Syntacticians of Old developed a simple, but largely effective, model of constituency. The point of that model is to map strings of words, or maybe morphemes, into larger units. For this class, we can think of those units as semantic objects, though it will be the syntax that individuates them. The units are called phrases.

1. **Goal:** Define $\mathcal{P}$.

$$ w_1, w_2, w_3, \ldots, w_n \leftrightarrow \mathcal{P} \rightarrow \{ \{ w_1, \{ w_j, w_k \} \}, \{ w_k, w_m \}, \ldots \} $$

Phrases (indicated with “{””) are those sets of words that have compositional denotations, and behave as units for the syntax and prosody. Like words, phrases can themselves be parts of phrases. Two ingredients in $\mathcal{P}$ are the denotations of the words and phrases and the morphosyntactic category of the words and phrases. For this class, I’ll assume that words and morphemes come into the syntactic system with their morphosyntactic category specified. A popular way of thinking of the part of $\mathcal{P}$ that is responsible for identifying the phrases is Chomsky’s (1995) **merge**.

2. $$ \text{merge}(\alpha, \beta) = \{ y, \alpha, \beta \} $$

The set formed by merge will have a category status which $y$ is meant to represent. We should understand $\alpha$ and $\beta$ as variables that range over (possibly trivial) strings of words, morphemes, and the output of merge. I will represent the output of merge with the somewhat misleading tree notation:

3. $\alpha \overset{y}{\rightarrow} \beta$

(3) is misleading because it graphically puts $\alpha$ and $\beta$ in a linear order. That ordering is not part of merge (which is why the set notation is used in (2)). Nonetheless, when the going gets tough, we’re going to appreciate trees. Despite the typography, therefore, be mindful that (3) corresponds to an unordered set of $\alpha$ and $\beta$. (2) becomes (4).

4. $$ \text{merge}(\alpha, \beta) = y $$

Defined this way, merge requires every phrase to have exactly two elements. That is controversial. We won’t ever find a need to increase the number of elements a phrase can contain. But we will find a need to decrease it. So, somewhat unorthodoxically, I will also allow (5).

5. $$ \text{merge}(\alpha) = y $$

Let us understand that when merge takes one argument, that argument must be a word and its output must be a phrase.

A working hypothesis is that $y$ is always the same morphosyntactic category as one of its elements. There are regularities that seem to govern which element determines that category but I will treat this as an independent, phrase specific, rule. These are the components that determine which sets of words can be formed into phrases. I’ll call this the “phrase part” of $\mathcal{P}$.

6. $$ \text{Phrase Part} $$

$$ \text{merge}(\alpha, (\beta)) = y $$

subject to the following conditions:

a. $\alpha$ and $\beta$ may be words, morphemes or phrases.

b. If $\beta$ is absent, $\alpha$ must be a word and $y$ a phrase.

c. $y$ is the category of $\alpha$ or $\beta$.

d. $\alpha$ and $\beta$ must compose semantically.

e. $\alpha$ and $\beta$ must be certain categories.

What I mean to point to with (6e) are things like the fact that a word of determiner type in English can only merge with something of nominal type. I will call a “derivation” a serial application of merge to a set of words.
An example of a derivation is:

\[ \text{(7)} \]

a. \[ \begin{array}{c}
\text{D}^0 & \text{T}^0 & \text{V}^0 & \text{D}^0 & \text{N}^0 & \text{P}^0 & \text{D}^0 & \text{N}^0 \\
\end{array} \]

\[ \begin{array}{c}
\text{she} & \text{will} & \text{meet} & \text{the} & \text{boy} & \text{in} & \text{the} & \text{store} \\
\end{array} \]

b. \[ \begin{array}{c}
\text{DP} & \text{T}^0 & \text{V}^0 & \text{D}^0 & \text{N}^0 & \text{P}^0 & \text{D}^0 & \text{N}^0 \\
\end{array} \]

\[ \begin{array}{c}
\text{D}^0 & \text{will} & \text{meet} & \text{the} & \text{boy} & \text{in} & \text{the} & \text{store} \\
\end{array} \]

c. \[ \begin{array}{c}
\text{DP} & \text{T}^0 & \text{V}^0 & \text{D}^0 & \text{NP} & \text{P}^0 & \text{D}^0 & \text{N}^0 \\
\end{array} \]

\[ \begin{array}{c}
\text{D}^0 & \text{will} & \text{meet} & \text{the} & \text{N}^0 & \text{in} & \text{the} & \text{store} \\
\end{array} \]

d. \[ \begin{array}{c}
\text{DP} & \text{T}^0 & \text{V}^0 & \text{D}^0 & \text{NP} & \text{P}^0 & \text{D}^0 & \text{NP} \\
\end{array} \]

\[ \begin{array}{c}
\text{D}^0 & \text{will} & \text{meet} & \text{the} & \text{N}^0 & \text{in} & \text{the} & \text{N}^0 \\
\end{array} \]

e. \[ \begin{array}{c}
\text{DP} & \text{T}^0 & \text{V}^0 & \text{D}^0 & \text{NP} & \text{P}^0 & \text{DP} \\
\end{array} \]

\[ \begin{array}{c}
\text{D}^0 & \text{will} & \text{meet} & \text{the} & \text{N}^0 & \text{in} & \text{D}^0 & \text{NP} \\
\end{array} \]

\[ \begin{array}{c}
\text{she} & \text{boy} & \text{store} \\
\end{array} \]

f. \[ \begin{array}{c}
\text{DP} & \text{T}^0 & \text{V}^0 & \text{D}^0 & \text{NP} & \text{PP} \\
\end{array} \]

\[ \begin{array}{c}
\text{D}^0 & \text{will} & \text{meet} & \text{the} & \text{N}^0 & \text{in} & \text{D}^0 & \text{NP} \\
\end{array} \]

\[ \begin{array}{c}
\text{she} & \text{boy} & \text{store} \\
\end{array} \]

g. \[ \begin{array}{c}
\text{DP} & \text{T}^0 & \text{V}^0 & \text{D}^0 & \text{NP} \\
\end{array} \]

\[ \begin{array}{c}
\text{D}^0 & \text{will} & \text{meet} & \text{the} & \text{NP} & \text{PP} \\
\end{array} \]

\[ \begin{array}{c}
\text{she} & \text{boy} & \text{store} \\
\end{array} \]

h. \[ \begin{array}{c}
\text{DP} & \text{T}^0 & \text{V}^0 & \text{DP} & \text{the} & \text{NP} & \text{PP} \\
\end{array} \]

\[ \begin{array}{c}
\text{D}^0 & \text{will} & \text{meet} & \text{D}^0 & \text{NP} & \text{PP} \\
\end{array} \]

\[ \begin{array}{c}
\text{she} & \text{boy} & \text{store} \\
\end{array} \]

i. \[ \begin{array}{c}
\text{DP} & \text{T}^0 & \text{VP} & \text{the} & \text{NP} & \text{PP} \\
\end{array} \]

\[ \begin{array}{c}
\text{D}^0 & \text{will} & \text{meet} & \text{D}^0 & \text{NP} & \text{PP} \\
\end{array} \]

\[ \begin{array}{c}
\text{she} & \text{meet} & \text{D}^0 & \text{NP} & \text{PP} \\
\end{array} \]

\[ \begin{array}{c}
\text{the} & \text{boy} & \text{store} \\
\end{array} \]
Or, the derivation can go differently at step (g).

\[
\text{j.'} \quad \text{DP} \quad \text{TP} \\
\quad \text{D}^0 \quad \text{T}^0 \quad \text{VP} \\
\quad \text{she} \quad \text{will} \quad \text{meet} \quad \text{D}^0 \quad \text{NP} \\
\quad \text{the} \quad \text{NP} \quad \text{PP} \\
\text{h.'} \quad \text{DP} \quad \text{T}^0 \quad \text{VP} \quad \text{PP} \\
\quad \text{D}^0 \quad \text{will} \quad \text{meet} \quad \text{D}^0 \quad \text{NP} \\
\quad \text{the} \quad \text{N}^0 \quad \text{in} \quad \text{D}^0 \quad \text{NP} \\
\text{i.'} \quad \text{DP} \quad \text{T}^0 \quad \text{VP} \\
\quad \text{D}^0 \quad \text{will} \quad \text{meet} \quad \text{D}^0 \quad \text{NP} \\
\quad \text{the} \quad \text{N}^0 \quad \text{in} \quad \text{D}^0 \quad \text{NP} \\
\text{j.'} \quad \text{DP} \quad \text{TP} \\
\quad \text{D}^0 \quad \text{T}^0 \quad \text{VP} \\
\quad \text{she} \quad \text{will} \quad \text{meet} \quad \text{D}^0 \quad \text{NP} \\
\quad \text{the} \quad \text{N}^0 \quad \text{in} \quad \text{D}^0 \quad \text{NP} \\
\quad \text{boy} \quad \text{store} \]
Conventions:

a. Words will be represented with $X^0$, phrases with XP, and non-word morphemes with X.

b. When one of the arguments of `merge` is a word, the output is a phrase.

c. Each step in this series I’ll call a Stage. (7a), the initial Stage, is also called a Numeration.

It is standard to require `merge` to bring everything in the Numeration into one phrase, so let’s add that.

a. `merge` applies serially until its output is a single phrase.

I think there are interesting examples which suggest that (6) shouldn’t be enforced, but for this class we’ll adopt this constraint as well. We can now define a sentence’s derivation:

(9) $\mathcal{D}(S_i), S_i$ a Numeration, is the series $(S_i, S_{i+1}, S_{i+2}, \ldots, S_m)$, where:

a. each $S_{k+1}$ is derived from $S_{k-1}$ by one application of `merge`, and

b. $S_m$ is a single phrase (i.e., a singleton set).

We should think of Stages as sets, and as each step in the derivation as reducing the size of that set by combining its elements.

There is a variant of (6) which characterizes the arguments of `merge` differently, and places a constraint on what a derivation can be. This definition changes what the arguments of `merge` can be so that they must be the elements of the Stage that the application of `merge` acts on. Let’s add that condition to `merge`. (I’ve smoothed out the constraints on the relationship between words and phrases in (10) as well.)

(10) $\text{merge}(\alpha, (\beta)) = \gamma$, subject to the following conditions:

a. $\alpha$ and $\beta$ are elements of a Stage.

b. If one of the arguments of `merge` is a word, $\gamma$ must be a phrase.

c. The sole argument of `merge` cannot be a phrase.

d. $\gamma$ is the category of $\alpha$ or $\beta$.

e. $\alpha$ and $\beta$ must semantically compose.

f. $\alpha$ and $\beta$ must be certain categories.

This prevents derivations in which `merge` applies to something that is inside another phrase. It corresponds to something Chomsky called the “Extension Condition,” and it puts a useful cap on the kinds of phrases that can be created.

This captures how the phrasal side of $\mathcal{P}$ is calculated. What we need now is something for the string part. There is a language component side to this, and an apparently universal side. The standard models right now are heavily influenced by Kayne (1994). One decision in Kayne is to model strings in terms of a set of ordered pairs. These ordered pairs are generated by a procedure that interprets the output of $\mathcal{D}$. This puts a direction on $\mathcal{P}$—it causes the phrasal part to feed the string part. I’ll call the procedure that maps the output of $\mathcal{D}$ onto a string a “linearization,” and represent it with $\mathcal{L}$.

(11) $\mathcal{L}(P) = \{ S; S = \{a<b: a \text{ and } b \text{ are terminals dominated by } P \} \}$

NB: “$a<b$” means “$a$ precedes $b$”

$\mathcal{L}(P)$ is the set, each element of which is a set of ordered pairs $a<b$, where $a$ and $b$ are terminals in $P$. This set, then, is all of the possible ways of ordering the terminals in $P$. For example:
(12) $L(\quad \mathrm{VP} \quad ) =$
\[
\begin{array}{c}
V^0 \\
| \\
meet \\
| \\
the \\
| \\
boy \\
\end{array}
\begin{array}{c}
D^0 \\
| \\
NP \\
| \\
\end{array}
\begin{array}{c}
\uparrow \\
DP \\
\uparrow \\
\end{array}
\]
\{ 
\{ \text{meet<the, meet<boy, the<boy, the<meet, boy<meet, boy<the} \}, \\
\{ \text{meet<the meet<boy, the<boy, the<meet, boy<meet} \}, \\
\{ \text{meet<the, meet<boy, the<boy} \}, \\
\{ \text{the<meet, boy<meet, the<boy} \}, \\
\{ \text{the<meet, boy<meet, boy<the} \}, \\
\{ \text{meet<the, meet<boy} \}
\}
\]

Many of these sets do not have a trivial expression in a string because they contain inconsistent orderings. The first three are of this kind. Kayne adopts a standard constraint on the output of $L$ that removes these sets from consideration. He calls it Antisymmetry.

(13) Antisymmetry
Ignore any set in $L$ which contains $a<b$ and $b<a$.

There are other sets in $L$ which are unacceptably partial. The string that corresponds to the last set, for instance, does not put boy into the string. Similarly, the second to last set does not fully specify a string: it says that both the and boy should follow meet, but it doesn't specify how the and boy should be ordered with respect to each other. We'll rule both of these options out with what Kayne calls Totality.

(14) Totality
Ignore any set in $L$ which does not contain $a<b$ for every $a$ and $b$ in $P$.

We are left, now, with (15). (I will henceforth suppress the outer brackets of the set that $L$ produces, and merely list the elements of this set.)

(15) $L(\quad \mathrm{VP} \quad ) =$
\[
\begin{array}{c}
V^0 \\
| \\
meet \\
| \\
the \\
| \\
boy \\
\end{array}
\begin{array}{c}
D^0 \\
| \\
NP \\
| \\
\end{array}
\begin{array}{c}
\uparrow \\
DP \\
\uparrow \\
\end{array}
\]
\[
\{ \text{meet<the, meet<boy, the<boy} \} = \text{meet the boy} \\
\{ \text{meet<the, meet<boy, boy<the} \} = \text{meet boy the} \\
\{ \text{the<meet, boy<meet, boy<the} \} = \text{boy the meet} \\
\{ \text{the<meet, boy<meet, the<boy} \} = \text{the boy meet} \\
\{ \text{meet<the, boy<meet, the<boy} \} = ?? \\
\{ \text{meet<the, boy<meet, boy<the} \} = \text{boy meet the} \\
\{ \text{the<meet, meet<boy, the<boy} \} = \text{the boy meet} \\
\{ \text{the<meet, meet<boy, boy<the} \} = ?? \\
\{ \text{meet<the, meet<boy} \}
\]

The two sets here that have no trivial interpretation as a string can be thrown out if we insist that the sets satisfy Transitivity.

(16) Transitivity
Ignore any set in $L$ that doesn't contain $a<c$ if it contains $a<b$ and $b<c$.

We're at:

(17) $L(\quad \mathrm{VP} \quad ) =$
\[
\begin{array}{c}
V^0 \\
| \\
meet \\
| \\
the \\
| \\
boy \\
\end{array}
\begin{array}{c}
D^0 \\
| \\
NP \\
| \\
\end{array}
\begin{array}{c}
\uparrow \\
DP \\
\uparrow \\
\end{array}
\]
\[
\{ \text{meet<the, meet<boy, the<boy} \} = \text{meet the boy} \\
\{ \text{meet<the, meet<boy, boy<the} \} = \text{meet boy the} \\
\{ \text{the<meet, boy<meet, boy<the} \} = \text{boy the meet} \\
\{ \text{the<meet, boy<meet, the<boy} \} = \text{the boy meet} \\
\{ \text{meet<the, boy<meet, the<boy} \} = ?? \\
\{ \text{meet<the, boy<meet, boy<the} \} = \text{boy meet the} \\
\{ \text{the<meet, meet<boy, the<boy} \} = \text{the boy meet} \\
\{ \text{the<meet, meet<boy, boy<the} \} = ?? \\
\{ \text{meet<the, meet<boy} \}
\]
\{the<meet, boy<meet, the<boy\} = \text{the boy meet}

\{meet<the, boy<meet, boy<the\} = \text{boy meet the}

\{the<meet, meet<boy, the<boy\} = \text{the meet boy}

This class is all about that line separating the last two sets from the others. The normal case is for these strings to be absent, and so standard theories prohibit them. I’ll do that with Contiguity.

\begin{align}
(18) & \quad \text{Let } d(P) \text{ be all the terminals dominated by } P. \\
(19) & \quad \text{Contiguity} \\
& \quad a \notin d(P) \rightarrow \forall x, x \in d(P), a < x \text{ or } \forall x, x \in d(P), x < a.
\end{align}

Unlike Antisymmetry, Totality and Transitivity, Contiguity is not a constraint that applies to sets created by \( \mathcal{L} \). It is part of \( \mathcal{L} \) since it relates the phrasal part to the string part. In Kayne (1994), Contiguity is a consequence of how his procedure introduces ordered pairs into the linearization. I want to focus on the effect that Contiguity brings to a linearization, and set aside the other goals of Kayne's project. Those goals are to give a model of language variation, and to capture certain trends, like the correlation between precedence and c-command. (If \( \alpha \) c-commands \( \beta \), then it’s good money that \( \alpha \) will precede \( \beta \).) For this reason, I reify this consequence of Kayne’s system. Any good linearization system must somehow make Contiguity the normal case, and so you can think of Contiguity as that portion of any linearization procedure that is successful. I will represent a \( \mathcal{L} \) that has Contiguity built in with: \( \mathcal{L}_C \).

