Proceedings of SALT 31: 000–000, 2021

# 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 a locally applied rule called z. 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 into variables and interpret these variables via assignment functions (as in Kratzer & Heim 1998), 'variable-free' approaches analyze pronouns as instances of the category $\text{NP}^{\text{NP}}$ that denote identity functions over entities, as exemplified in (1).

(1) $\textsc{cat}(it) = \text{NP}^{\text{NP}}$, and $[\![it]\!] = \lambda x_e.x$

More generally, as exemplified in (2-ii), pronouns and expressions that contain an unbound pronoun are referred to as '$\text{X}^{\text{NP}}$-categories', represented with a 'sup(erscript)-NP abstraction'. $\text{X}^{\text{NP}}$ has the same semantics as the functional counterpart $\text{X}|\text{NP}$, which has a 'slash-NP abstraction'.[2] However, in contrast to $\text{X}|\text{NP}$,

* 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 $A\backslash B$, which select a $B$ argument on the right and on

$X^{NP}$ doesn't actively merge with an NP in syntax. For example, *their mom* has the same semantics as *__'s mom*, but it doesn't take an NP argument.

(2)

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

To compose expressions with pronouns, Jacobson assumes a type-shifting rule g-sup(script), defined as in (3). Applying g-sup to a functional expression $\alpha$ that selects an X argument allows this $\alpha$ to merge with an $X^{NP}$ argument, and further, 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 $\alpha$ s.t. $\textsc{cat}(\alpha) = A|B$, we have:
$\textsc{cat}(\text{g-sup}(\alpha)) = A^C|B^C$, and $[\![\text{g-sup}(\alpha)]\!] = \lambda V_{cb}\lambda C_c.[\![\alpha]\!](V(C))$

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

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

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

The g-sup rule is analogous to the so-called 'Geach rule' (Geach 1970), referred to as 'g-sl(ash)' in Jacobson's system. These two g-rules have the same effects in semantics, but g-sl is used to pass up information about an unsaturated syntactic argument. For example, in *didn't come*, applying g-sl to the negative auxiliary *didn't* shifts it from an S/S to an (S\NP)/(S\NP), which can merge with the intransitive verb *come*. The result of composition is an S\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 $\alpha$ s.t. $\textsc{cat}(\alpha) = A|B$, we have:
$\textsc{cat}(\text{g-sl}(\alpha)) = (A/C)|(B/C)$, and $[\![\text{g-sl}(\alpha)]\!] = \lambda V_{cb}\lambda C_c.[\![\alpha]\!](V(C))$

the left, respectively, and return an $A$. (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 $\langle\textsc{type}(B),\textsc{type}(A)\rangle$, 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 (>FA) and backward Functional Application (<FA).

Further, to tackle sentences with multiple unbound pronouns, Jacobson (1999) proposes to define g-sup recursively. Applying g-sup$_n$ to a functional expression $\alpha$ has the effect of applying g-sup to $\alpha$ while skipping $n$-many abstractions in $\alpha$. The following definition also extends this idea to g-sl:

(6) **The recursive g-rules** (after Jacobson 1999)

a. For any meaning $F$ of type $\langle a,b\rangle$, $\mathsf{g}_0(F) = \lambda V_{cb}\lambda C_c.F(V(C))$.

b. For any expression $\alpha$ s.t. $\textsc{cat}(\alpha) = (A|B)\ddagger\vec{X}^n$, we have:

i. $\textsc{cat}(\text{g-sup}_n(\alpha)) = (A^C|B^C)\ddagger\vec{X}^n$

ii. $\textsc{cat}(\text{g-sl}_n(\alpha)) = ((A/C)|(B/C))\ddagger\vec{X}^n$

iii. $[\![\text{g-sup}_n(\alpha)]\!] = [\![\text{g-sl}_n(\alpha)]\!] = \lambda D.\mathsf{g}_{n-1}([\![\alpha]\!](D))$

Derivation (7) exemplifies recursive g-sup. As noted, a VP (of category S\NP) needs to be shifted by g-sup when it merges with a pronominal subject. The simple g-sup rule defined in (3), however, doesn't apply to the VP in (7), because this VP has an extra sup-NP abstraction from the possessive pronoun *his*. Defining g-sup recursively overcomes this problem: applying g-sup$_1$ means saturating the extra NP abstraction temporarily, applying g-sup, and finally abstracting the NP back.

(7) *She$_i$ invited his$_j$ mom.* (Unbound interpretation)

$$\dfrac{\dfrac{\textit{she}}{\mathrm{NP^{NP}}\quad \lambda x_e.x} \qquad \dfrac{\dfrac{[\text{g-sup}(\textit{invited})\ [\textit{his mom}]]}{(\mathrm{S\backslash NP})^{\mathrm{NP}}\quad \lambda y_e\lambda x_e.ivt(x, mom(y))}\ \text{(see steps in (4))}}{(\mathrm{S^{NP}\backslash NP^{NP}})^{\mathrm{NP}}\quad \lambda y_e\lambda g_{ee}\lambda x_e.ivt(g(x), mom(y))}\ \text{g-sup}_1}{(\mathrm{NP^{NP}})^{\mathrm{NP}}\quad \lambda y_e\lambda x_e.ivt(g(x), mom(y))}\ {<}\textsc{fa}_1$$

