and $(S^{NP})^{NP}$, as exemplified in (36a–c). In these three cases, the NP abstraction for the binder comes from an unsaturated syntactic argument (viz., the subject argument of *loves*), the ‘trace’-part of a two-dimensional quantifier (viz., the bottom component of *every boy*), and a pronominal expression (viz., *his*), respectively.

- (35) a. *i* [*loves him*]
 b. *i* [*every boy’s mom loves him*]
 c. *i* [*his mom loves him*]

In contrast, the *i*-rule cannot be applied in (34), neither to VP nor to the full sentence. Both strings have an ‘ $(X \ddagger NP)/NP$ ’-category at the bottom. The composition in (36) yields an S^{NP} , which expresses an unbound reading.



5 Paycheck pronouns

The following sentences involve a ‘paycheck pronoun’. In (37a), the pronoun *her* refers to Billy’s mom. Descriptively, this interpretation arises as follows: *her* refers to the function *his mom*, in which the possessive pronoun *his* is bound by *Billy*. The same applies to the pronoun *it* in (37b).

- (37) a. Andy_{*i*} loves his_{*i*} mom, but Billy_{*j*} hates **her**_{=his_{*j*} mom}.
 (Intended: ‘Andy loves his mom, but Billy hates his mom.’)
 b. The woman [who_{*i*} deposited her_{*i*} paycheck in the bank] was wiser than
 the woman [who_{*j*} deposited **it**_{=her_{*j*} paycheck} in the Credit Union].

Jacobson’s variable-free semantics naturally accounts for the functional interpretations of paycheck pronouns: first, *her* in lexicon is an NP^{NP} -category denoting

an identify function over individuals; second, applying g-sup to *her* turns it into an $(\text{NP}^{\text{NP}})^{(\text{NP}^{\text{NP}})}$ -category that denotes an identity function over $\langle e, e \rangle$ -type functions, such as $\llbracket \text{his mom} \rrbracket (= \lambda x_e. \text{mom}(x))$. This merit of Jacobson’s system is also manifested in analyzing Bach-Peters sentences (Jacobson 2000) as well as *wh*-questions and *wh*- relative clauses with functional interpretations (Jacobson 1999).

However, Jacobson’s system doesn’t generate non-c-commanding binding, which includes not only aforementioned possessor binding but also cross-sentential binding. Jacobson (2000) argues that in (37a) the dependency between *his mom* and the paycheck pronoun *her* isn’t really an instance of binding; instead, she treats *her* as a free pronoun, which picks up the contextually salient value denoted by *his mom*.

In contrast, the proposed system allows us to derive a non-c-commanding dependency between *his mom* and *her*. By lifting NP^{NP} -categories into two-dimensional generalized quantifiers, the analysis of quantificational binding also extends to the binding of a paycheck pronoun. I generalize the Montague lift rule as in (38), which works cross-categorically for NPs as well any pronominal categories.

(38) **Cross-categorical lift** (generalized from (27))

For any X-category α s.t. X is NP or a pronominal category, we have:

$$\text{CAT}(\text{LIFT}(\alpha)) = \left[\frac{\text{S}[(\text{S}/\text{X})]}{\text{X}/\text{X}} \right], \text{ and } \llbracket \text{LIFT}(\alpha) \rrbracket = \left[\frac{\lambda P_{\langle x,t \rangle}. P(\llbracket \alpha \rrbracket)}{\lambda X_x.X} \right]$$

(39) **Pronominal categories** (after Charlow To appear)

NP^{NP} is a pronominal category; if Y^{X} is a pronominal category, $(\text{Y}^{\text{NP}})^{(\text{X}^{\text{NP}})}$ is a pronominal category; nothing else is a pronominal category.