These are the universal parts of the string part of \( \mathcal{P} \). The language particular side determines the orders of sisters. It is responsible for putting verbs before their objects in English, but in the opposite direction in Japanese. There are patterns to these statements – they are not random – but we’ll start by ignoring those patterns. This component must also be part of \( \mathcal{L} \), as it too looks at the syntax to determine its output. As an expediency, let’s assume that part of what defines a language is a set of language particular phrase ordering rules (\( \mathcal{PO} \)) that applies to a phrase of a particular type and orders its daughter.

\[ \mathcal{PO}(\gamma) = \alpha < \beta \]

We’ll say that a set in \( \mathcal{L}(P) \) satisfies the language particular component if every phrase dominated by \( P \) satisfies the \( \mathcal{PO} \) defined for it.

Together, then, our String part (what I will call a Linearization) is:

\begin{align}
(20) & \quad \mathcal{PO}(\gamma) = \alpha < \beta \\
& \quad \alpha \bigwedge \beta
\end{align}

We’ll say that a set in \( \mathcal{L}(P) \) satisfies the language particular component if every phrase dominated by \( P \) satisfies the \( \mathcal{PO} \) defined for it.

In this case \textit{which lecture} seems to be semantically combining with \textit{left}, as well, perhaps, as with the string that follows it. And, \textit{have} combines semantically with \textit{left}, but not with \textit{we not left}.

The popular alternative is movement. A simple addition to our present system involves one addition and one change. (This is roughly the idea in Ross (1967).)

\begin{align}
(21) & \quad \mathcal{L}_C(P) = \{ S: S = \{ a < b: a \text{ and } b \text{ dominated by } P \} \}, \text{ such that } S \text{ satisfies Antisymmetry, Totality, Transitivity and the language particular component.} \\
& \quad \text{Nothing guarantees that } \mathcal{L}_C(P) \text{ is a singleton (i.e., determines a unique string), but that is the normal case.} \\
& \quad \text{So, in summary, what we have for a model of constituency is (22).} \\
(22) & \quad \mathcal{P} = \mathcal{L}_C(\mathcal{PO}(S_i))
\end{align}

There are many ways of achieving the same goals that this system does, but this uses notions that are familiar (i.e., from Kayne (1994)) and partitions the relevant information in a way that will be convenient for what is to come. An important ingredient is Contiguity. It ensures that there is a simple image of constituency in the strings:

\begin{align}
(23) & \quad \text{If } Q \text{ dominates } P, \text{ then } d(P) \text{ will form a contiguous substring of } d(Q). \\
& \quad \text{recall: } d(X) = \text{the set of terminals dominated by } X.
\end{align}

We must maintain an image of the phrasal structure in the string it maps to. That is the point of \( \mathcal{P} \). (23) is mostly true, and very simple.

But where it is not true is where syntacticians have been furiously busy for decades. One kind of counterexample involves parentheticals.

\begin{align}
(24) & \quad \text{a. John talked, of course, about you.} \\
& \quad \text{b. His father, according to John, is the richest man in Scarsdale.}
\end{align}

The strings of \textit{course} and \textit{according to John} do form a phrase, but don't semantically combine with phrases that correspond to the strings on either side of them. McCawley (1982), citing a precedent in Wells (1947), suggested that these should be seen as violations of Contiguity. Another kind of counterexample is in (25), and these too were suggested to be violations of Contiguity in Engdahl (1980).

\begin{align}
(25) & \quad \text{Which class have we not (yet) left?} \\
& \quad \text{In this case \textit{which lecture} seems to be semantically combining with \textit{left}, as well, perhaps, as with the string that follows it. And, \textit{have} combines semantically with \textit{left}, but not with \textit{we not left}.} \\
& \quad \text{The popular alternative is movement. A simple addition to our present system involves one addition and one change. (This is roughly the idea in Ross (1967).)}
\end{align}

\begin{align}
(26) & \quad \text{a. Copy(\alpha) = \alpha', an exact syntactic and semantic replica of } \alpha. \\
& \quad \text{\alpha is an element of a Stage.} \\
& \quad \text{b. Every element of the Numeration must be semantically interpreted.}
\end{align}
(26b) should replace the restriction in \textsc{merge} that requires the two arguments \textsc{merge} combines to be semantically composed as well. This will allow \textsc{merge} to bring things together that don’t combine semantically, while still preserving the idea that everything that gets \textsc{merge} ‘d into a sentence must be semantically part of that sentence.

This will permit derivations like (27).

(27) a. \[ \begin{array}{c}
D^0 \\
\text{John} \\
talked \\
\text{about} \\
\text{you} \\
\text{course} \\
\end{array} \]

b. \[ \begin{array}{c}
D^0 \\
\text{John} \\
talked \\
\text{about} \\
\text{you} \\
\text{course} \\
\end{array} \]

c. \[ \begin{array}{c}
D^0 \\
\text{John} \\
talked \\
\text{about} \\
\text{you} \\
\text{course} \\
\end{array} \]

d. \[ \begin{array}{c}
D^0 \\
\text{John} \\
\text{VP} \\
\text{PP} \\
\text{PP'} \\
\end{array} \]

I will represent the output of a \textsc{merge} operation that does not involve a semantic composition rule with a dashed branch. The derivation continues beyond this point to make a sentence that includes \textsc{tense} (which I’ve suppressed here) and the subject. We’ll need to add a rule that gets rid of the phrase that was copied.

(28) Copy Deletion
Delete either the copy or the term copied

Both options are needed, as we’ll see. Similar derivations can be made for the other parenthetical and for moving the wh-phrase in (25).

The movement of \textit{have} involves something new. To see this, let’s take a look at a different but related example:

(29) We had not finished.

Here \textit{have} forms a phrase with \textit{finished}, but not \textit{not finished}. The Numeration is:

(30) \[ \begin{array}{c}
D^0 \\
T \\
V^0 \\
\text{Neg}^0 \\
V^0 \\
\text{we} \\
\text{past} \\
\text{have} \\
\text{not} \\
\text{finished} \\
\end{array} \]

Copy will add a copy of \textit{have} and then repeated instances of \textsc{merge} give us:
(31) \[
\begin{array}{c}
\text{DP} \\
\downarrow \\
\text{D}^0 \\
\downarrow \\
\text{we} \\
\end{array}
\begin{array}{c}
\text{T} \\
\downarrow \\
\text{past} \\
\downarrow \\
\text{have} \\
\end{array}
\begin{array}{c}
\text{NegP} \\
\downarrow \\
\text{not} \\
\downarrow \\
\text{V}^0 \\
\end{array}
\begin{array}{c}
\text{V}^0' \\
\downarrow \\
\end{array}
\]

Note that the tense morpheme is not a word. It will have to become part of a tensed verb, and this will be done by **merge** here:

(32) \[
\begin{array}{c}
\text{DP} \\
\downarrow \\
\text{D}^0 \\
\downarrow \\
\text{we} \\
\end{array}
\begin{array}{c}
\text{T}^0 \\
\downarrow \\
\text{have past} \\
\downarrow \\
\text{have} \\
\end{array}
\begin{array}{c}
\text{NegP} \\
\downarrow \\
\text{not} \\
\downarrow \\
\text{V}^0 \\
\end{array}
\begin{array}{c}
\text{V}^0' \\
\downarrow \\
\end{array}
\]

From this point, the derivation merges \(T^0\) with \(\text{NegP}\), and then the resulting TP with the subject DP.

What this derivation shows is that this copy theory of movement allows things to move from one phrase into another, non-dominating, phrase (or word). This is thought to be made necessary by other cases as well. One of these is (33).

(33) Which papers of hers, that Mark brought, has no woman, talked to him about?

This question has an interpretation that allows her to be a variable bound by no woman. In general that is only possible when (34).

(34) If \(\alpha\) is understood as a variable bound by \(\beta\), then \(\alpha\) must c-command \(\beta\). \(\alpha\) c-commands \(\beta\) iff \(\beta\) is dominated by \(\alpha\)’s sister.

We want the which-phrase to be semantically composed with about, putting its contents within the sister of no woman. But (33) can also permit Mark and him to corefer, and that suggests that the wh-phrase is not semantically composed with about, because in (35):

(35) No woman has talked to him about papers of hers that Mark brought.

Mark and him cannot corefer. The copy theory of movement as formulated can almost handle this. We’ll have a derivation that delivers the following Stages.

(36) a. [\(\text{NP}\) papers of hers]
   b. [\(\text{NP}'\) papers of hers]
   c. [\(\text{IP}\) which]
   d. [\(\text{IP}'\) which]
   e. [\(\text{CP}\) that Mark brought]
   f. [\(\text{IP}\) about]
   g. [\(\text{VP}\) talked to him]
   h. [\(\text{IP}\) have]
   i. [\(\text{IP}'\) have]
   j. [\(\text{T}\) pres]
   k. [\(\text{T}'\) pres have’]
   l. [\(\text{C}\) Q]

The next step **merges** [\(\text{NP}'\) papers of her] with the CP.

(37) a. [\(\text{NP}\) papers of hers]
   b. [\(\text{NP}\) [\(\text{NP}'\) papers of hers] [\(\text{CP}\) that Mark brought] ]
   c. [\(\text{IP}\) which]
   d. [\(\text{IP}'\) which]
   e. [\(\text{IP}\) about]
   f. [\(\text{VP}\) talked to him]
   g. [\(\text{IP}\) have]
   h. [\(\text{IP}'\) have]
   i. [\(\text{T}\) pres]
   j. [\(\text{T}'\) pres have’]
   k. [\(\text{C}\) Q]

Then we **merge** the two which’s to the two NPs.

(38) a. [\(\text{DP}\) which papers of hers]
   b. [\(\text{DP}'\) which’ [\(\text{NP}'\) papers of hers] [\(\text{CP}\) that Mark brought] ]
   c. [\(\text{IP}\) about]
   d. [\(\text{VP}\) talked to him]
e. [\(v^0\) have]

f. [\(v^0\) have]

g. [\(t^0\) pres]

h. [\(t^0\) pres have']

i. [\(c\) Q]

From this, \textit{merge} can deliver (39).

(39)

\[
\text{papers of hers} \quad \text{that Mark brought}
\]

\[
\text{which} \quad \text{NP} \quad \text{CP}
\]

\[
\text{D}^0 \quad \text{NP} \quad \text{C}^0 \quad \text{TP}
\]

\[
\text{pres' have'} \quad \text{DP} \quad \text{TP}
\]

\[
\text{no woman} \quad \text{Q} \quad \text{VP}
\]

\[
\text{pres} \quad \text{have'} \quad \text{VP} \quad \text{PP}
\]

\[
\text{talked} \quad \text{P}^0 \quad \text{DP} \quad \text{about which papers of hers}
\]

\[
\text{to} \quad \text{him} \quad \text{NOM} \quad \text{IT} \quad \text{you want} \quad \text{Q}
\]

\[
\text{sleep-NOM} \quad \text{IT} \quad \text{you want} \quad \text{Q}
\]

(40) \quad \text{nŋ\textdegree o} \quad \text{wà} \quad \text{nā} \quad \text{hà} \quad \text{nkà} \quad \text{nŋ\textdegree o} \quad \text{à} \quad \text{\textit{(Vata)}}

\text{sleep} \quad \text{you want} \text{NA} \quad \text{you} \text{FUT-A} \quad \text{sleep} \quad \text{Q}

\text{‘Do you want to sleep?’}

\text{(Koopman 1984, (2a): 154)}

Here “nŋ\textdegree o” (‘sleep’) has been clefted and is pronounced in both the cleft position and the position inside its VP. This is probably movement since it can span long distances, but not every long distance. Indeed, there are a characteristic set of environments which movement relations cannot span: these are John Ross’s famous “islands” for movement. One of these environments are relative clauses. The clefted verb in Vata cannot span a relative clause.

(41) * tãkã ǹ wà fōtō’ mōmō’ ǹ tãkã bō abbix show you like picture ITTT \quad you showed REL Aba

\text{‘It’s show that you like the picture you showed Aba.’}

\text{(Koopman 1984, (15): 159)}

Also the two verbs involved must be the same. They are inflected differently as you can see, but it isn’t possible to change either of the verbs that bear the inflection. They really do look like copies.

When a verb clefts in Vata, both copies must be pronounced. (40) is ungrammatical if either verb isn’t pronounced. When nominal material clefts, by contrast, only the higher copy may be pronounced. (42) illustrates.

(42) nŋ\textdegree oūlī ǹ mĩ hà ǹ sleep-NOM IT you want

\text{‘Is it sleeping you want?’}

\text{(Koopman 1984, (2b): 154)}

When verbs, or predicates, move, \textit{Copy Deletion} is sometimes blocked.

I don’t know of anything similar, though, when a phrase moves. When DPs move, for instance, both DPs are never pronounced. What does happen, however, is that one of the DPs is “replaced” by a pronoun: a resumptive pronoun. For instance, in Lebanese Arabic there are resumptive pronouns that seem to stand in for a moved phrase when islands aren’t violated. (See Sichel 2014 and references therein.)

(43) talmiiiz-ạ́ ikosleen ma baddna ṇyabbī [wala ṃỵallme] hitting the bad NEG want.1P tell.1P [no teacher] that

\text{he cheated.3SM in-the-exam}

‘her bad student, we don’t want to tell any teacher that he cheated on the exam.’
I think, then, that there is a difference in how Copy Deletion applies, depending on the type of thing being moved. I’ll assume that there is a difference between moved XPs and moved other things that is responsible for how Copy Deletion is relaxed.

A good theory of movement is one that links together the two properties we have seen in these examples:

(45)  
   a. Semantic Displacement: A term is sometimes not in the place Contiguity requires.  
   b. Terseness: The term that violates Contiguity is usually only pronounced once.

We also need a semantics that can handle situations like (33).

An hypothesis that many are exploring is that (45) can be achieved by more directly fiddling with Contiguity. One possibility is to change our phrase part of $\mathcal{P}$ so that derivations like the following are permitted. I’m going to take a few, very small, steps in that direction, building on an idea about what movement is that was in an early unpublished manuscript by Stanley Peters and Robert Richie, carried forward by Engdahl (1980) and has now found many proponents, including Gärtner (1997), Starke (2001), Nunes (2001), Frampton (2004), Citko (2005), Koble (2006) and de Vries (2007). That idea is that MERGE can give an expression two positions by re-merging it.

(46)  
   (She asked) which book he knows.

You can see how (46e) provides a way of capturing Semantic Displacement. The moved term — here which book — is syntactically in two positions and so its denotation has two positions where it can be applied. Instead of a copy, then, a moved phrase is just the same phrase in two spots. The differences in how Semantic Displacement arise are going to come about, I will claim, from the particular ways in which the expressions that are “moved” get broken up into two different positions.