Analogously, following a suggestion from Pauline Jacobson (p.c.), I assume that the g-sup$_1$-shifted VP composes with the subject argument via 'generalized' backward FA ($<$ FA$_1$), which has the effect of applying g-sup$_1$-shifted VP to the subject while ignoring the sup-NP abstraction in the VP stemming from *his*.[4]

4 Jacobson 1999 doesn't generalize binary composition rules or define them recursively. She composes (7) by generalized lift: 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 $\lambda\vec{x}^n$ denotes a sequence of abstractions stemming from the $n$-many unbound pronouns in $\alpha$, and $\lambda\vec{y}^m$ denotes a sequence of abstractions corresponding to the $m$-many unsaturated syntactic arguments of $\alpha$. The application of $<$FA$_{n+m}$ to $[\beta\text{-}\alpha]$ says: apply $\alpha$ to $\beta$ while skipping the $n$-many pronominal abstractions in $\alpha$ as well as the $m$-many abstractions of $\alpha$ that should be filled by strings appearing on the right of $\alpha$. Other binary composition rules can be generalized analogously.

(i) For any expressions $\alpha$ and $\beta$ s.t. $\textsc{cat}(\alpha) = ((A\backslash B)/\vec{Y}^m)^{\vec{X}^n}$ and $\textsc{cat}(\beta) = B$, we have: $\textsc{cat}([\beta\text{-}\alpha]) = (A/\vec{Y}^m)^{\vec{X}^n}$, and $[\![\beta\text{-}\alpha]\!] \xrightarrow[<\textsc{fa}_{n+m}]{} \lambda\vec{x}^n\lambda\vec{y}^m.[\![\alpha]\!](\vec{x}^n)(\vec{y}^m)([\![\beta]\!])$.

For binding, Jacobson assumes that pronoun binding is achieved by the locally applying the rule 'z', defined as 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 $\alpha$ s.t. $\textsc{cat}(\alpha) = (A\backslash\text{NP})/B$, we have:
$\textsc{cat}(\mathsf{z}(\alpha)) = (A\backslash\text{NP})/B^{\text{NP}}$, and $[\![\mathsf{z}(\alpha)]\!] = \lambda f_{eb}\lambda x_e.[\![\alpha]\!](f(x))(x)$.

In (9), before merging with its arguments, the type-shifted verb z(*invited*) is already encoded with a co-argument dependency. It composes with the two arguments by FA, yielding an S with the bound interpretation that John invited John's mom.

(9) *John$_i$ invited his$_i$ mom.* (Bound interpretation)

$$\dfrac{\dfrac{\mathit{John}}{\text{NP}\quad j} \qquad \dfrac{\dfrac{\dfrac{\mathit{invited}}{(\text{S}\backslash\text{NP})/\text{NP}\quad \lambda y_e\lambda x_e.\mathit{ivt}(x,y)}}{(\text{S}\backslash\text{NP})/\text{NP}^{\text{NP}}\quad \lambda f_{ee}\lambda x_e.\mathit{ivt}(x,f(x))}\,\mathsf{z} \qquad \dfrac{\mathit{his\ mom}}{\text{NP}^{\text{NP}}\quad \lambda y_e.\mathit{mom}(y)}}{\text{S}\backslash\text{NP}\quad \lambda x_e.\mathit{ivt}(x,\mathit{mom}(x))}\,{>}\text{FA}}{\text{S}\quad \mathit{ivt}(j,\mathit{mom}(j))}\,{<}\text{FA}$$

## 1.2 Problems of under-generation

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

(10) Schema of co-argument binding: [[BINDER] z(verb) [... BINDEE ...]]

However, there are many cases where the bindee and the binder do not stand in a configuration like (10). First, as seen in (11), the bindee may occur in an adjunct, not an argument. (For Jacobson's analysis of 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 itself may not be an argument of a functional expression (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', which include 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 aims to solve the binding under-generation problems while keeping the basic apparatus of Jacobson's 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

For binding in general, I propose to replace Jacobson's z-rule with the i-rule defined in (13) (a la Szabolcsi's (1992) W-combinator).[6] The i-rule has an 'identity' effect: for any multi-ary predicate $\alpha$ that contains a bindable pronoun, applying i to $\alpha$ has the consequence that whatever saturates the sup-NP abstraction from the bindable pronoun contained in $\alpha$ also saturates the next NP abstraction of $\alpha$.

(13) **The i-rule** (for deriving dependencies)
For any expression $\alpha$ s.t. $\textsc{cat}(\alpha) = (A\ddagger\text{NP})^{\text{NP}}$, we have:
$\textsc{cat}(\mathsf{i}(\alpha)) = A\ddagger\text{NP}$, and $[\![\mathsf{i}(\alpha)]\!] = \lambda x_e.[\![\alpha]\!](x)(x)$.