The composition of (37a) is given in (40), which omits the g-rules applied for resolving type-mismatch. This composition includes the following operations: (i) as assumed by Jacobson (2000), applying g-sup to the paycheck pronoun *her* allows it to refer to an $\langle e, e \rangle$ -type function such as *his mom* (as opposed to an individual); (ii) in both clauses, applying i to VP yields a dependency between the subject and the pronominal object in each clause; (iii) together with the results of operation (i), lifting *his mom* and applying i to the entire coordination make *her* be bound by *his mom*. This analysis also applies to (37b).

$$(40) \quad \frac{\frac{\frac{\llbracket \text{Andy i [loves LIFT(his mom)]} \rrbracket}{\left[\frac{\text{S}[(\text{S}/\text{NP}^{\text{NP}})]}{\text{S}/\text{NP}^{\text{NP}}} \right]}{\left[\frac{\lambda P_{\langle ee,t \rangle}. P(\lambda x_e. \text{mom}(x))}{\lambda f_{ee}. \text{lv}(a, f(a))} \right]}{\left[\frac{\text{S}[(\text{S}/\text{NP}^{\text{NP}})}{(\text{S}/\text{NP}^{\text{NP}})^{(\text{NP}^{\text{NP}})}]}{\left[\frac{\lambda P_{\langle ee,t \rangle}. P(\lambda x_e. \text{mom}(x))}{\lambda g_{ee} \lambda f_{ee}. \text{lv}(a, f(a)) \wedge \text{hate}(b, g(b))} \right]} \right]}{\text{S}^{(\text{NP}^{\text{NP}})} \quad \lambda g_{ee}. \text{hate}(b, g(b))} \quad \text{g, FA}}{\frac{\frac{\left[\frac{\text{S}[(\text{S}/\text{NP}^{\text{NP}})}{(\text{S}/\text{NP}^{\text{NP}})^{(\text{NP}^{\text{NP}})}]}{\left[\frac{\lambda P_{\langle ee,t \rangle}. P(\lambda x_e. \text{mom}(x))}{\lambda f_{ee}. \text{lv}(a, f(a)) \wedge \text{hate}(b, f(b))} \right]} \right]}{\text{S}} \quad \text{i}}{\text{lv}(a, \text{mom}(a)) \wedge \text{hate}(b, \text{mom}(b))} \quad \vee \text{FA}} \quad \text{i}$$

6 Comparison with Barker (2005)

To account for ‘non-c-commanding’ binding, Barker (2005) enriches the variable-free system of Jacobson 1999 by assuming a separate pair of rules, called s and q. As exemplified in (42), the s-rule, defined as a dual of z, allows a quantificational binder to bind one level up. Composition with quantifiers is achieved by value raising (VR) and argument lift (ALIFT). Further, in cases where the binder is deeply embedded, the q-rule (cf. g) passes up the information about this binder (details omitted).

(41) The s-rule

For any expression α s.t. $\text{CAT}(\alpha) = (A|B)|\text{NP}$, we have:

$\text{CAT}(s(\alpha)) = (A|B^{\text{NP}})|\text{NP}$, and $\llbracket s(\alpha) \rrbracket = \lambda x_e \lambda f_{eb} \cdot \llbracket \alpha \rrbracket(x)(f(x))$

(42) Every boy_i’s mom loves him_i. (*G* abbreviates $S/(S \setminus \text{NP})$)

$$\begin{array}{c}
 \begin{array}{c}
 \text{'}s\ mom \\
 \text{NP} \setminus \text{NP} \quad \lambda x. \text{mom}(x) \\
 \text{(S/(S \setminus NP))} \setminus \text{NP} \quad \lambda x \lambda \pi. \pi(\lambda P.P(\text{mom}(x))) \quad \text{VR} \\
 \text{S} \\
 \text{every boy} \quad \text{(S/(S \setminus NP)}^{\text{NP}}) \setminus \text{NP} \quad \dots \\
 \text{G} \quad \text{eb} \quad \text{(S/(S \setminus NP)}^{\text{NP}}) \setminus \text{G} \quad \dots \quad \text{ALIFT} \\
 \text{S/(S \setminus NP)}^{\text{NP}} \quad \lambda R. \text{eb}(\lambda x. R(x)(\text{mom}(x))) \quad <_{\text{FA}} \\
 \text{S} \quad \text{eb}(\lambda x. \text{lv}(\text{mom}(x), x)) \quad \text{loves him} \\
 \text{(S \setminus NP)}^{\text{NP}} \quad \lambda y \lambda x. \text{lv}(x, y) \quad <_{\text{FA}}
 \end{array}
 \end{array}$$