One way of allowing $\text{MERGE}$ to produce (46e) is to relax the Extension Condition. We could allow $\text{MERGE}$ to see parts of phrases. This is called the Remerge Theory of movement. In these lectures, I want to examine a different possibility. I will suggest that Remerge derivations are blocked – I will keep the Extension Condition. Instead, I will suggest that derivations tolerate what Citko (2005) calls “parallel merge.” These look like:
Forcing movement to arise only through parallel movement might give us a handle on a third property that a good theory of movement should link to Semantic Displacement and Terseness. That's that movement is subject to island constraints. Suppose islands are those phrases at which a phonological or semantic evaluation must take place. This might be what "phases" are. If islands are phases that arise at the point in a derivation where there remain two roots, as in any but the last of the steps in (47), then arguably a semantic and phonological evaluation will not be possible, as our semantics and phonology are (perhaps) defined only for representations with a single root. This, at any rate, is one way of thinking about how to seek an answer to the question of why islands emerge uniquely with movement on an account that uses multidominant representations. I do not see how to bring this idea off. Nonetheless, it is important to have a way of stating what an island is in the present framework. I'll adopt (50).

An Island is a Stage in which every element must have one root.

I'll have nothing more to say about islands in these classes. I'll focus on the remaining two properties: terseness and semantic displacement. I think it's clear how the multidominant interpretation of copies provides a way of modeling semantic displacement. It's not clear how the right meanings are delivered, of course, and that will take some work. We'll investigate that issue in some detail in the last three lectures of this series. But we'll start by examining how to capture terseness – the property of movement that inclines the material that has two positions in the phrasal part of a sentence to have only one position in the resulting string. On the program that Nunes sets out, this is the result of how linearization works, and I'll join that program. On his way of executing the idea, a linearization forces the material that occupies two positions in the phrasal portion to occupy two positions in the string, in violation of Antisymmetry. This illegal linearization is made legal by a special purpose-built rule of deletion, that removes that material from one of the positions it occupies in the string. Expressing this deletion operation in a multidominance theory of copies will require something different than what Nunes offered. Unlike the copy theory that Nunes relied on, there are not two objects – a thing and its copy – that can be distinguished by the deletion rule in a multidominant representation. In the next class, I'll look at how to design a linearization procedure that builds in Nunes's deletion operation, and makes use of multidominance.
References


One of the important properties I suggested yesterday that we should take as diagnostic of movement – one of the properties that should emerge from our syntax of movement – is what I called “Terseness.”

(1) Terseness
Only one of the two phrasal positions that a moved term is related to is mapped onto the string that is spoken.

a. exception 1: sometimes moved verbs can be spoken in both positions
b. exception 2: sometimes moved DPs can have a pronoun spoken in the lower position

First, recall the phrase part of our theory. It involves a serial process of putting parts of sentences together into phrases.

(2) Derivation
\[ \mathcal{D}(S_i), S_i \text{ a Numeration, is the series of Stages } (S_i, S_{i+1}, S_{i+2}, \ldots, S_m), \text{ where:} \]
a. Each \( S_{k+1} \) is derived from \( S_{k-1} \) by one application of \( \text{MERGE} \) or \( \text{COPY} \).
b. \( S_m \) is a singleton set (i.e., a root node)

(3) \[ \text{MERGE}(\alpha, (\beta)) = \frac{\gamma}{\alpha \quad (\beta)} \]

where:

a. \( \alpha \) and \( \beta \) are root nodes in (i.e., elements of) a Stage.
b. If one of the arguments of \( \text{MERGE} \) is a word, then \( \gamma \) must be a phrase.
c. If \( \text{MERGE} \) has a sole argument, it must not be a phrase.
d. \( \gamma \) is the category of \( \alpha \) or \( \beta \).
e. \( \alpha \) and \( \beta \) must be certain categories.

(4) \( \text{COPY}(\alpha) = \alpha' \), an exact syntactic and semantic replica of \( \alpha \).

(5) Principle of Full Interpretation
If \( \alpha \) is a member of a Numeration, \( S_i \), then \([\alpha]\) must be part of the denotation of the output of \( \mathcal{D}(S_i) \).

This enforces a serial application of \( \text{MERGE} \) and \( \text{COPY} \), and it allows for one of the two terms in a copy relation to not be semantically interpreted. That is, it decouples \( \text{MERGE} \) from the requirement that the terms \( \text{MERGE} \) brings together must be semantically combined. It places the requirement that everything with a denotation contribute its denotation to the sentence its part of in a separate condition, which I’ve called here the Principle of Full Interpretation.

This system produces derivations like (6). (From now on I will take the liberty of representing derivations incompletely. I will give just those stages that I feel are sufficient for the reader to envision how the entire series looks.)

(6) (She asked) which book he will read.

\[
\begin{array}{llllllll}
\text{a. } & \text{C} & \text{D} & \text{N} & \text{D} & \text{T} & \text{V} \\
& \text{Q} & \text{which} & \text{book} & \text{he} & \text{will} & \text{read} \\
\text{b. } & \text{DP} & \text{DP'} & \text{C} & \text{DP} & \text{T} & \text{V} \\
& \text{D} & \text{NP} & \text{D'} & \text{NP'} & \text{Q} & \text{D} & \text{will} & \text{read} \\
& \text{which N} & \text{which N'} & \text{he} \\
& \text{book} & \text{book} \\
\text{c. } & \text{TP} & \text{VP} & \text{DP'} & \text{C} & \text{DP} \\
& \text{T'} & \text{VP} & \text{D'} & \text{NP'} & \text{Q} & \text{D} \\
& \text{will V} & \text{which book} & \text{he} \\
& \text{read} & \text{book} & \text{book}
\end{array}
\]
This representation allows Terseness to arise from an operation that "deletes" one of the two phrases in the copy relation.

(7) Copy Deletion
Delete α or its copy.

What Terseness amounts to, then, is the observation that (with the exceptions noted) this deletion process is obligatory. I'll sketch the way this is done in Nunes (2004, chapter 1), which is a reworked version of his 1995 University of Maryland dissertation, and an improved version of Nunes (1999), and then I'll modify it so it works with phrase markers with multidominance in them.

Recall now the string part of our theory.

(8) \( \mathcal{L}_C(P) = \{ S : S = [a < b : a \text{ and } b \text{ dominated by } P] \} \), such that S satisfies Antisymmetry, Totality, Transitivity, Contiguity, and the language particular component.

a. **Totality**
If \( a < b \) is in \( P \), then \( a < b \) is in \( S \).

b. **Antisymmetry**
\( \neg (a < b \land b < a) \)

c. **Transitivity**
\( (a < b \land b < c) \Rightarrow a < c \).

d. Let \( d(P) \) be all the terminals dominated by \( P \).

e. **Contiguity**
\( a \notin d(P) \Rightarrow \forall x, x \in d(P), a < x \) or \( \forall x, x \in d(P), x < a \).

Applied to (6e), this will give us:

(9)
\[
\begin{align*}
&\text{which'}<\text{book'} \quad \text{book'}<\text{Q} \\
&\text{C}<\text{he} \quad \text{he}<\text{will} \quad \text{will}<\text{read} \quad \text{read}<\text{which} \\
&\text{which'}<\text{Q} \quad \text{book'}<\text{he} \\
&\text{Q}<\text{will} \quad \text{will}<\text{which} \quad \text{read}<\text{book} \\
&\text{which'}<\text{he} \quad \text{book'}<\text{will} \\
&\text{Q}<\text{read} \quad \text{will}<\text{book} \\
&\text{which'}<\text{will} \quad \text{book'}<\text{read} \\
&\text{Q}<\text{which} \\
&\text{which'}<\text{read} \quad \text{book'}<\text{which} \\
&\text{Q}<\text{book} \\
&\text{which'}<\text{which} \quad \text{book'}<\text{book} \\
&\text{which'}<\text{book} \\
\end{align*}
\]

\( \equiv \) which' book' Q he will read which book

Nunes suggests that Copy Deletion can be forced by modifying Antisymmetry so that it cannot tell a copy apart from the thing it is a copy of.

(10) **Antisymmetry**
Let \( \alpha' \) be a copy of \( \alpha \), then a linearization cannot have \( x < \alpha' \) or \( x < \alpha \) and also either of \( \alpha < x \) or \( \alpha' < x \).

This makes (9) have multiple Antisymmetry violations. Copy Deletion is intended as a method of repairing (9) so that it satisfies Antisymmetry. Nunes calls his rule Chain Reduction. We can use our machinery to express Chain Reduction as:

(11) **Chain Reduction**
Chain Reduction applied to \( d(X) \) deletes every ordered pair in a linearization that contains a word in \( d(X) \).

Applied to (9), this can rescue Antisymmetry by removing ordered pairs in a variety of ways. If we represent those various sets with the strings they correspond to, Chain Reduction can produce from (9) the Antisymmetry obeying sets in (12).
(12)  a. which book should he read  
    b. which should he read book  
    c. book should he read which  
    d. should he read which book

To form the strings in (12), Chain Reduction will delete from (9) the ordered pairs indicated in (13).

(13)  a. To form (12a), Chain Reduction applies to d(DP).
    b. To form (12b), Chain Reduction applies to d(NP) and d(D0).
    c. To form (12c), Chain Reduction applies to d(D0) and d(NP).
    d. To form (12a), Chain Reduction applies to d(DP').

Nunes assumes, and so shall I, that (12a) and (12d) are possible outcomes – some languages choosing one or the other – but that (12b) and (12c) are not. To block these two outcomes, Nunes adopts:

(14)  Economy
Let |CR| be the number of applications of Chain Reduction to a Linearization. If Antisymmetry can be satisfied by CR, then all other applications of Chain Reduction, CR', are ungrammatical if |CR'| > |CR|.

Economy will block the applications of Chain Reduction in (13b) and (13c) because of the equally Antisymmetry compliant applications of Chain reduction in (13a) and (13d).

Consider now how this system will work for Head Movement. Let's consider first a case where the moved verb does get spoken in both of its positions.

(15)  lì ʒ dà sákà lì  
    eat she/he perf rice eat  
    'she/he has eaten rice.'

(16)  |FOC|
    |V0| Foc  
    TP
    dà
    VP |
    V0

It's not clear in this particular example, but the clefted verb and the verb downstairs are not inflected the same way. The clefted verb is always inflected with mid-tones, no matter what the lexical tones of the moved verb are. This was clearer in the example I gave you yesterday:

(17)  ngônô n wà nā h kā ngônô á (Vata)  
    sleep you want NA you FUT-A sleep Q  
    'Do you want to sleep?'
    (Koopman 1984, (2a): 154)

This is very characteristic of the cases of verb movement where both copies are pronounced. The pronounced verbs have different inflections.

Nunes' system makes this relevant. He suggests that the things that are linearized by L_C are just words. In our terms:

(18)  L_C(P) = {S: S - {a<b: a and b are X^0 dominated by P} }, such that S satisfies Antisymmetry, Totality, Transitivity and the language particular component.

This will cause L_C to assign (16) the linearization in (19).

(19)  
    { Foc^0 < ʒ  ʒ < dà  dà < sákà  sákà < lì  
      Foc^0 < dà  ʒ < sákà  dà < lì  
      Foc^0 < sákà  ʒ < lì  
      Foc^0 < lì  
    }

This linearization satisfies Antisymmetry as well. When Foc^0 is matched against the vocabulary item that spells out “lì+Foc,” the correct string associated with (16) is produced.
This is Nunes’s system. If it is a successful derivation of Terseness, then what we’d want to do is find an explanation for why Antisymmetry treats copies as the same thing that they are copied from. I suggest that we consider this evidence for multidominance.

As foreshadowed yesterday, we must a way of changing our phrase part to allow for a term to be in two different positions. And then we need to think about how to relax Contiguity to allow for these cases to get linearized. I proposed yesterday that we leave everything about merge intact, but we allow it to apply numerous times in mapping one representation to another in the derivation.

\[
\text{Merge}(\alpha, (\beta)) = y, \quad \alpha \quad (\beta)
\]

where:

a. \(\alpha\) and \(\beta\) are elements of a Stage.

b. If one of the arguments of Merge is a word, \(y\) must be a phrase.

c. The sole argument of Merge cannot be a phrase.

d. \(y\) is the category of \(\alpha\) or \(\beta\).

e. \(\alpha\) and \(\beta\) must be certain categories.

\[
\mathcal{D}(S_i), S_i \text{ an initial Numeration, is the series of Stages } (S_i, S_{i+1}, S_{i+2}, \ldots, S_m), \text{ where:}
\]

a. each \(S_{k+1}\) is derived from \(S_{k-1}\) by at least one application of Merge, and

b. \(S_m\) is a single phrase (i.e., a singleton set).

This will allow for derivations like (22).

\[
\begin{align*}
\text{c. } & \quad \text{DP} \quad T^0 \quad \text{Foc}^0 \quad \text{VP} \\
& \quad \text{D}^0 \quad \text{dā} \quad \text{MT} \quad \text{li} \quad \text{N}^0 \quad \text{Foc}^0 \\
& \quad \quad \text{N}^0 \quad \text{li} \\
& \quad \quad \text{sākā}
\end{align*}
\]

\[
\begin{align*}
\text{d. } & \quad \text{DP} \quad \text{TP} \quad \text{Foc}^0 \\
& \quad \text{D}^0 \quad \text{T}^0 \quad \text{V}^0 \quad \text{VP} \quad \text{Foc} \\
& \quad \text{dā} \quad \text{NP} \quad \text{V}^0 \quad \text{li} \\
& \quad \text{N}^0 \quad \text{li} \\
& \quad \text{sākā}
\end{align*}
\]

\[
\begin{align*}
\text{e. } & \quad \text{TP} \\
& \quad \text{DP} \quad \text{TP} \quad \text{Foc}^0 \\
& \quad \text{D}^0 \quad \text{T}^0 \quad \text{V}^0 \quad \text{VP} \quad \text{Foc} \\
& \quad \text{dā} \quad \text{NP} \quad \text{V}^0 \quad \text{li} \\
& \quad \text{N}^0 \quad \text{li} \\
& \quad \text{sākā}
\end{align*}
\]
The step from the Stage in (22b) to the Stage in (22c) involved \texttt{merge} applying both to \texttt{V}^0+\texttt{NP} and \texttt{V}^0+\texttt{Foc}. All three of these terms are elements of that Stage – that is they are all root nodes – and so they are among the things that \texttt{merge} can operate on. Note, then, that these applications of \texttt{merge} happen “simultaneously.” That is, applications of \texttt{merge} are not serial. It’s derivations that define the serial nature of sentence construction.

Just as on the Copy Theory of Movement, \texttt{ZC} will produce (19), which is the only linearization that meets Totality and Transitivity, and it will satisfy Antisymmetry for the same reason it does in the copy theory.

Let’s consider next how this system will produce wh-movement. Reconsider (6). This will now go as in (23).

(6) (She asked) which book he will read.

(23)  a. \begin{tabular}{c|c|c|c|c|c|c|c}
\texttt{X}^0 & \texttt{C}^0 & \texttt{D}^0 & \texttt{N}^0 & \texttt{D}^0 & \texttt{T}^0 & \texttt{V}^0 \\
\hline
\texttt{?} & \texttt{Q} & \texttt{which} & \texttt{book} & \texttt{he} & \texttt{will} & \texttt{read} \\
\end{tabular}

b. \begin{tabular}{c|c|c|c|c|c|c|c}
\texttt{X}^0 & \texttt{C}^0 & \texttt{D}^0 & \texttt{NP} & \texttt{DP} & \texttt{T}^0 & \texttt{V}^0 \\
\hline
\texttt{?} & \texttt{Q} & \texttt{which} & \texttt{N}^0 & \texttt{D}^0 & \texttt{will} & \texttt{read} \\
\end{tabular}
I had to make up $X^0$ to produce this. There's no other way our definition of Merge will allow multidominance here. For the program I am advancing to be successful, then, we must identify $X^0$.

The linearization that $\mathcal{L}_C$ produces for this is:

\[(24)\] Q he will read which book ?

This satisfies Totality, Antisymmetry, Transitivity and Contiguity. But it violates the language particular component; English requires that XP precede its sister. There is no way of achieving that goal here and preserving Contiguity. That will be our next task.