The proposed i-rule and Jacobson's z-rule stand in the following relation:

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

Consider (15b) for illustration. First, the type-shifted verb g-sup(*invited*) applies to *his mom* and returns a VP with two NP abstractions, one from the pronoun *his* ($\lambda y$), and one from the unsaturated subject ($\lambda x$). Second, applying i to this VP makes the two NPs abstractions 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 (as in (9)):

$$\dfrac{\dfrac{(\text{S}\backslash\text{NP})/\text{NP} \quad \textit{invited}}{(\text{S}\backslash\text{NP})/\text{NP}^{\text{NP}} \quad \lambda f_{ee}\lambda x_e.\mathit{ivt}(x, f(x))}\,\mathsf{z} \qquad \dfrac{\textit{his mom}}{\text{NP}^{\text{NP}} \quad \lambda y_e.\mathit{mom}(y)}}{\text{S}\backslash\text{NP} \quad \lambda x_e.\mathit{ivt}(x, \mathit{mom}(x))}\,>\!\text{FA}$$

b. By the i-rule:

$$\dfrac{\dfrac{[\mathsf{g\text{-}sup}(\textit{invited})\ [\textit{his mom}]]}{(\text{S}\backslash\text{NP})^{\text{NP}} \quad \lambda y_e\lambda x_e.\mathit{ivt}(x, \mathit{mom}(y))}\,\text{(see steps in (4))}}{\text{S}\backslash\text{NP} \quad \lambda x_e.\mathit{ivt}(x, \mathit{mom}(x))}\,\mathsf{i}$$

6 The i-rule is inspired by Szabolcsi's (1992) W-combinator. Szabolcsi builds the W-combinator into the lexicons of pronouns, and Shan & Barker (2006) and Charlow (To appear) build it into the quantificational antecedent.

As seen above, the z-rule is locally applied to the transitive verb *invited*; in contrast, the i-rule is applied after the verb has merged with a bindee-containing expression.

Example (15) has shown how the i-rule accounts for **co-argument binding** — due to the equivalence relation in (14), the proposed i-rule, together with the existing rule g-sup, can account for any binding relations that can be derived by the z-rule. Moreover, as I will argue below, the i-rule captures more 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(*invited*) selects an $NP^{NP}$-argument — it cannot merge with a VP-modifier (of category $(S\backslash NP)\backslash(S\backslash NP)$).[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]]

Option (16c) will be discussed in section 3.3: by type-lifting the binder *John* into a quantifier, which will be analyzed as a two-dimensional expression with a 'trace'-like component, we can derive the dependency by applying i to the entire sentence.

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 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 capture this binding relation: in (17a), z-shifting the verb *loves* would make the object be bound by the entire subject.

7 Jacobson derives binding into adjuncts by applying generalized Montague lift (*et*-LIFT) to the VP, which allows this VP to take a VP-adjunct as an argument. The analysis precedes as follows:

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

First, applying *et*-LIFT to the VP *left* shifts it into an $(S\backslash NP)/((S\backslash NP)\backslash(S\backslash NP))$, which can take the PP-adjunct (of category $(S\backslash NP)\backslash(S\backslash NP)$) as an argument. Second, applying z to the type-lifted VP closes off the dependency between the two argument slots that will be saturated by the subject and the adjunct. Last, the resulting VP merges with the bindee-containing adjunct and the subject binder. (See Jacobson 1999 for a related discussion tackling unbound readings.) Despite the distinct analytical moves, the analyses in (16b) and in (i) 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 analysis in (i) has limitations; it doesn't extend to cases where the binder is inside the VP, such as in *We will sell no wine$_i$ before its$_i$ time*. For my own explanation, see section 3.3.

(17) [His$_i$ mom] loves him$_i$.

a. # [[His mom] z(loves) him] (Meaning: 'His mom loves his mom.' $\times$)

b. [i (His mom loves him) ] (Meaning: 'His mom loves him.' $\surd$)

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

(18)

$$
\dfrac{
\begin{array}{c} \textit{His mom} \\ \hdashline \mathrm{NP}^{\mathrm{NP}} \quad \lambda x_e.mom(x) \end{array}
\quad
\begin{array}{c} \textit{loves him} \\ \hdashline (\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}} \quad \lambda y_e \lambda x_e.lv(x,y) \\ \hdashline (\mathrm{S}^{\mathrm{NP}}\backslash\mathrm{NP}^{\mathrm{NP}})^{\mathrm{NP}} \quad \lambda y_e \lambda f_{ee} \lambda x_e.lv(f(x),y) \end{array}\ \text{g-sup}_1
}{
\begin{array}{c} (\mathrm{S}^{\mathrm{NP}})^{\mathrm{NP}} \quad \lambda y_e \lambda x_e.lv(mom(x),y) \\ \hdashline \mathrm{S}^{\mathrm{NP}} \quad \lambda x_e.lv(mom(x),x) \end{array}\ \mathrm{i}
}\ {<}\mathrm{FA}_1
$$

## 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 $\lambda i$ [ $t_i$ loves his$_i$ mother ]]

This section will propose a variable-free approach to quantificational binding that takes the insight in (20) but does not make use of quantifier raising or indexation.[8] Briefly, by applying the i-rule, my 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, semantic composition takes place at LF, an abstract level of representations in which the linear order of strings can be different from that in the surface 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. For this reason, my analysis of scoping doesn't resort to quantifier raising.

### 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 $\left[\frac{\text{TOP}}{\text{BOTTOM}}\right]$. For example, the category and meaning of *everyone* are as follows ($\mathit{eo}$ abbreviates $\lambda P_{et}.\mathit{eo}(\lambda x_e.P(x))$):

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

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 $\Pi$) 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 quantificational binder 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'.

To tackle composition with two-dimensional expressions, I define binary FA rules as in (22). (22a) says: when a two-dimensional expression $\alpha$ composes with a one-dimensional expression by FA, it is the bottom component of $\alpha$ that participates in the reduction by FA. (22b) tackle cases with multiple quantifiers: if both of the expressions that participate in FA are two-dimensional, the one that serves as the function takes narrow scope. Other binary rules are defined analogously.