However, the s-rule is abandoned in Jacobson 1999 because it allows for crossover. To avoid generating crossover constructions, Barker builds a value assigning and transmitting system, which however largely increases the complexity of the system.

In comparison, while Barker assumes two binding rules z and s, my analysis uses only one binding rule, namely, the i-rule. Crossover constructions are automatically ruled out by a syntactic constraint on the application of i. Further, by defining quantificational binders as two-dimensional expressions with a ‘trace’-like component of the NP/NP-category, I argue that the information about the binder is transmitted by the Geach rule g-sl, which has already been assumed in Jacobson’s system.

7 Conclusion

This paper presented a solution to the binding under-generation problems of Jacobson’s variable-free system. I argued to derive dependencies uniformly by the i-rule, which is more flexible than Jacobson’s locally applied z-rule. I also argued to analyze quantifiers as two-dimensional expressions, which consist of a quantificational part and a ‘trace’ part. This two-dimensional treatment is particularly useful to deal with scoping and ‘non-c-commanding’ binding. In future research, I would like to see how the enriched system connects to and differs from competing variable-free systems (Szabolcsi 1992; Barker & Shan 2014; Charlow To appear; a.o.).

References

- Barker, Chris. 2005. Remark on Jacobson 1999: Crossover as a local constraint. *Linguistics and Philosophy* 28(4). 447–472.
- Barker, Chris. 2012. Quantificational binding does not require c-command. *Linguistic inquiry* 43(4). 614–633.
- Barker, Chris & Chung-chieh Shan. 2014. *Continuations and natural language*, vol. 53. Oxford: Oxford University Press.
- Charlow, Simon. To appear. Variable-free semantics and flexible grammars for anaphora. *Studies in Linguistics and Philosophy* 100. <https://ling.auf.net/lingbuzz/004503>.
- Cooper, Robin. 1983. Quantifier storage. In *Quantification and Syntactic Theory*, 52–78. Springer.
- Geach, Peter Thomas. 1970. A program for syntax. *Synthese* 3–17.
- Hendriks, Herman. 1993. *Studied Flexibility: Categories and Types in Syntax and Semantics*: Institute for Logic, Language and Computation, University van Amsterdam PhD dissertation.
- Jacobson, Pauline. 1999. Towards a variable-free semantics. *Linguistics and Philosophy* 22(2). 117–185.
- Jacobson, Pauline. 2000. Paycheck pronouns, Bach-Peters sentences, and variable-free semantics. *Natural Language Semantics* 8(2). 77–155.
- Jacobson, Pauline. 2014. *Compositional semantics: An introduction to the syntax/semantics interface*. Oxford: Oxford University Press.
- Kratzer, Angelika & Irene Heim. 1998. *Semantics in generative grammar*. Oxford: Wiley-Blackwell.
- Shan, Chung-chieh & Chris Barker. 2006. Explaining crossover and superiority as left-to-right evaluation. *Linguistics and Philosophy* 29(1). 91–134.
- Szabolcsi, Anna. 1992. Combinatory grammar and projection from the lexicon. In Ivan Sag & Anna Szabolcsi (eds.), *Lexical matters*, 241–268. CSLI publications Stanford.

Yimei Xiang
18 Seminary Pl, Room 205
Department of Linguistics
Rutgers University
New Brunswick, NJ 08901
yimei.xiang@rutgers.edu