But first, consider how $\mathcal{L}$ (the Contiguity free version) derives Terseness. Suppose that $\mathcal{L}$ were to produce a linearization of (23f) that corresponds to (25).

\[(25)\] ? which book Q he will read which book

This violates Antisymmetry. But Totality doesn't require that which and book be linearized according to every position they occupy. It only requires that they be linearized according to (at least) one of those positions. On this revision to Nunes' system, we both explain why a thing and its copy are treated as if they were indistinguishable by Antisymmetry (they are indistinguishable) and we don't need Chain Reduction. Terseness emerges from the string component unaided.

We need to relax Contiguity to allow for the right linearization of (23f). When we do this, we need to preserve Contiguity's effects for all of the phrases that aren't in two positions. And we also need to make sure Contiguity makes all of the moved phrase hang together as a contiguous string. We want to allow (26a) and block (26b).

\[(26)\] a. ? which book Q he will read
b. ? which Q he will read book

I pointed out yesterday that Contiguity projects a simple map of constituency into the string it forms. That map can be described with (27).

\[(27)\] If phrase $XP_1$ dominates phrase $XP_2$, then the words in $XP_2$ (i.e., $d(XP_2)$) will form a contiguous substring of the string formed by the words in $XP_1$ (i.e., $d(XP_1)$).

Indeed, the transitive closure of (27) holds for phrase markers that obey Contiguity and don't contain multidominance.

\[(28)\] Let $p=(XP_1, XP_2, \ldots, XP_n)$ be a series of phrases such that every $XP_i$ in $p$ is dominated by every $XP_{j>i}$ in $p$. For every $p$ in a phrase marker, $d(XP_i)$ must be a contiguous substring of $d(XP_{j>i})$ for every $XP$ in $p$.

I will call a series of phrases that form a $p$, a "path."

Interestingly, (28) isn't obeyed in a phrase-marker that allows for multidominant representations. To see this, consider (29) and the linearization of (29) that corresponds to overt movement, in (30).

\[(29)\]
(30) overt movement linearization:
    which flower Q she should bring here

Two paths that contain DP and NP in (29) are (31). (Note that "dominance" is reflexive.)

(31) a. paths for NP:
    (NP, DP, VP†, VP, TP†, TP, CP†, CP)
    (NP, DP, XP, CP)

b. paths for DP:
    (DP, VP†, VP, TP†, TP, CP†, CP)
    (DP, XP, CP)

(30) makes (31a-i) and (31b-i) violate (28); neither flower (=d(NP)) nor which flower (=d(DP)) are contiguous substrings of d(TP) (=she should bring which flower here), d(TP†) (=should bring which flower here), d(VP) (=bring which flower here) or d(VP†) (=bring which flower here). If Contiguity were to be expressed in a way that derives (28), then only covert movement operations would be permitted. That's not a desirable outcome.

Notice, however, that if the paths in (31a-i) and (31b-i) are ignored, the linearization in (30) doesn't violate (28). Conversely, the paths in (31a-ii) and (31b-ii) violate (28) if the linearization is (32).

(32) Q she should bring which flower here

Under this linearization, neither d(NP) (=flower) nor d(DP) (=which flower) are contiguous substrings of d(XP) (=which flower). This linearization doesn't violate (28), however, if the paths in (31a-ii) and (31b-ii) are ignored. Paths give us a way, then, of linearizing a phrase that is in two positions in either one of those positions. We can use paths to make movement overt or covert.

The linearization algorithm I will propose is based on paths. As we've seen, framing Contiguity in terms of paths in the way that (28) does leaves its effects unchanged for phrase markers that don't have multidominance in them, but has useful effects in situations where multidominance arises. Words will get into a linearization by virtue of the paths they have, and so I will state Totality in terms of paths too. This will also allow a phrase marker that has multidominance, and therefore more than one path for a word or group of words, to satisfy Totality by choosing just one of those paths. Finally, because the formalism for representing linearizations is a set of ordered pairs, (28) will have to be expressed in a way that references those ordered pairs rather than the strings they correspond to. Here, then, is a system that does those things.

(33) Linearization

\( \mathcal{L}_{PC}(P) = \{ S : S = \{ \alpha < \beta : \alpha, \beta \text{ are } X^0 \text{s in } P \} \}, \) such that S obeys Path Contiguity, Totality, Antisymmetry, Transitivity, and the language particular component.

(34) a. Let \( p(w)=(XP_1, XP_2, \ldots, XP_n) \), a path, be the phrases that dominate \( w \), an \( X^0 \), and include the root phrase, where every \( XP_i \) in \( p \) is dominated by every \( XP_j \) in \( p \).

b. \( \Pi(P) \) is a set of paths formed from the words in \( P \).

c. \( d(XP) \) is the set of \( w \)s such that \( XP \) is in \( p(w) \).

d. Path Contiguity

If \( p \in \Pi \), then for every \( XP \in p \) and YP, sister of XP:

i. \( \beta \in d(YP) \rightarrow \forall \alpha \in d(XP) \alpha < \beta \), or

ii. \( \beta \in d(YP) \rightarrow \forall \alpha \in d(XP) \beta < \alpha \).

e. Totality

For every \( w \) in \( P \), \( \Pi(P) \) must contain \( p(w) \).

f. Antisymmetry

\( \neg (\alpha < \beta \land \beta < \alpha) \)

g. Transitivity

\( (\alpha < \beta \land \beta < \gamma) \rightarrow \alpha < \gamma \).

Totality requires that every word in a sentence be associated with a path that is used to linearize it. If, then, is at minimum a set of paths, one for each word. For each of these paths, Path Contiguity then ensures that Contiguity-preserving ordered pairs for each word is in the linearization. Path Contiguity doesn't make the language particular correct choices – that must come from a part of the linearization scheme that fixes the choices among the cross-linguistic word-orders – but it limits those choices to just ones that preserve Contiguity relative to a word's path. Antisymmetry and Transitivity do the familiar jobs of weeding out those orderings that don't have an interpretation as a string. In fact, because Path Contiguity now requires every word in every \( d(XP) \) to be ordered in the same way to every word in \( XP \)'s sister, Transitivity is ensured by a combination of Totality and Path Contiguity. We needn't state Transitivity as an independent constraint on a linearization, and I shall henceforth drop it.

We'll look at two case studies to see how Path Contiguity does its job. Consider first a vanilla phrase-marker with no multidominance.
For each of the words in (35), there is only one path. Consequently, the only \( \Pi \) that satisfies Totality is (36).

(36) a. \( p(\text{she}) = \{\text{DP}, \text{TP}\} \)
    b. \( p(\text{should}) = \{\text{TP}^\dagger, \text{TP}\} \)
    c. \( p(\text{protest}) = \{\text{VP}, \text{TP}^\dagger, \text{TP}\} \)

From these paths, we can calculate \( d \), which relates phrases to the words that must be in the linearization in order to comply with Path Contiguity. The \( d \) of a phrase is all the words that contain that phrase in its path.

(37) a. \( d(\text{TP}) = \{\text{she}, \text{should}, \text{protest}\} \)
    b. \( d(\text{DP}) = \{\text{she}\} \)
    c. \( d(\text{TP}^\dagger) = \{\text{should}, \text{protest}\} \)
    d. \( d(\text{VP}) = \{\text{protest}\} \)

Path Contiguity requires that each of the sets in (37) map onto a contiguous substring in the linearization. For instance, for Path Contiguity to hold of \( \text{TP}^\dagger \), all of the words in \( d(\text{TP}^\dagger) \) (i.e., \( \text{should} \) and \( \text{protest} \)) must be ordered in the same way to the words in \( \text{TP}^\dagger \)'s sister: \( d(\text{DP}) \) (i.e., \( \text{she} \)). Every phrase that is in some word’s path will be subject to this requirement, and so every word will be part of a series of phrases that are contiguous, each larger phrase in that path mapping onto a larger contiguous superstring containing that word.

Path Contiguity therefore allows for the linearizations of (35) in (38).

(38) a. \( \text{she should protest} \)
    b. \( \text{should protest she} \)
    c. \( \text{she protest should} \)
    d. \( \text{protest should she} \)

This is probably more possibilities than should be allowed – (38d) is a sufficiently rare way for a language to linearize this structure that we might want to block it – but it comes close to what’s cross-linguistically available. I will assume that the language particular choices narrow this set down to the outcomes appropriate for any particular language. English (a head initial, Specifier initial language) chooses (38a).

The second case study is (39).

As we’ve seen, \( \text{which} \) and \( \text{flower} \) have two paths in (39), and so the largest \( \Pi \) contains them both:

(40) a. \( p(\text{which}) = \{\text{DP}, \text{VP}^\dagger, \text{VP}, \text{TP}^\dagger, \text{TP}, \text{CP}^\dagger, \text{CP}\} \)
    b. \( p(\text{which}) = \{\text{DP}, \text{XP}, \text{CP}\} \)
    c. \( p(\text{flower}) = \{\text{NP}, \text{DP}, \text{VP}^\dagger, \text{VP}, \text{TP}^\dagger, \text{TP}, \text{CP}^\dagger, \text{CP}\} \)
    d. \( p(\text{flower}) = \{\text{NP}, \text{DP}, \text{XP}, \text{CP}\} \)
    e. \( p(\text{bring}) = \{\text{VP}^\dagger, \text{VP}, \text{TP}^\dagger, \text{TP}, \text{CP}^\dagger, \text{CP}\} \)
    f. \( p(\text{here}) = \{\text{PP}, \text{VP}, \text{TP}^\dagger, \text{TP}, \text{CP}^\dagger, \text{CP}\} \)
    g. \( p(\text{should}) = \{\text{TP}^\dagger, \text{TP}, \text{CP}^\dagger, \text{CP}\} \)
    h. \( p(\text{she}) = \{\text{DP}^\dagger, \text{TP}, \text{CP}^\dagger, \text{CP}\} \)
    i. \( p(\text{Q}) = \{\text{CP}^\dagger, \text{CP}\} \)
    j. \( p(?) = \{\text{XP}, \text{CP}\} \)
The values for \( d \) are:

(41) a. \( d(CP) = \{Q, she, should, bring, here, ?, which, flower\} \)
    b. \( d(XP) = \{?, which, flower\} \)
    c. \( d(CP^\dagger) = \{Q, she, should, bring, here, which, flower\} \)
    d. \( d(TP) = \{she, should, bring, here, which, flower\} \)
    e. \( d(DP^\dagger) = \{she\} \)
    f. \( d(TP^\dagger) = \{should, bring, here, which, flower\} \)
    g. \( d(VP) = \{bring, here, which, flower\} \)
    h. \( d(VP^\dagger) = \{bring, which, flower\} \)
    i. \( d(DP) = \{which, flower\} \)
    j. \( d(NP) = \{flower\} \)

Path Contiguity prevents almost all linearizations of (40). It allows a linearization for this \( \Pi \) only under very narrow circumstances: when the language's word order settings would allow the multidominant phrase to be simultaneously contiguous to the sisters it has in both of its positions. Because of (41b), Path Contiguity requires the linearization to have a contiguous string made from \( ?, which \) and \( flower \). But because of (41g) and (41h), it also requires contiguous substrings made from \( \{bring, which, flower\} \) and \( \{bring, which, flower, here\} \), which means the linearization must have one of the strings in (42) in it.

(42) a. i. bring which flower here
    ii. bring flower which here
    b. i. here bring which flower
    ii. here bring flower which
    c. i. which flower bring here
    ii. flower which bring here

The strings in (42a) can't coexist in a linearization that also puts \( ? \) contiguous with \( \{which, flower\} \). That's because Path Contiguity will also require that \( ? \) be ordered in the same way with \( here \) that it is with \( bring \). The strings in (42b) and (42c) can survive Path Contiguity if nothing in larger phrases separates \( ? \) from \( which, flower \). For instance, the strings in (43) would satisfy Path Contiguity.

(43) a. \( ? \) which flower bring here should she Q
    b. Q she should here bring which flower ?

I don't know of such a case, but I don't know of any harm in letting in this possibility. If (43) don't arise in natural languages, then I propose they are blocked by the language particular component, not by Contiguity. In general, (40) is too large to have a viable outcome. A smaller \( \Pi \) will have to be chosen.

There are four other \( \Pi \)s that satisfy Totality. They all give to \( which \) and \( flower \) just one path. One such \( \Pi \) chooses paths for \( which \) and \( flower \) that go through XP; another chooses paths for \( which \) and \( flower \) that go through VP\(^\dagger\) instead. The first of these is (44) and the second (45).

(44) a. \( p(which) = \{DP, XP, CP\} \)
    b. \( p(flowers) = \{NP, DP, XP, CP\} \)
    c. \( p(bright) = \{VP\^\dagger, VP, TP\^\dagger, TP, CP\^\dagger, CP\} \)
    d. \( p(here) = \{PP, VP, TP\^\dagger, TP, CP\^\dagger, CP\} \)
    e. \( p(should) = \{TP\^\dagger, TP, CP\^\dagger, CP\} \)
    f. \( p(she) = \{DP\^\dagger, TP, CP\^\dagger, CP\} \)
    g. \( p(Q) = \{CP\^\dagger, CP\} \)
    h. \( p(?) = \{XP, CP\} \)

(45) a. \( p(which) = \{DP, VP\^\dagger, VP, TP\^\dagger, TP, CP\^\dagger, CP\} \)
    b. \( p(flowers) = \{NP, DP, VP\^\dagger, VP, TP\^\dagger, TP, CP\^\dagger, CP\} \)
    c. \( p(bright) = \{VP\^\dagger, VP, TP\^\dagger, TP, CP\^\dagger, CP\} \)
    d. \( p(here) = \{PP, VP, TP\^\dagger, TP, CP\^\dagger, CP\} \)
    e. \( p(should) = \{TP\^\dagger, TP, CP\^\dagger, CP\} \)
    f. \( p(she) = \{DP\^\dagger, TP, CP\^\dagger, CP\} \)
    g. \( p(Q) = \{CP\^\dagger, CP\} \)
    h. \( p(?) = \{XP, CP\} \)

The \( ds \) for (44) are in (46), and they correspond to the string in (47) in a head-initial and Specifier-initial language like English.

(46) a. \( d(CP) = \{Q, she, should, bring, here, ?, which, flower\} \)
    b. \( d(XP) = \{?, which, flower\} \)
    c. \( d(CP^\dagger) = \{Q, she, should, bring, here\} \)
    d. \( d(TP) = \{she, should, bring, here\} \)
    e. \( d(DP^\dagger) = \{she\} \)
    f. \( d(TP^\dagger) = \{should, bring, here\} \)
    g. \( d(VP) = \{bring, here\} \)
    h. \( d(VP^\dagger) = \{bring\} \)
    i. \( d(DP) = \{which, flower\} \)
    j. \( d(NP) = \{flower\} \)

(47) \( ? \) which flower Q she should bring here
The $d$s for (45) are in (48), and they correspond to the string in (49), in a head-initial, Specifier-initial language.

(48)  
   a. $d$(CP) = \{Q, she, should, bring, here, ?, which, flower\}  
   b. $d$(XP) = \{?\}  
   c. $d$(CP†) = \{Q, she, should, bring, here, which, flower\}  
   d. $d$(TP) = \{she, should, bring, here, which, flower\}  
   e. $d$(DP) = \{she\}  
   f. $d$(TP†) = \{should, bring, here, which, flower\}  
   g. $d$(VP) = \{bring, here, which\}  
   h. $d$(VP†) = \{bring, which\}  
   i. $d$(DP) = \{which, flower\}  
   j. $d$(NP) = \{flower\}

(49)  
   ? Q she should bring which flower

These are the desired outcomes; they correspond to the overt and covert movement possibilities.

The remaining two IIs that satisfy Totality give to which and flower divergent paths. They are both blocked by Path Contiguity. To see how, consider (50), where flower is given a path through XP and which is given a path through VP†.