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

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

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

Inward FA applies to a single two-dimensional expression. As schematized in (23a), when the bottom component of a two-dimensional expression can serve as the argument of the top, apply the top to the bottom and reduce this two-dimensional

9 To be exact, Jacobson's framework doesn't assume empty categories like traces in grammar. 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.

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)

$$\text{a. } \left[\frac{\mathrm{A}\,[\,\mathrm{B}}{\mathrm{B}}\right] \xrightarrow[\vee\mathrm{FA}]{} \mathrm{A} \qquad \text{b. } \left[\frac{\mathrm{C}\,[\,\mathrm{A}}{\left[\frac{\mathrm{A}\,[\,\mathrm{B}}{\mathrm{B}}\right]}\right] \xrightarrow[\vee\mathrm{FA}]{} \left[\frac{\mathrm{C}\,[\,\mathrm{A}}{\left(\left[\frac{\mathrm{A}\,[\,\mathrm{B}}{\mathrm{B}}\right]\right)_{\vee\mathrm{FA}}}\right] = \left[\frac{\mathrm{C}\,[\,\mathrm{A}}{\mathrm{A}}\right] \xrightarrow[\vee\mathrm{FA}]{} \mathrm{C}$$

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, we have:

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

### 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 binder role of the pronominal possessor in (25a) (of category $\mathrm{NP}^{\mathrm{NP}}$) is now taken by the ‘trace’-like component of the quantificational possessor (of category $\mathrm{NP}/\mathrm{NP}$).

(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 $(\mathrm{S}/\mathrm{NP})^{\mathrm{NP}}$-category denoting a two-place predicate. In this output, the sup-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 to the resulting two-dimensional sentence yields a dependency between these two NPs and shifts the bottom component of the sentence into an $\mathrm{S}/\mathrm{NP}$ denoting a one-place predicate. This $\mathrm{S}/\mathrm{NP}$ may serve as an argument of the quantificational part at the top. (Recall that $\Pi$ abbreviates $\mathrm{S}[(\mathrm{S}/\mathrm{NP})$.) 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$_i$.

$$\underbrace{\left[\frac{\Pi}{T}\right]\left[\frac{\mathit{eb}}{\lambda x_e.x}\right]}_{\textit{every boy}} \quad \underbrace{\mathrm{NP}\backslash\mathrm{NP} \quad \lambda x_e.\mathit{mom}(x)}_{\textit{'s mom}} \xrightarrow{\text{g-sl}} T\backslash T \quad \lambda f_{ee}\lambda x_e.\mathit{mom}(f(x))$$

$$\xrightarrow{<\text{FA}} \left[\frac{\Pi}{T}\right]\left[\frac{\mathit{eb}}{\lambda x_e.\mathit{mom}(x)}\right]$$

$$\underbrace{(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}} \quad \lambda y_e\lambda x_e.\mathit{lv}(x,y)}_{\textit{loves him}} \xrightarrow{\text{g-sl}_1} ((\mathrm{S}/\mathrm{NP})\backslash T)^{\mathrm{NP}} \quad \lambda y_e\lambda g_{ee}\lambda x_e.\mathit{lv}(g(x),y)$$

$$\xrightarrow{<\text{FA}_1} \left[\frac{\Pi}{(\mathrm{S}/\mathrm{NP})^{\mathrm{NP}}}\right]\left[\frac{\mathit{eb}}{\lambda y_e\lambda x_e.\mathit{lv}(\mathit{mom}(x),y)}\right]$$

$$\xrightarrow{\text{i}} \left[\frac{\Pi}{\mathrm{S}/\mathrm{NP}}\right]\left[\frac{\mathit{eb}}{\lambda x_e.\mathit{lv}(\mathit{mom}(x),x)}\right]$$

$$\xrightarrow{\vee\text{FA}} \mathrm{S} \quad \mathit{eb}(\lambda x_e.\mathit{lv}(\mathit{mom}(x),x))$$

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

(27) **Lift** (a la Montague lift)
For any expression $\alpha$ s.t. $\textsc{cat}(\alpha) = \mathrm{NP}$, we have:
$\textsc{cat}(\textsc{lift}(\alpha)) = \left[\frac{\mathrm{S}[(\mathrm{S}/\mathrm{NP})}{\mathrm{NP}/\mathrm{NP}}\right]$ (abbr.: $\left[\frac{\Pi}{T}\right]$), and $[\![\textsc{lift}(\alpha)]\!] = \left[\frac{\lambda P_{et}.P([\![\alpha]\!])}{\lambda x_e.x}\right]$.

### 3.3 More cases of binding into adjuncts

The analysis of binding into adjuncts given in (16b) only works in cases where the binder serves as the subject and the bindee-containing adjunct serves as a VP-modifier. It doesn't apply to the cases below: 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 account allows us to derive dependencies in the above 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 sentence.

For example, in (28a), applying g-sup to each of 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 $(\mathrm{S}/\mathrm{NP})^{\mathrm{NP}}$-category at the bottom. Further, applying i to this sentence closes off the dependency between *its* and the 'trace'-part of *no wine*, yielding a binding-into-adjunct effect.

### 3.4 Scope ambiguity

In GB-style compositional semantics, scope ambiguity of quantifiers is derived by quantifier raising. In the enriched variable-free system proposed in this paper, as seen in the definition of FA in (22b), the scopal relation between two quantifiers is determined by how they participate 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 illustrated in Figure 1, composing the subject *everyone* with the VP *loves someone* by backward FA derives the surface scope reading, where $\mathit{eo}$ scopes above $\mathit{so}$ in (a). The resulting stack is reduced into a simple S by applying inward FA twice.