(50)  
   a. $p$(which) = \{DP, VP†, VP, TP†, TP, CP†, CP\}  
   b. $p$(flower) = \{NP, DP, XP, CP\}  
   c. $p$(bring) = \{VP†, VP, TP†, TP, CP†, CP\}  
   d. $p$(here) = \{PP, VP, TP†, TP, CP†, CP\}  
   e. $p$(should) = \{TP†, TP, CP†, CP\}  
   f. $p$(she) = \{DP†, TP, CP†, CP\}  
   g. $p$(Q) = \{CP†, CP\}  
   h. $p$(?) = \{XP, CP\}

The $d$s for (50) are (51).

(51)  
   a. $d$(CP) = \{Q, she, should, bring, here, ?, which, flower\}  
   b. $d$(XP) = \{?, flower\}  
   c. $d$(CP†) = \{Q, she, should, bring, here, which\}  
   d. $d$(TP) = \{she, should, bring, here, which\}  
   e. $d$(DP) = \{she\}  
   f. $d$(TP†) = \{should, bring, here, which\}  
   g. $d$(VP) = \{bring, here, which\}  
   h. $d$(VP†) = \{bring, which\}  
   i. $d$(DP) = \{which, flower\}  
   j. $d$(NP) = \{flower\}

d(VP†) and $d$(VP) together require that the linearization produce the string bring which here (once English-specific choices are made). But $d$(DP) requires that the linearization also produce the string which flower. There is no way of linearizing these words that preserves these two requirements. Exactly the same incompatibility arises if the path for flower goes through VP† and the path for which goes through XP – the other way of choosing divergent paths for these words. The reason these choices lead to a conflict is because all choices of paths for which and flower will contain DP, and Path Contiguity will consequently require which and flower to be contiguous. This is how this system prevents the words in a moved phrase from getting linearized in different positions.

Path Contiguity, then, allows for both overt and covert movement and explains why multidominant structures allow for a selective relaxation of Contiguity. It makes Contiguity the driving force behind a linearization. The formalization of Contiguity involved enforces a particular kind of “nesting” condition on entire phrase markers. It allows multidominance in just those cases where that nesting condition can be satisfied for every word in the phrase marker without considering the complete structure of the sentence.

It also derives Terseness. Well, no. It doesn’t derive the presence of resumptive pronouns. We must not be done.
References


Our system from last time.
The phrase part is:

(1) Derivation
\[ D(S_i), S_i \] a Numeration, is the series of Stages \( (S_1, S_{i+1}, S_{i+2}, \ldots, S_m) \), where:
- Each \( S_{k+1} \) is derived from \( S_k \), by at least one application of merge.
- \( S_m \) is a singleton set (i.e., a root node)

(2) \( \text{MERGE}(\alpha, (\beta)) = \frac{\gamma}{\alpha} \) (\( \beta \))

where:
- \( \alpha \) and \( \beta \) are root nodes in (i.e., elements of) a Stage.
- If one of the arguments of \( \text{MERGE} \) is a word, then \( \gamma \) must be a phrase.
- If \( \text{MERGE} \) has a sole argument, it must not be a phrase.
- \( \gamma \) is the category of \( \alpha \) or \( \beta \).
- \( \alpha \) and \( \beta \) must be certain categories.

(3) Principle of Full Interpretation
If \( \alpha \) is a member of a Numeration, \( S_i \), then \( [\alpha] \) must be part of the denotation of the output of \( D(S_i) \).

The string part is:

(4) \( L_{pc}(P) = \{ S : \{ \alpha < \beta, \alpha \text{ and } \beta X^0 \text{ s in } P \} \} \), such that \( S \) satisfies Path Contiguity, Totality, Antisymmetry, and the language particular component.
- Let \( p(w) = (X_{P1}, X_{P2}, \ldots, X_{Pn}) \), a path, be the phrases that dominate \( w \), an \( X^0 \), and includes the root phrase, where every \( X_{Pi} \) is dominated by every \( X_{Pj} \).
- \( \Pi(P) \) is a set of paths formed from the words in \( P \).
- \( d(XP) \) is the set of \( w \)s such that \( XP \) is in \( p(w) \).

(5) \( L_{pc}(D(S_i)) \)

In many respects, the parallel merge derivations that \( \text{MERGE} \) allows mimic the Copy theory’s effects. Like the Copy theory, it is possible to “move” a term into a phrase that doesn’t dominate where it originates. Indeed, this is required by the condition on \( \text{MERGE} \) that limits the terms it can bring together to just those things that are an element of a Stage (i.e., \((\alpha a)\), the Extension Condition. It permits only derivations that achieve multidomnance through “parallel merge.” This, in turn, forces the resulting phrase markers to be able to be factored into a series of Contiguity preserving non-multidomnent trees. I think this is a desirable effect, but it does bring consequences that might be unwelcome. It is a more constrained theory than one that would allow multidomnent representations by relaxing the Extension Condition. There are problems for relaxing the Extension Condition to achieve Multidomnance. Sportiche (2017) notes that such a system makes it very difficult to state island conditions. Consider for instance (6).

(6) * Near Paris, John met [someone who lives].

Adopt a copy theory of movement, or multidomnent, story that relaxes the Extension Condition and it is difficult to find what would be responsible for the ungrammaticality of (6). For instance, adopt the copy theory and consider a derivation that builds the PP Near Paris, copies it, and then builds the rest of the sentence. There will be the Numeration in (7) in this derivation.
(7) a. \([PP \text{ near Paris}]\)
    b. \([PP' \text{ near' Paris'}]\)
    c. \([TP \text{ John met someone who lives}]\)

If \textsc{merge} is not constrained by the Extension condition, it can now put both PPs in their proper places and Chain Reduction (or its equivalent) will remove the lower one from the string. We'll get an instance of island violating movement. In the present framework, the term that moves will have to be positioned in its lowest position as soon as it becomes part of the phrase that will be in the higher position. That makes it possible to state islands as we did in the first class.

(8) An Island is a Numeration in which every element must have one root.

I left demonstrating how the string part, in tandem with the phrase part, can capture the fact that when verbs move, it is possible for them to be spoken in both of the positions that \textsc{merge} assigns to them. The phrase part can manufacture representations like (9).

(9)

And the string part maps it onto:

(10) \[
\begin{cases} 
  \text{Foc}^0 < \delta & \delta < \text{dá} < \text{sáká} < \text{li} \\
  \text{Foc}^0 < \text{dá} & \delta < \text{sáká} < \text{li} \\
  \text{Foc}^0 < \text{sáká} & \delta < \text{li} \\
  \text{Foc}^0 < \text{li} 
\end{cases}
\]

We derive, from Nunes' ideas, that this instance of movement allows double pronunciation.

1 See, among many others: Vikner (1995).

Not all instances of Verb Movement lead to double-pronunciation, however. Verbs move to Tense in Icelandic, for instance:

(11) Mary kaupir ekki skó?

    (Icelandic)

Mary buys not shoes

Head Movement

On our theory, this will get a representation like (12).

(12)\[
\begin{array}{c}
\text{TP} \\
\text{DP} \quad \text{TP} \\
\text{Mary} \quad \text{T}^0 \quad \text{VP} \\
\text{T} \quad \text{ekki} \quad \text{VP} \\
\text{ir} \quad \text{V}^0 \quad \text{DP} \\
\quad \text{kaup} \quad \text{skó}
\end{array}
\]

A winning output — one that satisfies Totality and Antisymmetry — is (13)

(13) \[
\begin{cases} 
  \text{Mary} < \text{kaupir} < \text{ekki} < \text{kaup} < \text{skó} \\
  \text{Mary} < \text{ekki} < \text{kaupir} < \text{kaup} < \text{ekki} < \text{skó} \\
  \text{Mary} < \text{kaup} < \text{kaupir} < \text{skó} \\
  \text{Mary} < \text{skó}
\end{cases}
\]

This isn't the outcome we want. Here, we want to force the verb to not be pronounced in its lower position.

Why are the Icelandic and Vata outcomes different? I speculate that it has to do with morpho-phonological requirements. Perhaps Icelandic has no vocabulary item that corresponds to a verbal root, the thing I've glossed as \textsl{kaup}. Indeed, the citation forms of Icelandic verbs are not roots but, as in English, inflected forms. If there is no vocabulary item that can be matched to the V position in (12), then this will explain why nothing is pronounced in this position.

This account predicts that the only cases where there is double pronunciation of a moved term are ones in which the moved term becomes part of a word. It is by being hidden in a word that a term passes Antisymmetry unscathed. It looks like the only scenarios where this happens arise when a morpheme or word is moving.
Phrases do not seem to be able to become parts of words. This predicts that the only time a moved term can be spoken twice is when it is a morpheme or a word. And yet there are certain apparent counterexamples to that claim, and so today I want to come clean and look at them.

There are a variety of languages that fit the Vata pattern: the verb is pronounced twice and the fronted verb has a different morphological form than the lower verb. Modern Hebrew has this pattern too:

\[
\text{(14)} \quad \text{liknot, hi kanta et ha-praxim.}
\]
\[
\text{to-buy, she bought ACC the-flowers.}
\]

'As for buying, she bought the flowers.'

(\text{Landau 2004}, (8b): 37)

As in Vata, this movement can be long-distant:

\[
\text{(15)} \quad \text{lā`azor le-Rina, eyn li safek še-Gil hivtiyax še-hu to-help Rina, there-isn't to-me doubt that-Gil promised that-he will-help}
\]
\[
\text{‘As for helping Rina, I have no doubt that Gil promised he would help.’}
\]

(\text{Landau 2006}, (21a): 42)

but not out of islands:

\[
\text{(16)} \quad * \text{likro et ha-sefer, Gil daxa et ha-te`ana še-hu kvar kara. to-read acc the-book Gil rejected acc the-claim that-he already read}
\]
\[
\text{‘As for reading the book, Gil rejected the claim that he had already read.’}
\]

(\text{Landau 2006}, (24a): 43)

We could tell the same story here as the one we've just seen for Vata. The parse would be something like (17).

\[
\text{(17)}
\]
\[
\begin{array}{c}
\text{TopP} \\
\text{Top}^0 \quad \text{TP†} \\
\text{Top}^0 \quad \text{DP†} \quad \text{TP} \\
\text{inf} \\
\text{T^0} \\
\text{V} \\
\text{T} \\
\text{NP} \\
\text{N} \\
\text{flowers}
\end{array}
\]

Modern Hebrew is like Icelandic: Hebrew verbs move to T^0 to get inflected. Indeed, Modern Hebrew is exactly like Icelandic. When its verb moves to T^0, there is only one pronunciation and it is in the position T^0 occupies. I'll give the same treatment of this. V doesn't dominate a word in Hebrew, but Top^0 and T^0 do. Top^0 is pronounced as the infinitival form of the verb and T^0 is pronounced as the inflected verb (past, say). The largest Π(17) is (18).

\[
\text{(18)}
\]

a. \( p(\text{Top}^0) = \{\text{TopP}\} \)

b. \( p(\text{D}^0) = \{\text{DP}†, \text{TP}†, \text{TopP}\} \)

c. \( p(\text{T}^0) = \{\text{TP, TP}†, \text{TopP}\} \)

d. \( p(\text{D}^0) = \{\text{DP, VP, TP, TP}†, \text{TopP}\} \)

e. \( p(\text{N}^0) = \{\text{NP, DP, VP, TP, TP}†, \text{TopP}\} \)

Note that there is no \( p(\text{V}) \), because \( V \) isn't a word. Paths are defined only for \( X^0 \), things that get parsed as a word. From (18) we can calculate the \( d \) mappings:

\[
\text{(19)}
\]

a. \( d(\text{TP}†) = \{\text{D}^0, \text{T}^0, \text{D}^0, \text{N}^0\} \)

b. \( d(\text{TP}) = \{\text{T}^0, \text{D}^0, \text{N}^0\} \)

c. \( d(\text{VP}) = \{\text{D}^0, \text{N}^0\} \)

d. \( d(\text{DP}) = \{\text{D}^0, \text{N}^0\} \)

e. \( d(\text{NP}) = \{\text{N}^0\} \)
Notice, again, that V is not included. When the language particular component of Hebrew is considered, this set of paths has only one linearization that satisfies Path Contiguity (I'll give it in a more compact string form):

\[ (\text{two.oldstyle}/\text{zero.oldstyle}) \text{ Top} \]

I have credited Top with giving the moved verb its infinitival form. Landau proposes a different idea but one that nonetheless involves putting the verb inside a larger word. He notes that a topicalized term gets a high tone, which we can think of as being the morpheme associated with Top. It's providing this high tone with a host that the verb's movement does.

Modern Hebrew also allows the verb to be spelled out in two places when not the verb, but the whole VP topicalizes:

\[ (\text{two.oldstyle}/\text{one.oldstyle}) \text{ Top} \]

This is the kind of example that counter-exemplifies the generalization that only moved words can get spoken twice.

Landau argues that what we're seeing in these examples involves the verb moving out of the topicalized VP. We can apply the Nunes-like procedure here too, since in these cases as well, the verb in the fronted VP and the verb in the lower position bear different morphology. Our parse looks like (22).

\[ (\text{two.oldstyle}/\text{three.oldstyle}) \text{ Top} \]

Landau's account for this is that the high tone can only fall on a verb and it can only fall on the first tone bearing word in the phrase that has topicalized. But other phrases can topicalize in Hebrew and when they do presumably the high tone's requirement to be pronounced is satisfied without falling on a verb. For instance, (24) is grammatical.

\[ (\text{three.oldstyle}/\text{nine.oldstyle}) \text{ Top} \]

\[ (\text{two.oldstyle}/\text{four.oldstyle}) \text{ Top} \]
Some kinds of constituents cannot topicalize in Hebrew. Finite CPs can’t, for instance.

(25) *še-hu heŏliv et Rina, Gil hictaĕr.
that-he insulted acc Rina Gil regretted
‘As for that he had insulted Rina, Gil regretted.’

compare:

(26) le’hictaĕr še-hu heŏliv et Rina, Gil hictaĕr.
to-regret that-he insulted acc Rina Gil regretted
‘As for regretting that he had insulted Rina, Gil regretted.’

(Landau 2006, (42b): 54)

So we need some way of controlling what things can satisfy Top’s needs. One way of getting this is to restrict the class of heads that can move in Hebrew, and require that the item that bears Top’s high tone move into Top. Maybe V and N can move in Hebrew (there is independent evidence for that), but complementizers of finite clauses cannot. If we were to pursue Landau’s approach, we would postulate a parallel set of constraints that don’t use head movement, and just stipulate that the head at the left edge of the constituent must bear Top’s high tone.

Because the VP has two mothers in (22), there are two paths available for the words it contains: D⁰ and N⁰. If we choose the paths that Hebrew likes for this structure – one that puts the VP together with Top⁰ – then the Π(22) which satisfies Totality is (27).

(27) a. p(N⁰) = {NP, DP, VP, TopP, TopP†}
    b. p(D⁰) = {DP, VP, TopP, TopP†}
    c. p(T⁰) = {TP, TP†, TopP†}
    d. p(D†0) = {TP†, TopP†}
    e. p(Top⁰) = {TopP, TopP†}

This gives us the following values for d.

(28) a. d(NP) = {N⁰}
    b. d(DP) = {D⁰, N⁰}
    c. d(VP) = {D⁰, N⁰}
    d. d(TP) = {T⁰}
    e. d(TP†) = {D†0, T⁰}
    f. d(TopP) = {Top⁰, D⁰, N⁰}
    g. d(TopP†) = {D†0, T⁰, D⁰, N⁰, Top⁰}

The only precedence relations that honors the head-initial/specifier-initial nature of Hebrew, and also satisfies Path Contiguity is the correct:

(29) Top⁰ D⁰ N⁰ D†0 T⁰
≡ liknot et-ha-praxim hi kanta

To explain the obligatory pronunciation of the verb in both Top and T position, Landau invokes:

(30) X is associated with phonetic content iff:
    a. X has phonetic content, or
    b. X is in a position specified with some phonological requirement.

(31) P-Recoverability
In a chain <X₁...Xₙ>, where some Xᵢ is associated with phonetic content, Xᵢ must be pronounced.

(Landau 2006, (48-9):56)

He is working with the copy+delete model of movement, and so he conceives of P-Recoverability as a condition limiting deletion. I’ve built essentially the same idea into our linearization process, after the idea that the word bearing the tone that Top has must head move into Top is added. To prevent the lowest verb in these representations from being pronounced, Landau suggests:

(32) Economy of Pronunciation
Delete all chain copies at PF up to P-recoverability.

(Landau 2006, (51): 57)

In my system, the work of this condition is done by treating the unpronounced heads as something other than words. As a consequence, they won’t get into the linearization. I don’t see anything inconsistent with my (or other) multidominant systems in (32). It might be that we could use it too. I’m not sure how to distinguish (32) from the system I’ve invoked here.

One difference between Landau’s system and the one I’ve built from Nunes is that Landau’s system would allow double pronunciation of a moved verb under a variety of different circumstances. Anytime the phonology requires it, it should be obligatory. In my system, it’s only when moved verbs are in different words that a double pronunciation becomes possible. To my knowledge, this makes the accurate prediction that when a moved verb is pronounced twice, it is not the exact same word. As in the Vata and Hebrew examples we’ve seen, the verbs are visibly parts of different words – that is, they have different forms. On the view I’ve taken, the only time a moved verb is pronounced twice and has the same morphological form in both of its pronunciations will be when the two words these verbs are are homophonous.
I think a thesis that is interesting is that every instance of VP movement with a copy of the head V pronounced in the lower position has the syntax of Hebrew. That is, it’s really a special case of simple verb movement. If that can be maintained, then the only cases where a moved term is truly spelled out in both of the places it occupies are ones in which a head has moved. Nunes’s type of account for this exception to Terseness explains that constraint. But I don’t see clearly how a system built upon Economy of Pronunciation would.

There are some cases of VP movement in which the verb seems to be spoken only in the lower position. The conditions that license these cases aren’t well understood. English might have one of these constructions.

(33) Into the room quietly crept the mouse.

Rochemont and Culicover (1990) argues that this involves movement of a VP from which the verb has moved. One possible parse is (34). (I’ve ignored the existence of vP.)

Yiddish can do just what Modern Hebrew does:

(35) a. Essen est Maks fish
to-eat eats Max fish
‘As for eating, Max eats fish.’
b. Essen fish est Maks.
to-eat fish eats Max
‘As for eating fish, Max eats them.’

Notice that Yiddish is verb initial and Verb Second. So just like Modern Hebrew, the verb has moved out of the VP that is fronted, and the verb in the fronted VP is an infinitive – a different form than in the lower position.

But Yiddish can also front verbs, or their projections, and pronounce the verbs in the fronted position and the lower one with the same morphology:

(36) a. Gegessen hot Maks gegessen fish.
eaten has Max eaten fish
‘As for having eaten, Max has eaten fish.’
b. Gegessen fish hot Maks gegessen.
eaten fish has Max eaten
‘As for having eaten fish, Max has eaten them.’

This is exactly what Nunes’s system leads us to expect won’t happen, modulo homophony. And note too that we’d have to move the participle out of the VP in the lower position before doing topicalization, if we are to preserve exactly the same syntax that Landau suggests for Hebrew.

Yiddish clefted VPs are island sensitive, so we do want to invoke movement here.

(37) a. Veysn hostu mir gezogt az er veyst a sakh.
to-know have-you me told that he knows a lot
‘As for knowing, you told me that he knows a lot.’
b. *Veysn hob ikh gezem dem yidn vos veyst a sakh.
to-know have I seen the man who knows a lot
‘As for knowing, I saw the man who knows a lot.’

(Davis and Prince 1986)

Interestingly, Cable points out that there is a difference in the meanings of the cases where the lower participle is spoken and where it isn’t. In the case where it is
spoken, it must be focused and is appropriate only in a context where it represents new information. This is what the minimal contrast in (38) is about.

(38)  a. Gevust hob ikh es.
      known have I it
'As for having known, I have known it.'

b. Gevust hob ikh es GEVUST
known have I it known
'As for having known, I have KNOWN it.'

(Cable 2004, (9): 6)

But, as we’ve seen in other contexts, this is still a strategy that is reserved to verbs (and their predicates). A nominal cannot be topicalized and its lower copy focus marked:

(39)  * Maksn hob ikh gezon MAKSN.
      Max have I seen MAX
'As for Max, I have seen MAX.'

(Cable 2004, (10): 6)

If we could figure out why nominals can’t be repeated, we could use the focus fact to explain why verbs can. Perhaps the lower position has a focus mark on it that the higher position doesn’t, and this might serve the same role that morphology does in Vata and Modern Hebrew. (For a similarly prosodic story about why a verb is forced to be pronounced in two places, see Kandybowicz 2015.)

In addition to predicate clefts like these – where the topicalized verbal projection contains a root that is repeated in the lower position – there is garden variety remnant movement in Yiddish. In remnant movement, there is no lower copy of the root, and the fronted phrase is missing a constituent that is otherwise obligatory in a VP. So, for instance, a locative is obligatory in Yiddish VPs headed by “put.”

(40)  * Er leygt dos bukh
      He put the book
'He put the book.'

(41)  Er leygt dos buch afn tisch
      he put the book on-the table
'He put the book on the table.'

A fronted VP can fail to have the locative, if that locative is left below.

(42)  Gelygt dos bukh hot er afn tisch.
      put the book has he on-the table
'He put the book on the table.'

But predicate clefts, where the verb is repeated, don’t allow this.

(43)  * Leygn dos bukh leygt er afn tisch.
      to-put the book put he on-the table
'He put the book on the table.'

compare:

(44)  Leygn dos bukh afn tisch leygt er
      to-put the book on-the table put he
'He put the book on the table.'

(Cable 2004, (12): 7)

In these constructions, the VP that has topicalized has to be complete and well-formed. This, incidentally, is true of VP topicalization in Modern Hebrew as well. Interestingly, the VP that has topicalized doesn’t have to be identical to the VP that is pronounced lower. And that suggests that we are not looking at phrases in two positions in these examples.

(45)  a. Essen ish est Maks hekht
      to-eat fish eats Max pike
'As for eating fish, Max eats pike.'

b. Essen frukht est Maks bananes.
      to-eat fruit eats Max bananas
'As for eating fruit, Max eats bananas.'

(Cable 2004, (15): 9)

The same thing happens in Brazilian Portuguese.

(46)  a. Comer peixe, eu normalmente como samão.
      to-eat fish, I usually eat salmon
'As for eating fish, I usually eat salmon.'

b. Comer peixe, a Mary acha que eu como samão.
      to-eat fish, Mary thinks that I eat salmon
'As for eating fish, Mary thinks I eat salmon.'

(Cable 2004, (21a,b): 11)

And for Brazilian Portuguese (which otherwise resembles Yiddish with respect to the cleft construction), Cable demonstrates that island effects hold even when
the material in the higher and lower VPs is not the same. (His Yiddish informant didn't allow long-distance clefts at all, so he couldn't test for islands. The literature reports that when Yiddish speakers do, islands are obeyed.)

\[(47)\]
\[
\begin{align*}
a. \quad & \text{If a verb topicalizes, it can be pronounced in the lower position only if the lower and higher pronunciations are different morphologically.} \\
\text{b.} \quad & \text{If a VP topicalizes, it cannot be spelled out in the lower and higher positions. (Apparent exception: Yoruba).} \\
\text{c.} \quad & \text{If a VP topicalizes, the verb can be spelled out in the higher and lower position only if the verb has moved independently out of the VP.}
\end{align*}
\]

If these conclusions are correct, they are captured by the method of achieving multidominance proposed here, in concert with the linearization scheme I described yesterday.
References


Sportiche, Dominique. 2017. Somber prospects for late merger. UCLA.

In our examination of verb movement, we haven’t had to tackle the semantics very seriously. It’s conceivable that all of the examples we’ve looked at are ones in which the moved verb, or VP, is interpreted semantically in only one of the positions it occupies or, perhaps, in both of those positions. But this can’t be what’s happening in wh-movement, as we saw on the first day. Examples like (1) teach us that the moved wh-phrase must be able to be interpreted in both positions, but not in the same way!

(1) Which papers of hers that Mark brought will no woman talk to him about?

The pronoun can be interpreted only in its lower position – bringing it within the scope of no woman – while at the same time Mark can be interpreted only in its higher position where it isn’t in the scope of him. We start to look at what’s necessary to get this example today, but it will take all of the rest of the classes before we will have enough ingredients to finish this project.

We start by looking at the semantics of constituent questions. I’m going to sketch an approach to this problem that is in Engdahl (1980), who also used a multidominant model of movement. She did not have (1) in her sights, but the simpler (2).

(2) Which papers of hers will no woman talk about?

Fox (2002) translated Engdahl’s ideas into the copy theory of movement and broadened it to account for cases like (1), and others. I’ll get us there, however, by starting with Engdahl.

Engdahl favored a Remerge model of multidominance. One, that in our terms, would be achievable by loosening the Extension Condition. Simple wh-questions have, on her view, the shape in (3).

(3) (I know) which child she kissed.

I’ve designed our system for parallel merge derivations, and banned representations like (3). But let’s enter Engdahl’s world for a while and look at her ideas for trees of this shape.

A standard, simple, view of the meaning of questions is that they denote a set of propositions, each proposition offering a kind of answer in those cases where the question is answer-seeking. This is the view introduced by Hamblin (1973) and modified by Karttunen (1977). One way of representing a set is with the λ-operator, which can be used to represent a function that characterizes the set.

(4) λxP(x) = that function which, when applied to a, gives P(a).

Natural language functions are restricted; they are defined in such a way that they only apply to certain kinds of terms. One way in which they are restricted is by semantic type. Natural language functions apply only to arguments of the semantic type that they select.

Using the characteristic function language, we can give (3) a denotation like (5).

(5) λp ∃x x is a child & p = she kissed x
The “p” represents a proposition: this function is restricted to applying to propositions. (6) characterizes the set of propositions, then, for which there is a child and that proposition is that she kissed that child. Our challenge, then, is to get this kind of meaning out of (3).

The central problem a remerge definition of movement poses is that it baldly predicts that the single meaning that is associated with the moved item should be found in both of its positions. It is that feature of multidominant representations that does the relevant work of deriving Terseness. But that isn’t what we want for the meaning of the questions formed by wh-movement. Instead, we must associate the moved wh-phrase with both a binder meaning and a variable meaning. This is the paradox of movement. If Nunes’ approach to Terseness is correct, then we want a theory of movement that requires the term that has moved to be treated by the linearization algorithm as though it is one thing in two positions. Multidominant representations give a trivial expression to this idea. But the semantics of movement seems to require something quite different. It requires a moved term to create in the two positions involved in the movement different semantic objects. One a binder, the other a variable. This is quite straightforwardly not what a multidominant representation of movement predicts. Indeed, and this will be our first problem, it seems to prevent the moved term from having two different denotations in the positions it occupies.

Engdahl (1980, 1986) proposes that the moved wh-phrase simply has two meanings, and they are introduced selectively at their different positions. There is nothing about movement itself that creates these two meanings. Rather, the terms that are movable with the binder/variable semantics are just those terms that can be assigned two meanings. The meaning introduced in the lower position must be a variable and the meaning it has in the higher position is something that binds that variable. We should do this in such a way that we can model “reconstruction,” the process that allows variables in a moved phrase to act as if they are in their unmoved position, as for example in (2) or (6).

(6) Which picture of himself should no one put on his website?

Note that it is the position from which movement has occurred that matters.

(7) a. Which picture of himself₁ does this indicate that no one₁ should bring?

b. * Which picture of himself₁ does the thing no one₁ heard about indicate I should bring?

c. * Which picture of himself₁ indicates that no one₁ should bring it?

So the fact that himself is bound by no one in (6) should occur because the position which picture of himself is “moved” from is c-commanded by no one. It is the ability of this wh-phrase to be interpreted in its lower position that allows himself to be bound by no one.

These two considerations together might lead us to a picture like that shown in (8). (I take the possibly perverse view in (8), that put selects a small clause.)

If we adopt a Karttunen style analysis of questions, then for (8) what we want is to get an interpretation along the lines of (9).

(9) \( \lambda \phi \exists x \text{ picture}(x) \land \phi = \text{no one}_2 \) should put picture-of-himself₂(x) on his website.

How can we do that, though? Notice that himself is not interpreted in the higher position because that would not put it in the scope of its binder. How can we convert the representation in (8) into something that approximates (9)?

But actually, (9) isn’t quite right either. (9) characterizes the question as seeking the identity of a single picture with the expansive property of being of a bunch of guys, none of whom should put it on their website. That’s not what we want. We want something that allows the pictures to vary with the variable it contains. The anaphoric connection between a moved phrase and its trace must be capable of carrying this duty; this is a fact about questions that don’t have overt bound vari-
ables in them. Even (10) needs a semantics in which the pictures vary with the quantifier.

(10) Which picture should no one forget?

We need to build into our semantics for a question a way for the quantification that the Wh-phrase invokes to be able to depend on quantifiers it has crossed over. Whatever way of doing that we can manage should be related to the question we have about how himself gets bound. The pictures in (7), after all, vary with the values given to the anaphor.

Elisabet Engdahl gave us a way of doing that.

Skolem Functions

Engdahl (1980) makes a case for treating questions as being able to involve quantification over functions, and then she uses this to solve the problem of reconstruction. The need for this can be seen by considering, for instance, the contrast in (11).

(11) a. Vilka av sina dikter ville varje medlem i Lyrikklobben att which of his-own poems wants every member of poetry club that en författare skulle komma och läsa. an author should come and read
   (Which of his poems did every member of the poetry club want an author to come and read?)
   b. Varje medlem i Lyrikklobben ville att en författare skulle every member of poetry club wants that an author should komma och läsa en av sina dikter. come and read one of his-own poems.
   (Every member of the poetry club wants an author to come and read one of his poems.)


(NB: Engdahl 1980 is about Swedish.) Both sentences lead to an interpretation in which each poetry club member wants a different author to come and read any one of that author's poems. But in (11a), there is also a presupposition that there is one way of characterizing those sets of poems. That "way" could be characterized in an answer, like that in (11a), for instance.

(12) his first

Engdahl's idea, then, is that questions can be asking for ways of characterizing sets of individuals. In some cases it seems particularly clear that a question is not seeking the identity of an individual, or a set of individuals, but is instead seeking a formula that identifies such individuals. One of these is (13).

(13) Which number does every number immediately preceede? (Engdahl 1986, (72):181)

I don't believe there is a way of naming an individual that provides a (true) answer to this question. By contrast, there are ways of naming a formula that constitutes a true answer. One of these is: "its successor."

Engdahl gives a couple of ways of characterizing this feature of questions, and they solve our problem (and bear generally upon how to do reconstruction). Let's begin by considering the proposal in Engdahl (1986). She builds her idea on the interesting speculation that interrogative phrases invoke as one of their meanings something parallel to what is found in donkey pronouns, such as the it in (14).

(14) Every man who owns a donkey kisses it.

The puzzle about donkey pronouns is that they vary, like variables do, but in a way that is indirectly related to the quantifiers involved. In (14), for instance, one might attempt to paraphrase the meaning it has with something like (15).

(15) Every man who owns a donkey kisses the one he owns.

Cooper (1979) proposes to capture this meaning by allowing the denotations of pronouns to involve an open relation variable, which picks out individuals based on some relation they bear to other individuals. In this example, that relation is given overtly by the relative clause in (15). Engdahl builds her idea about questions on this model of "e-type" (or "d-type") pronouns as donkey pronouns are sometimes called.

So, let the pronoun in (14) have the denotation in (16).

(16) \[ [\text{it}] = f_{\text{ec},\phi'}(x) \]

\( f \) is a function that returns a value of type \( e \) based on the value given to \( x \). We can approximate the appropriate meaning for (15) with (17).

(17) \( \forall u [\text{man_who_owns_donkey}(u) \rightarrow u \text{ kisses } f(u)] \)

\( f = \text{the donkey owned by} \)

What is needed to complete this account of donkey-anaphora is a story about how \( f \) manages to get its value. In this example, the relative clause, somehow, makes that relation salient and \( f \) picks it up.
Engdahl suggests that the readings for the questions we see in (13) and (12) are ones in which the wh-phrase invokes in its lower position a hidden relational variable of the same type that is seen in the donkey cases. Instead of letting a question invoke an existential quantification over individuals, she suggests that we let them introduce an existential quantification over these relations. Questions that do this are known as "functional" questions.