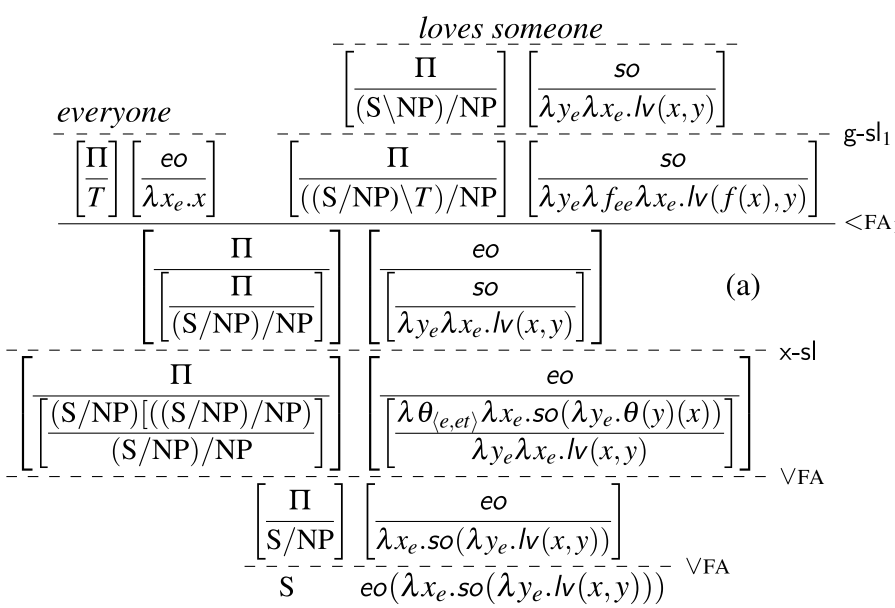


**Figure 1** Surface scope reading of *Everyone loves someone*

One caveat here is that inward FA doesn't apply to (a) in Figure 1: 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 slash-NP abstraction (corresponding to $\lambda x$) stemming from the 'trace'-part of *everyone*. I resolve this type-mismatch by the following x-rules, which apply to particular two-dimensional expressions (here the subscript $c$ in $\mathbf{x}_c$ refers to the category $C$):

(30) **The x-rules**

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

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

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

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

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

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

In Figure 1, applying x-sl to (a) converts the quantificational part of *someone* into a complex category that selects an (S/NP)/NP argument (corresponding to $\lambda\theta$). It also transmits the information about the 'trace' of *everyone* from the bottom component to the existential quantification (viz., passing up the abstraction of $x$ over *so*), which further forms the bottom component of the universal quantification. After the application of x-sl, inward FA can proceed as normal.

### 3.4.2 Inverse scoping

As mentioned in (29), in "Q1 V Q2", an inverse scope reading arises if it is the entire Q1 (as opposed to the 'trace'-part of Q1 only) that participates in 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 slot into one that needs to be filled by a two-dimensional quantifier, defined as in (31).

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

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

$\textsc{cat}(\text{ALIFT}(\alpha)) = \begin{bmatrix}\Pi \\ \hline \text{X/NP}\end{bmatrix} | \begin{bmatrix}\Pi \\ \hline T\end{bmatrix}$ and $[\![\text{ALIFT}(\alpha)]\!] = \lambda[\frac{\pi}{f_{ee}}].\begin{bmatrix}\pi \\ \hline \lambda x_e.[\![\alpha]\!](f(x))\end{bmatrix}$

Details of composition are given in Figure 2. First, by applying $\text{ALift}_1$ to the VP, the subject argument slot which formerly called for a $T$ now wants to be filled by a two-dimensional quantifier. Second, 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 type-shifting and composition rules, except the last inward FA, carry the subscript 1, since they need to skip the extra slash-NP abstraction (corresponding to $\lambda y$) stemming from the 'trace'-part of the quantificational object *someone*.

*loves someone*

$$\left[\frac{\Pi}{(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}}\right]\left[\frac{\mathit{so}}{\lambda y_e\lambda x_e.\mathit{lv}(x,y)}\right]$$

$$\left[\frac{\Pi}{((\mathrm{S}/\mathrm{NP})\backslash T)/\mathrm{NP}}\right]\left[\frac{\mathit{so}}{\lambda y_e\lambda f_{ee}\lambda x_e.\mathit{lv}(f(x),y)}\right]\quad \text{g-sl}_1$$

*everyone*

$$\left[\frac{\Pi}{T}\right]\left[\frac{\mathit{eo}}{\lambda x_e.x}\right]\qquad \left[\frac{\Pi}{\left(\left[\frac{\Pi}{\mathrm{S}/\mathrm{NP}}\right]\backslash\left[\frac{\Pi}{T}\right]\right)/\mathrm{NP}}\right]\left[\frac{\mathit{so}}{\lambda y_e\lambda[\frac{\pi}{f_{ee}}].\left[\frac{\pi}{\lambda x_e.\mathit{lv}(f(x),y)}\right]}\right]\quad \mathrm{A}\textsc{lift}_1$$

$$\left[\frac{\Pi}{\left[\frac{\Pi}{\mathrm{S}/\mathrm{NP}}\right]/\mathrm{NP}}\right]\left[\frac{\mathit{so}}{\lambda y_e.\left[\frac{\mathit{eo}}{\lambda x_e.\mathit{lv}(x,y)}\right]}\right]\quad \text{(a)}\quad <\mathrm{FA}_1$$

$$\left[\frac{\Pi}{\mathrm{S}/\mathrm{NP}}\right]\left[\frac{\mathit{so}}{\lambda y_e.\mathit{eo}(\lambda x_e.\mathit{lv}(x,y))}\right]\quad \vee\mathrm{FA}_1$$

$$\mathrm{S}\qquad \mathit{so}(\lambda y_e.\mathit{eo}(\lambda x_e.\mathit{lv}(x,y)))\quad \vee\mathrm{FA}$$