The denotation Engdahl (1986) assigns the wh-phrase in its higher position introduces the restrictor (NP) part in a way that allows the reconstruction effects to be manufactured. She gives us denotations like (18).

(18) Which number does every number immediately precede?
\[ \lambda p \exists f [\forall x \text{number}(f(x)) \land p = \forall y \text{number}(y) \rightarrow y \text{ immediately precedes } (f(y))] \]

The “\( \forall x \text{number}(f(x)) \)” part tells us that the individual which \( f \) picks out must be a number. We want this part to come from the NP portion of the DP which \( \text{number} \) in its higher position. And the object position of \( \text{precede} \) is occupied by \( f(y) \), so we want that to be what the \( \text{which}- \)phrase denotes in its lower position. A sketch of how (18) might be derived from the syntactic representation is (19).

(19) \[ \lambda p \exists f \forall x \text{number}(f(x)) \land p = \forall y \text{number}(y) \rightarrow y \text{ precedes } f(y) \]

\[ \lambda W \lambda p \exists f \forall x \text{number}(f(x)) \land W(f)(p) \]
\[ \lambda Q \lambda W \lambda p \exists f \forall x Q(f(x)) \land W(f)(p) \]
\[ \lambda y \text{number}(y) \]
\[ \lambda Q \lambda W \lambda p \exists f \forall x Q(f(x)) \land W(f)(p) \]
\[ \lambda y \text{number}(y) \]
\[ \lambda Q \lambda W \lambda p \exists f \forall x Q(f(x)) \land W(f)(p) \]
\[ \lambda y \text{number}(y) \]
\[ \lambda Q \lambda W \lambda p \exists f \forall x Q(f(x)) \land W(f)(p) \]
\[ \lambda y \text{number}(y) \]
\[ \lambda Q \lambda W \lambda p \exists f \forall x Q(f(x)) \land W(f)(p) \]
\[ \lambda y \text{number}(y) \]
\[ \lambda Q \lambda W \lambda p \exists f \forall x Q(f(x)) \land W(f)(p) \]
\[ \lambda y \text{number}(y) \]
\[ \lambda Q \lambda W \lambda p \exists f \forall x Q(f(x)) \land W(f)(p) \]
\[ \lambda y \text{number}(y) \]

I've interpreted the index on the trace in a very Heim and Kratzer (1998) way: it gets interpreted as a \( \lambda \) in the higher position, and as a variable in the lower position. The key thing here is that the relation “\( I \)” takes as its argument is a variable of type \( e \), and this variable can be bound by quantifiers that scope over it. For every value given to the quantifier, a possibly different individual will be chosen by “\( I \)”. This is the effect we are trying to achieve: numbers that the variable in object position ranges over varies as a function of the values given to the subject.

Engdahl (1986) makes the natural decision to semantically compose \( \text{which} \) with the NP it combines with syntactically. If \( \text{which} \) is introducing an existential quantification over functions, \( f \), then it isn’t a stretch to imagine that the NP is restricting what these \( f \)s range over. That is what she does, but because the \( f \) comes with a free variable, we need to close that variable in order to restrict what \( f \) picks out. Engdahl’s rule of closure does that with a universal quantifier.

Closure equips us with what we need to solve the problem that cases like (20) brought out.

(20) …which picture of himself no one put on his website.

In (20), the NP that combines with \( \text{which} \) contains a reflexive, that, we can assume, is necessarily interpreted as a variable. Thus it could get caught by closure. Engdahl suggests, then, that the \( \text{himself} \) in this sentence only \( \text{appears to be bound directly by no one} \). Instead, it is interpreted as one of the individuals that \( f \) involves by virtue of being bound by the universal quantifier that closes the NP part of the wh-phrase. She would give to (20) an interpretation like that in (21).

(21) \[ \lambda p \exists f [\forall x \text{picture_of_x}(f(x))] \land p = \exists y y \text{ put } f(y) \text{ on his website} \]
spoke in and let it influence the interpretation of the meaning the phrase has in the lower position.

That's the basic idea. In the examples we've looked at so far, \( f \) is a one-place relation. It takes one variable over individuals and delivers another individual that is in the specified relation. Engdahl notes that the relation \( f \) can involve more than just one argument, however. She points to examples like (22) for the need to let \( f \) be a two-place relation.

\[ \lambda p \exists f^2 \forall x \forall y [x's \ letters \ to \ y \ (f(x, y))] \land p = \forall z [\text{woman}(z) \to \text{return} \ (f(m, z)) \ to \ im[z's \ lover](m)] \]

Engdahl suggests the interpretation in (23).

(22)  

| a. Which of his letters to her does every woman return to her lover?  
| b. The ones where he asks her for money.  

(Engdahl 1986, (77): 186)

I've adopted here the convention of showing the valency of \( f \) with a superscript; \( f^2 \) means that \( f \) takes two arguments. Engdahl lets \( f \) involve less than two arguments as well. She suggests that there is a \( f \) that is a constant function, picking out the same individual under all assignment functions. This is the meaning, we might imagine, behind a vanilla, individual-based question, like (24).

\[ \lambda p \exists f^1 \forall x [\text{boy}(f^\#) \land p = \text{win}(f^\#)] \]

Let "\( f^\# \)" represent this constant function, and (24) can be given the interpretation in (25).

(25)  

(Engdahl 1986, (62'): 177)

It's also possible to get a constant out of an \( f \) that is interpreted in a dependent way. For instance, it is possible to model the exchange in (26) with (27).

\[ \lambda p \exists f^1 [\forall x \text{woman}(f(x)) \land p = \forall y \text{Frenchman}(y) \to y \text{admires} \ (f(y))] \]

This, presumably, would be the representation required for (26a) in order to furnish a suitable question for (28).

\[ (\lambda f) \exists f^1 \forall x \forall y [x's \ letters \ to \ y \ (f(x, y))] \land p = \forall z [\text{woman}(z) \to \text{return} \ (f(m, z)) \ to \ im[z's \ lover](m)] \]

(28)  

| a. Which of his letters to her does every woman return to her lover?  
| b. The ones where he asks her for money.  

(Engdahl 1986, (62'): 177)

Let "\( f^\# \)" represent this constant function, and (24) can be given the interpretation in (25).

(25)  

(Engdahl 1986, (62'): 177)

It's also possible to get a constant out of an \( f \) that is interpreted in a dependent way. For instance, it is possible to model the exchange in (26) with (27).

(26)  

| a. Which woman does every Frenchman admire?  
| b. Brigitte Bardot.  

(27)  

\[ \lambda p \exists f^1 [\forall x \text{woman}(f(x)) \land p = \forall y \text{Frenchman}(y) \to y \text{admires} \ (f(y))] \]

This, presumably, would be the representation required for (26a) in order to furnish a suitable question for (28).

\[ (\lambda f) \exists f^1 \forall x \forall y [x's \ letters \ to \ y \ (f(x, y))] \land p = \forall z [\text{woman}(z) \to \text{return} \ (f(m, z)) \ to \ im[z's \ lover](m)] \]

(28)  

| a. Which of his letters to her does every woman return to her lover?  
| b. The ones where he asks her for money.  

(Engdahl 1986, (62'): 177)

Let "\( f^\# \)" represent this constant function, and (24) can be given the interpretation in (25).

(25)  

(Engdahl 1986, (62'): 177)

It's also possible to get a constant out of an \( f \) that is interpreted in a dependent way. For instance, it is possible to model the exchange in (26) with (27).

(26)  

| a. Which woman does every Frenchman admire?  
| b. Brigitte Bardot.  

(27)  

\[ \lambda p \exists f^1 [\forall x \text{woman}(f(x)) \land p = \forall y \text{Frenchman}(y) \to y \text{admires} \ (f(y))] \]

This, presumably, would be the representation required for (26a) in order to furnish a suitable question for (28).

(28)  

| a. Which of his letters to her does every woman return to her lover?  
| b. The ones where he asks her for money.  

(Engdahl 1986, (77): 186)

Engdahl suggests the interpretation in (23).

(23)  

Engdahl notes that the relation \( f \) can involve more than just one argument, however. She points to examples like (22) for the need to let \( f \) be a two-place relation.

\[ (\lambda p) \exists f^2 \forall x \forall y [x's \ letters \ to \ y \ (f(x, y))] \land p = \forall z [\text{woman}(z) \to \text{return} \ (f(m, z)) \ to \ im[z's \ lover](m)] \]

(28)  

| a. Which of his letters to her does every woman return to her lover?  
| b. The ones where he asks her for money.  

(Engdahl 1986, (77): 186)

Engdahl notes that the relation \( f \) can involve more than just one argument, however. She points to examples like (22) for the need to let \( f \) be a two-place relation.

\[ (\lambda p) \exists f^2 \forall x \forall y [x's \ letters \ to \ y \ (f(x, y))] \land p = \forall z [\text{woman}(z) \to \text{return} \ (f(m, z)) \ to \ im[z's \ lover](m)] \]

(28)  

| a. Which of his letters to her does every woman return to her lover?  
| b. The ones where he asks her for money.  

(Engdahl 1986, (77): 186)

Engdahl notes that the relation \( f \) can involve more than just one argument, however. She points to examples like (22) for the need to let \( f \) be a two-place relation.

\[ (\lambda p) \exists f^2 \forall x \forall y [x's \ letters \ to \ y \ (f(x, y))] \land p = \forall z [\text{woman}(z) \to \text{return} \ (f(m, z)) \ to \ im[z's \ lover](m)] \]

(28)  

| a. Which of his letters to her does every woman return to her lover?  
| b. The ones where he asks her for money.  

(Engdahl 1986, (77): 186)
(32) Which picture should no woman’s father bring?
answer: “the one on her facebook page”

So:
(33) A personal pronoun and $f(x)$ can be bound by something in its QR’d position, but a reflexive cannot be.

Now consider (34).

(34) a. Which picture of her should no woman’s father bring?
b. Which picture of herself should no woman’s father bring?

On Engdahl’s (1986) account, there should be no difference in these examples. The reflexive or personal pronouns is not bound by no woman, on her account, but instead by the universal quantifier that closes the free variable in the restrictor. (See (35).)

(35) $\lambda p \exists f \forall x \text{picture_of}_x(f(x)) \wedge p = \text{no woman}_1$’s father $f(x_1)$

But I believe there is a contrast in (34a): (34b) is worse than (34a). This will follow if the variable expressed by her or herself is bound directly by no woman, rather than by an intermediary binder.

So, let’s revert to a model in which the NP part of the moved wh-phrase is interpreted in its lower position. If we don’t use Engdahl’s closure operation, then when the DP contains a variable, like her or herself, we must moreover prevent the DP from being semantically interpreted in its higher position in these examples. So, here’s a way of doing that which is loosely modeled on Heim (2013).

(36) Assume, with Engdahl, that a wh-phrase has two denotations, one that it uses in its lower position and a second that it uses in its higher position. These denotations are:

a. higher position = $[\text{which NP}] = \lambda Q \lambda p \exists f Q(f)(p)$
b. lower position = $[\text{which NP}] = f(x)$, defined when $[\text{NP}](f(x)) = 1$

On this view, the quantification over functions is restricted by virtue of a presupposition introduced by the denotation the phrase has in its lower position. This presupposition has been expressed by making the denotation associated with the lower phrase partial. It has a value only when that value makes $[\text{NP}]$ true. Thus, this entire sentence has a value only when the values $f(x)$ takes make $[\text{NP}]$ true, and this garners a result that is very close (though not identical) to restricting the quantification over those functions. If we now revert back to our syntax of multidominance, and away from Engdahl’s, we can illustrate how these denotations work with (37).

(37) Which picture of herself did no woman bring?

$\lambda p \exists f p = \text{no woman}_1$; bring $f(1)$
[VP] and every node dominating it up to λ2 is defined only if [pictures of x] holds of 2(1).

We don’t know what X is yet. Assume it contributes no meaning. As a consequence, XP has just the denotation that the wh-phrase has, and, since this is its higher position, this is (36a). Note that I have to assume that Wh Movement also places the λ2 into the structure. And, finally, we have to assume that when XP combines with its sister CP, the presupposition projects up in a way that restricts the domain of f.

References


We left time looking at how Engdahl (1980) treated reconstruction effects. She had a remerge theory of movement, so her trees looked like (1).

1. Which picture of herself should no woman bring?

To get the fact that the same DP *which picture of hers* is both a variable and a binder, she decides to simply give that DP those two meanings and let them combine with the right material. We did that in a way that follows Heim (2013):

2. \[\llbracket \text{which picture of hers} \rrbracket = \]
   \[a. \quad \text{higher position} = \lambda Q \quad \lambda p \quad \exists f \quad Q(f)(p)\]
   \[b. \quad \text{lower position} = f(x), \text{defined when} \quad \llbracket \text{pictures of hers} \rrbracket (f(x))=1\]

This model has a couple flaws. One is that it requires that we embrace a theory that allows certain expressions to come with two denotations, one for each of the positions they occupy. This is an enrichment of the grammar’s power that we should avoid if we can. The other is that it doesn’t give us a handle on resumptive pronouns. Why are violations of Terseness in the cases where a DP moves ones in which the lower position is pronounced as a pronoun? Interestingly, Engdahl’s semantics provides a reason – the lower position has the meaning of an e-type pronoun – but her syntax doesn’t. So, I’m going to try to refine Engdahl’s system into a syntax that is aimed at explaining this.

First, we need to get a handle on the internal syntax of pronouns. Our model for the lower position of wh- phrases is the e-type pronoun. We did that in a way that follows Heim (2013):

3. Every man who owns a donkey kisses it.

Built into this pronoun is a function, that like the ones bound in questions picks out an entity that satisfies the NP, here *donkey*, depending on the values given to *every man*. Interestingly, a definite description can also get this meaning:

4. Every man who owns a donkey kisses the donkey.

Pronouns share certain other characteristics with definite descriptions. For instance, their context of use is, many times, similar. The sentences in (5) are odd in the context of this class for the same reason.

5. a. The woman is bored.
    b. She is bored.

I’m going to assume that there are bored women in my class by the time we get to this point in the handout, so these sentences should both be true. But they’re not because the subjects do not refer properly, and this is because there is more than one woman in my class (unless all but one of them have left by now). This is modeled by making both *the woman* and *her* trigger the presupposition that there is one unique female in the situation being described by the sentence. This presupposition comes with *the*, so we’ll build it into the denotation of *the*.

6. \[\llbracket \text{the} \rrbracket = \lambda P.t \times P(x), \text{defined when there is exactly one} \]
   \[x \text{ in} \quad s \text{ such that} \quad P(x) = 1.\]

Understand the operator \(t\) to make the predicate it combines with refer in the same way that a name does. The definedness condition is restricted to the situation being described, which is represented in (6) by the free variable \(s\). This gives us, then, something like (7) for (5a).
(7) \( \lambda x \ x \text{ is bored} \)

\[ \text{DP} \]

\[ \text{TP} \]

(10) \( \lambda x \ \text{is bored} \)

\[ \text{DP} \]

\[ \text{TP} \]

\( [\text{DP}] \) and \( [\text{IP}] \) are defined just in case there is a unique \( x \) in the relevant situation which is a woman.

To capture the fact that pronouns are similar to definite descriptions in these ways, we can give them a similar composition. Imagine, for instance, that a pronoun is just the way English speaks a definite description whose NP part is the relevant person, gender and number features. I’ll assume that “third person” is the absence of first or second person. So the pronoun \( \text{her} \) on this view might look like (8). (See Postal 1969, and Elbourne 2005 after which this is modeled.)

(8) \( D^0 \)

\[ \text{NP} \]

(11) \[ n = \lambda x. x = n \]

For an index that has a function in it, we can adopt the denotation in (12).

(12) \[ f = \lambda x. x = f(y) \]

where “\( y \)” is the argument of \( f \).

A definite description with an index in it, then, will have a denotation like that in (13).