**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 city* take wide scope, and (iii) non-c-commanding binding: to allow *every city* to bind the pronoun *it* from a non-c-commanding position. Discussion in previous sections has explained (ii) and (iii), resorting to ALIFT and i, respectively. This subsection will add one more ingredient to derive restrictor modification.

(32) [Someone from every city$_i$] [despises it$_i$].

Given the assumption that it is the ‘trace’-part of *someone*, which is a $T$-category denoting an identity function, that participates in composition, I define a restrictor modification rule as in (33). This rule converts a predicative modifier into a $T\backslash T$-category, which functions as a restrictor of $T$-categories. (Here $f' \leq f$ means that every pair in function $f'$ is a pair in function $f$.)

(33) **The RES-rule** (to derive restrictor modification)
For any $P$ s.t. $\textsc{cat}(P) = \mathrm{S}\backslash\mathrm{NP}$, we have: $\textsc{cat}(\textsc{Res}(P)) = T\backslash T$, and

$[\![\textsc{Res}(P)]\!] = \lambda f_{ee}.\textsc{Max}_{f' \leq f}[\forall x' \in \mathrm{Dom}(f')[[\![P]\!](f'(x'))]]$
(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 a member of $[\![P]\!]$.)

The complex subject is now composed as in Figure 3. Applying the restriction rule to the PP *from every city* converts it 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): Composing the complex quantificational subject

*from every city*

$$\left[\frac{\Pi}{(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}}\right]\left[\frac{\mathsf{ec}}{\lambda y_e \lambda x_e.\mathit{fr}(x,y)}\right]$$

$$\xrightarrow{\textsc{Res}_1}\left[\frac{\Pi}{(T\backslash T)/\mathrm{NP}}\right]\left[\frac{\mathsf{ec}}{\lambda y_e \lambda f_{ee}.\textsc{Max}_{f'\leq f}[\forall x' \in \mathrm{Dom}(f')[\mathit{fr}(f'(x'),y)]]}\right]$$

$$\xrightarrow{\textsc{ALift}_1}\left[\frac{\Pi}{\left(\left[\frac{\Pi}{T}\right]\backslash\left[\frac{\Pi}{T}\right]\right)/\mathrm{NP}}\right]\left[\frac{\mathsf{ec}}{\lambda y_e \lambda\left[\frac{\pi}{f}\right].\left[\frac{\pi}{\textsc{Max}_{f'\leq f}[\forall x' \in \mathrm{Dom}(f')[\mathit{fr}(f'(x'),y)]]}\right]}\right]$$

*someone*

$$\left[\frac{\Pi}{T}\right]\left[\frac{\mathsf{so}}{\lambda x_e.x}\right]$$

$$\xrightarrow{<\mathrm{FA}_1}\left[\frac{\Pi}{\left[\frac{\Pi}{T}\right]/\mathrm{NP}}\right]\left[\frac{\mathsf{ec}}{\lambda y_e.\left[\frac{\mathsf{so}}{\textsc{Max}_{f'\leq(\lambda x_e.x)}[\forall x' \in \mathrm{Dom}(f')[\mathit{fr}(f'(x'),y)]]}\right]}\right]$$

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

---

In the composition output, the $T$-category part at the 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 $\lambda x_e : \mathit{fr}(x,y).x$, 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 $\left[\frac{\Pi}{T}\right]/\mathrm{NP}$. In particular, the first $\mathsf{g\text{-}sl}_1$ and $\textsc{ALift}_1$ are applied to allow the VP to select a two-dimensional quantificational argument, and the second $\mathsf{g\text{-}sl}_1$ is applied to get ready to transmit 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 $\mathsf{i}$-rule yields a dependency between the 'trace' of *every city* and the object pronoun *it* (corresponding to $\lambda y$ and $\lambda z$, respectively). Last, applying inward FA twice reduces the stack into a simple $\mathsf{S}$ expressing inverse linking.

Step (B): Type-shifting the VP

$$
\begin{array}{ll}
\multicolumn{2}{c}{\textit{despises it}} \\
\hdashline
(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}} & \lambda z_e \lambda x_e.\mathit{desp}(x,z) \\
\hdashline
((\mathrm{S}/\mathrm{NP})\backslash T)^{\mathrm{NP}} & \lambda z_e \lambda f_{ee} \lambda x_e.\mathit{desp}(f(x),z) \qquad \text{g-sl}_1 \\
\hdashline
\left(\left[\dfrac{\Pi}{\mathrm{S}/\mathrm{NP}}\right]\backslash\left[\dfrac{\Pi}{T}\right]\right)^{\mathrm{NP}} & \lambda z_e \lambda [\tfrac{\pi}{f}].\left[\dfrac{\pi}{\lambda x_e.\mathit{desp}(f(x),z)}\right] \qquad \text{ALIFT}_1 \\
\hdashline
\left(\left(\left[\dfrac{\Pi}{\mathrm{S}/\mathrm{NP}}\right]/\mathrm{NP}\right)\backslash\left(\left[\dfrac{\Pi}{T}\right]/\mathrm{NP}\right)\right)^{\mathrm{NP}} & \lambda z_e \lambda \rho_{\langle e,[\frac{\langle et,t\rangle}{ee}]\rangle} \lambda y_e.\left[\lambda [\tfrac{\pi}{f}].\left[\dfrac{\pi}{\lambda x_e.\mathit{desp}(f(x),z)}\right]\right](\rho(y)) \qquad \text{g-sl}_1
\end{array}
$$

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

$$
\begin{array}{ll}
\text{(A) \textit{someone from every city}} & \text{(B) \textit{despises it}} \\
\hdashline
\left[\dfrac{\Pi}{\left(\left[\dfrac{\Pi}{\mathrm{S}/\mathrm{NP}}\right]/\mathrm{NP}\right)^{\mathrm{NP}}}\right] & \left[\dfrac{\mathit{ec}}{\lambda z \lambda y.\left[\dfrac{\mathit{so}}{\lambda x.\mathit{desp}((\mathrm{MAX}_{f'\leq(\lambda x_e.x)}[\forall x'\in \mathrm{Dom}(f')[\mathit{fr}(f'(x'),y)]])(x),z)}\right]}\right] \qquad <\mathrm{FA}_1 \\
\hdashline
\left[\dfrac{\Pi}{\left(\left[\dfrac{\Pi}{\mathrm{S}/\mathrm{NP}}\right]/\mathrm{NP}\right)}\right] & \left[\dfrac{\mathit{ec}}{\lambda y.\left[\dfrac{\mathit{so}}{\lambda x.\mathit{desp}((\mathrm{MAX}_{f'\leq(\lambda x_e.x)}[\forall x'\in \mathrm{Dom}(f')[\mathit{fr}(f'(x'),y)]])(x),y)}\right]}\right] \qquad \text{i} \\
\hdashline
\left[\dfrac{\Pi}{\mathrm{S}/\mathrm{NP}}\right] & \left[\dfrac{\mathit{ec}}{\lambda y.\mathit{so}(\lambda x.\mathit{desp}((\mathrm{MAX}_{f'\leq(\lambda x_e.x)}[\forall x'\in \mathrm{Dom}(f')[\mathit{fr}(f'(x'),y)]])(x),y))}\right] \qquad \vee\mathrm{FA}_1 \\
\hdashline
\left[\dfrac{\Pi}{\mathrm{S}/\mathrm{NP}}\right] & \left[\dfrac{\mathit{ec}}{\lambda y.\mathit{so}(\lambda x.\mathit{fr}(x,y)\wedge \mathit{desp}(x,y))}\right] \qquad \text{equivalent} \\
\hdashline
\mathrm{S} & \mathit{ec}(\lambda y.\mathit{so}(\lambda x.\mathit{fr}(x,y)\wedge \mathit{desp}(x,y))) \qquad \vee\mathrm{FA}
\end{array}
$$

**Figure 4** Composition of *Someone from every city*$_i$ *despises it*$_i$ (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 '$(\mathrm{X}\ddagger\mathrm{NP})^{\mathrm{NP}}$'. According to this constraint, the sup-NP abstraction provided by a bindable pronoun has to be reduced first, before any other NP abstraction saved for the binder of this pronoun is reduced.

According to this constraint, the i-rule can be applied to $(S\backslash NP)^{NP}$, $(S/NP)^{NP}$, 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 the VP nor to the full sentence. As seen below, both strings have an $(S{\ddagger}NP)/NP$-category at the bottom. The composition in (36) yields an $S^{NP}$ that expresses an unbound reading.

(36)

*invited everyone*

$$\left[\frac{\Pi}{(S\backslash NP)/NP}\right] \quad \text{(can't be i-shifted)}$$

$$\text{- - - - - - - - - - - - - - - - - - - - g-sup}_1$$

*His mom* $\quad$ $NP^{NP}$ $\quad$ $\left[\frac{\Pi}{(S^{NP}\backslash NP^{NP})/NP}\right]$

$$\text{—————————————— <FA}_1$$

$$\left[\frac{S[(S/NP)}{S^{NP}/NP}\right] \quad \text{(can't be i-shifted)}$$

$$\text{- - - - - - - - - - - - - - - - - - - - x-sup}$$

$$\left[\frac{S^{NP}[(S^{NP}/NP)}{S^{NP}/NP}\right]$$

$$\text{- - - - - - - - - - - - - - - - - - - - } \vee\text{FA}$$

$$S^{NP} \quad \lambda x.eo(\lambda y.ivt(mom(x),y))$$

## 5 Paycheck pronouns

In the following sentences, the pronouns in bold are called 'paycheck pronouns'. In (37a), the pronoun *her* refers to Billy's mom. Descriptively, this reading 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 $\underline{\text{his}_i \text{ mom}}$, but Billy$_j$ hates **her**$_{=\text{his}_j\text{ mom}}$.
(Intended: 'Andy loves his mom, but Billy hates his mom.')

b. The woman [who$_i$ deposited $\underline{\text{her}_i \text{ paycheck}}$ in the bank] was wiser than the woman [who$_j$ deposited **it**$_{=\text{her}_j\text{ paycheck}}$ 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 $[\![\textit{his mom}]\!]$ $(= \lambda x_e.\mathsf{mom}(x))$. 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 co-referential relation between *his mom* and the paycheck pronoun *her* isn't really a binding relation; 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 cross-sentential dependency between *his mom* and *her*. By lifting $\text{NP}^{\text{NP}}$-categories into two-dimensional quantifiers, the analysis made for non-c-commanding 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 $\alpha$ s.t. X is NP or a pronominal category, we have:
$\textsc{cat}(\textsc{lift}(\alpha)) = \left[\dfrac{\text{S}[(\text{S}/\text{X})}{\text{X}/\text{X}}\right]$, and $[\![\textsc{lift}(\alpha)]\!] = \left[\dfrac{\lambda P_{\langle x,t\rangle}.P([\![\alpha]\!])}{\lambda X_x.X}\right]$

(39) **Pronominal categories** (after Charlow To appear)
$\text{NP}^{\text{NP}}$ is a pronominal category; if $\text{Y}^{\text{X}}$ 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 this pronoun 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)
$$\dfrac{\dfrac{\dfrac{\dfrac{\text{[Andy i [loves } \textsc{lift}\text{(his mom)]]}}{\left[\dfrac{\text{S}[(\text{S}/\text{NP}^{\text{NP}})}{\text{S}/\text{NP}^{\text{NP}}}\right] \left[\dfrac{\lambda P_{\langle ee,t\rangle}.P(\lambda x_e.\mathsf{mom}(x))}{\lambda f_{ee}.\mathsf{lv}(a,f(a))}\right]} \quad \dfrac{\text{[Billy i [hates g-sup(}\textbf{her}\text{)]]}}{\text{S}^{(\text{NP}^{\text{NP}})} \quad \lambda g_{ee}.\mathsf{hate}(b,g(b))}}{\left[\dfrac{\text{S}[(\text{S}/\text{NP}^{\text{NP}})}{(\text{S}/\text{NP}^{\text{NP}})^{(\text{NP}^{\text{NP}})}}\right] \left[\dfrac{\lambda P_{\langle ee,t\rangle}.P(\lambda x_e.\mathsf{mom}(x))}{\lambda g_{ee}\lambda f_{ee}.\mathsf{lv}(a,f(a)) \wedge \mathsf{hate}(b,g(b))}\right]}\text{g, FA}}{\left[\dfrac{\text{S}[(\text{S}/\text{NP}^{\text{NP}})}{\text{S}/\text{NP}^{\text{NP}}}\right] \left[\dfrac{\lambda P_{\langle ee,t\rangle}.P(\lambda x_e.\mathsf{mom}(x))}{\lambda f_{ee}.\mathsf{lv}(a,f(a)) \wedge \mathsf{hate}(b,f(b))}\right]}\text{i}}{\text{S} \quad \mathsf{lv}(a,\mathsf{mom}(a)) \wedge \mathsf{hate}(b,\mathsf{mom}(b))}\vee\text{FA}$$

## 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 $\alpha$ s.t. $\textsc{cat}(\alpha) = (A|B)|\mathrm{NP}$, we have:
$\textsc{cat}(\mathsf{s}(\alpha)) = (A|B^{\mathrm{NP}})|\mathrm{NP}$, and $[\![\mathsf{s}(\alpha)]\!] = \lambda x_e \lambda f_{eb}.[\![\alpha]\!](x)(f(x))$

(42) Every boy$_i$'s mom loves him$_i$. ($G$ abbreviates $\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})$)

$$\frac{\textit{'s mom}}{\mathrm{NP}\backslash\mathrm{NP} \quad \lambda x.\mathit{mom}(x)}$$

$$\frac{\mathrm{NP}\backslash\mathrm{NP} \quad \lambda x.\mathit{mom}(x)}{(\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP}))\backslash\mathrm{NP} \quad \lambda x\lambda\pi.\pi(\lambda P.P(\mathit{mom}(x)))}\ \textsc{vr}$$

$$\frac{(\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP}))\backslash\mathrm{NP} \quad \lambda x\lambda\pi.\pi(\lambda P.P(\mathit{mom}(x)))}{(\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}})\backslash\mathrm{NP} \quad \ldots}\ \mathsf{s}$$

$$\frac{(\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}})\backslash\mathrm{NP} \quad \ldots}{(\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}})\backslash G \quad \ldots}\ \textsc{ALift}$$

$$\frac{\dfrac{\textit{every boy}}{G \quad \mathit{eb}} \qquad (\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}})\backslash G \quad \ldots}{\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}} \quad \lambda R.\mathit{eb}(\lambda x.R(x)(\mathit{mom}(x)))}\ <\textsc{fa}$$

$$\frac{\mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}} \quad \lambda R.\mathit{eb}(\lambda x.R(x)(\mathit{mom}(x))) \qquad \dfrac{\textit{loves him}}{(\mathrm{S}\backslash\mathrm{NP})^{\mathrm{NP}} \quad \lambda y\lambda x.\mathit{lv}(x,y)}}{\mathrm{S} \quad \mathit{eb}(\lambda x.\mathit{lv}(\mathit{mom}(x),x))}\ <\textsc{fa}$$

The s-rule, however, is abandoned in Jacobson 1999 since it allows for crossover. To avoid generating crossover constructions, Barker further builds a separate value assigning and transmitting 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 already exists 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'-like part. This two-dimensional treatment is particularly useful for tackling scoping and 'non-c-commanding' binding. In future research, I would like to see how the enriched system compares to 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