(13) \( \lambda P. \lambda x P(x) \)

\( \lambda x. x = f(y) \)

\( \lambda x. x \) is donkey

\[ \text{DP} \]

\[ \text{TP} \]

\( [\text{DP}] \) and \( [\text{TP}] \) are defined just in case there is a unique \( x \) in the relevant situation which is female and singular.

The donkey pronoun cases show us that these expressions can have \( f \) in them, and our model of questions requires that this \( f \) be bindable. I’ll treat this \( f \) as an index – the kind that makes an expression a variable. And I’ll adopt an idea in Danny Fox’s work that an index is a kind of adjective. In the case of a simpler index, for instance, one that varies over entities, we can give it a denotation that makes it a simple predicate that holds of any individual that the value of the index refers to.

(14) \[ n = \lambda x. x = n \]

For an index that has a function in it, we can adopt the denotation in (12).

(15) \[ f = \lambda x. x = f(y) \]

where “\( y \)” is the argument of \( f \).

A definite description with an index in it, then, will have a denotation like that in (13).
In a donkey anaphora context, the $y$ in this formula will be bound by a quantificational expression, and the DP will pick out different donkeys depending on the value of $y$.

Rullmann and Beck (1998) suggest that we think of wh-phrases as kinds of definite descriptions and Fox (1999) took this idea up and suggested that the lower copy of a movement relation be given a denotation like a definite description built upon a definite determiner that comes with an index, like the one in (13). I suggested in Johnson (2012) that we incorporate these ideas into a multidominant representation in the following way.

First, assume that the denotation for which is the same as the denotation for the.

(14) $\langle$ which $\rangle = \lambda x. P(x)$, defined when there is exactly one $x$

in $s$ such that $P(x) = 1$.

We must also assume that which necessarily has an index in its complement. One way this has been achieved is to assume that the index is part of the determiner (see Elbourne 2005 and Schwarz 2009, for examples). We could, for instance, assume that which has a denotation like (15).

(15) $\langle$ which $\rangle = \lambda n. \lambda x. P(x) \land x = g(n)$,

defined when there is exactly one $x$ in $s$ such that $P(x) = 1 \land x = g(n)$.

The syntax that matches (15) is:

(16) $\langle$ which $\rangle$ = $\langle$ book $\rangle$ $\land x = g(2)$

One way of modeling demonstratives, such as that, is as definite determiners with an index built in. On this picture of which, it is more closely allied with demonstratives than plain definite determiners. There may be other ways of modeling how a definite combines with its obligatory index – see especially Hanink (2018) and Hanink and Grove (2017) – but this is what I will adopt here.

To ensure that which is used only in a question context, I suggested that this is the form of the determiner that arises when it is in an Agreement relation with our erstwhile mysterious X. It’s X that carries the meaning that makes questions: that is, it’s the term that introduces an existential quantification over $f$. Let me rename X, Q; its denotation will be:

(17) $[Q] = \lambda W. \lambda p \exists f. W(f)(p)$

These pieces go together as shown in (19) on the next page. (Recall that a dashed line is used to represent a position some phrase is in syntactically but not semantically.) Because Q does not semantically combine with its sister, the DP which picture, and as a consequence the QP that embeds it, has the same denotation that Q has. This is required by our observation that the variables in the DP which picture cannot be bound in the higher position. What we’re seeing here is plausibly what we saw in certain cases of Verb Movement: the moved verb is not semantically interpreted in its higher position, but just in its lower position. In this context, then, the DP is interpreted in just its lower position and not at all in its higher position.

Recall that when Terseness is violated in cases of Phrasal movement, it is not the phrase that occupies the two positions – the moved phrase – that is spoken twice, that is what we see only when a head moves. When a phrase moves, Terseness is violated only by speaking a pronoun in the lower position. This should follow from the syntax of movement, and it’s achieving that goal which animates the path I’ve taken here. By putting in the lower position all of the material that is normally associated with definite descriptions, we bring the syntax of the phrase in the lower position closer to what a pronoun is. At least that is correct if pronouns have the syntactic shape like that in (8). By moving the parts of a wh-phrase whose semantics are associated uniquely with questions, and not pronouns, I’ve segregated syntactically the information that could potentially be found in a pronoun from that which isn’t. I’m still a few steps from being able to spell out how violations of Terseness in these contexts might be modeled, but we should face now a problem that the steps already taken in that direction might pose.

The problem is that if the only way a variable meaning can be created in the lower position is by putting a DP with an index in that position, this predicts that movement will only create a variable-binding relationship when a DP has moved. Because the semantics offered here for questions requires that the lower position of a wh-phrase be occupied by a variable, this predicts that only DPs should move in constituent questions. One sort of counter-example to that arises in cases of Pied-Piping. In Pied-Piping, a phrase that contains the wh-phrase moves, and this phrase can, itself, be a DP (as in (18a)), but it needn’t be, as in (18b).

(18) a. Whose problems will we solve?

b. About which problems will we talk?
The denotation for DP, and every phrase connected to DP by a solid line up to \( \lambda_1 \), is defined only if there is a unique \( x \) for which \( \llbracket \text{picture} \rrbracket (x) = 1 \) and \( 2(1) \) is that \( x \).

The syntax of Pied-Piping presents a puzzle all by itself, of course, and not one that I will be able to engage in detail in these lectures. The solution that dovetails best with syntax we've developed so far is one in which more of the phrase that's moved is not semantically interpreted in the higher of its two positions. For instance, we might associate (18b) with (20).
I shall take for granted that such a syntax is possible. What is necessary is to understand how to give an explanation for which class of things Q can attach to and still be proximate in the required way to Agree with the D⁰ that contains which. See Cable (2007, 2010b,a) for suggestions.

If a syntax for Pied-Piping of this sort is achievable, then examples like (18) don't present a problem for the thesis that the lower position of all wh-movement is occupied by a definite description that functions as a variable. But examples involving wh-movement of APs might.

(21) How pleased with herself, was almost every woman, at the conference?

I'd like to suggest that these examples include a hidden DP within them.

Let's take a closer look at APs. They are thought to include a degree head, which is often silent but sometimes pronounced:

(22) Mary is so happy.
    Mary is that happy.

We can think of adjectives of this kind as naming relations between entities and degrees. I'll take the view that APs are embedded within Degree Phrases, in the way shown. One of the ways of pronouncing Deg is how. Indeed, we can think of how as having the semantics, and perhaps also the syntax, of which degree. We can see that how questions can also be understood as introducing quantification over functions from examples like (23).¹

(23) How rich is almost every heiress?
    Answer:
(24) more than her chauffeur

Indeed, we can think of how as having the semantics, and perhaps also the syntax, of which degree. This would allow us to assign to (23), the representation in (25).

¹ An example I owe to Irene Heim.
(25) \[ \lambda p \exists f.p = \forall x \text{heiress}(x) \rightarrow x \text{is} (\text{id} d = f(x) \land d \text{is degree})\text{-rich} \]

We can think of Q as being related by Agreement to how in the same way that it is related to which in simple cases like which picture.

Let's now engage the question of how these structures yield pronouns in the lower position when Terseness is violated. We want to find a syntax that allows a portion of the material in the lower position to not be subject to Antisymmetry and match how a pronoun is pronounced.

There are two scenarios I know of in which a pronoun is pronounced in the lower position of a movement chain. In one, a wh-phrase leaves a pronoun in an intermediate position. These are found in the constituent questions of certain African languages. I'll hold off on looking at these for a moment. The other scenario is one in which the pronoun is spoken in just its lowest position, and not any of the intermediary ones. To my knowledge, these examples are most prevalent in relative clauses or clefts, which could have the same basic syntax. How can we get resumptive pronouns out of this system? I want to take a clue from the fact that most cases of resumptives occur in relative clauses. The resumptive that we started with, found in Arabic, is like that. And that's where we find resumptive pronouns in Modern Hebrew as well, which also shows the connection between islands and reconstruction effects. I should have a better understanding of the typology of resumptive pronouns before seeing how well the machinery I've outlined here can describe them. But because resumptive pronouns clearly arise in relative clauses, I'll start sketch how we might produce them by focusing on deriving (26).

(26) The lower position of a movement relation can be pronounced as a pronoun if the movement relation is building a relative clause.

Relative clauses don't have the semantics of questions, but they do still have the variable binding property that wh-questions invoke. That we can get reconstruction effects in English relative clauses can be seen (dimly) from (27).

(27) A framer examined the picture of herself that every woman brought into the shop.

\[ \text{compare:} \]

* A framer examined the picture of herself that every woman's father brought into the shop.

Interestingly, (except in certain specialized examples) we don't get reconstruction effects of this kind when other quantifiers are involved.

(28) * A reporter wanted to see the picture of himself that no actor revealed to his mother.
Or do we?

(29) A reporter wanted to see the picture of himself that no actor would show his mother.

I’m unsure of the generalizations in this domain.

Sometimes universal quantifiers can appear to take exceptionally wide scope. I will assume that this is what distinguishes (27) from (28). I will construe this as evidence that it is necessary for the quantifier inside the relative to be able to scope out of the relative clause in cases of reconstruction. (I have not found a satisfying account of the conditions under which reconstruction like this in relative clauses is possible. See Sharvit (1996, 1999a,b), Chierchia (1992), Jacobson (1994), Szabolcsi (1997.) The contrast in (27), however, shows us that the quantifier must also c-command the lower position of relative clause movement. This suggests that the NP part of a relative clause is being semantically interpreted in both positions, and therefore that any variables it contains must be bound from the same thing in both positions.

Relative clauses have no Q, and consequently no wh-determiner. They have, instead, just plain vanilla determiners – the one that we see, and, if we have the right model of the variable in a movement chain, the one that is in the lower position. Unless we change the denotations for determiners, this will prevent us from using the same architecture for relatives that we have used for questions. This would amount to making a definite description (the lower DP) the complement of an indefinite, or another definite, determiner (the higher determiner). Instead, I suggest we take the NP to be the term that has moved in relative clauses, giving us a representation like that in (30).

As with questions, the mysterious “X” and “XP” is required if multidominant structures only arise by parallel merge. We should look for the identity of X, and count it as a strike against my hypothesis if it cannot be found. One possibility, I suppose, is that X is simply the higher determiner. This boils down to the “raising” analysis of relative clauses (see Vergnaud 1974 and Bianchi 2000).
But the semantics of this structure is obscure. We would hope to get the meaning out of a relative clause by conjoining its denotation with the denotation of the preceding NP. Because the relative clause combines with a CP here, we’d have to fiddle with the semantics to get things to work. In any case, if both the determiners in this structure have denotations that require them to semantically combine with the NP/XP they are sisters to, then the NP will have to be semantically interpreted in both positions of a relative clause.

Our linearization scheme will force the NP in (30) to be linearized in just one of its positions, and this is what happens in English relative clauses. If this is the right structure for relatives, it means that English must have a silent definite determiner which is used in the lower position, because nothing is spoken in that position in English. (That is why I’ve put the determiner in shaded font in (30) and (31)).

Consider now a language in which $\phi$-features can be part of a DP that has an overt NP. This would be a language, then, which allows DPs like (32).

We can use predicate conjunction to put the parts of the determiner’s complement together. If DPs like this are available – and if DPs can Agree with $\phi$ features, then this seems plausible – we should expect languages that build relative clauses as in (33).

Recall that pronouns are the way that definite descriptions are pronounced when their NP part just has $\phi$ features. The NP that is the complement to the lower the in (33) consists only of $\phi$ features – at least in terms of its string information that is all that is inside it – and perhaps this allows it to be pronounced as a pronoun.
References


Quantifier Raising

Quanti fier Raising is the rule that allows the quantificational determiner every in (1) to be interpreted in a position higher than where it is pronounced. This is required to give us the interpretations where the indefinite’s value varies as a function of the universal quantifier.

(1) A guard stood in front of every bank.

In its original formulation – which I think goes back to Chomsky (1977), but is usually credited to May (1977) – movement is allowed to create the representation in (2) “covertly” from (1).

(2)

\[
\begin{array}{c}
\text{TP} \\
\text{QP}_2 \\
\text{every bank} \\
\text{DP} \\
\text{a guard} \\
\text{T}^0 \\
\text{VP} \\
\text{stood} \\
\text{P}^0 \\
\text{in} \\
\text{N}^0 \\
\text{front} \\
\text{P}^0 \\
\text{of} \\
\text{x}_2 \\
\end{array}
\]

\[
\begin{array}{c}
\text{TP} \\
\text{L}_2 \text{a guard stood in front of 2} \\
\text{DP} \\
\text{TP} \\
\text{a guard} \\
\text{T}^0 \\
\text{VP} \\
\text{stood} \\
\text{P}^0 \\
\text{in} \\
\text{N}^0 \\
\text{front} \\
\text{P}^0 \\
\text{of} \\
\text{x}_2 \\
\end{array}
\]

A standard semantics for every gives us the interpretation in (4).

(3) \[\text{every} = \lambda Q. \lambda P. \forall y. Q(y) \rightarrow P(y)\]

(4) \[\forall y. y \text{ is a bank} \rightarrow \text{a guard stood in front of } y\]
∀y. y is a bank → a guard stands in front of ix, x = y such that x is a bank

TP

λQ. ∀y. y is a bank → Q(y)

QP

every bank

∀y. y is a bank → a guard stands in front of ix, x = n such that x is a bank

TP

λn. every bank

∀y. y is a bank → Q(y)

QP

The object DP, and everything up to TP*, has a value only if n is a bank.

(6) Trace Conversion

Interpret [DP D NP] as [DP the, NP], where

[the] = λP : there exists exactly one x, x = g(n) such that P(x) = 1. ix, x = g(n) ∧ P(x)

(The material between “:” and “.” in (6) expresses a presupposition. That is, for the denotation of “the” to be “ix, x = g(n) ∧ P(x)” it must be the case that the context in which “the” is uttered be one in which there exists exactly one x such that x = g(n) and P(x) = 1 in the situation being described.) Trace Conversion causes the lower copy to be interpreted as a definite description. It introduces the presupposition that the object it refers to is whatever the NP part describes – in our example, that the object refers to a bank. Notice the extreme overkill on that bank business. The presupposition restricts the quantification to banks, the restrictor on the QP does too, and the definite description further more makes sure that the object it refers to is a bank. The rules of Spell Out cause just the lower copy to be pronounced, and that is one of the central interests of Quantifier Raising.

Johnson (2012) translates this into a multidominant structure without changing the semantics. That’s done by sharing the NP part with two separate heads: a definite determiner downstairs and a quantifier upstairs.

∀y, y is a bank → a guard stands in front of ix, x = y ∧ x is a bank

TP

λQ. ∀y, y is a bank → Q(y)

QP

The object DP, and everything up to TP*, has a value only if n is a bank.

I have put the index inside the NP – I’ll explain why momentarily.
Note that on all these accounts, the NP part of the moved phrase is interpreted in both of its positions. Unlike what we saw for questions, the NP part of the moved phrase must be interpreted in its higher position if we are to restrict the quantification. We can see that this is a desirable outcome by considering the fact that (8) does not allow an inverse scope reading.

(8) A guard stands in front of every picture of himself.

Because picture of himself must be interpreted together with the universal quantifier, it cannot be interpreted lower than where the universal quantifier is interpreted. And because himself must be interpreted within the scope of a guard, the result is that the universal quantifier must too. We should keep a semantics for the universal quantifier – the standard one will do – that has this effect. Rather than Trace Conversion, this model separates the two meanings of QR: the quantificational part (=Q) from the variable part (=D). Q, then, has the denotation that the other models ascribe to every. every is the way that the definite determiner gets pronounced when it's expressing the presence of a universal quantifier somewhere else.

But unlike questions, we cannot rely on Agreement or the like to characterize when the definite determiner is in the right relation to Q to be pronounced as every. Because Q does not c-command every, we cannot use Agreement. In my 2012 paper, I had a solution to this problem that I suggested might explain some of the spell out properties of QR. I will set this problem aside for this class, but it'll be necessary to have a description of how the Q in the high position is related to its exponent in the D⁰ of the lower DP. We do not want our syntax to give to (9) a representation like (10).

(9) She will read every book about the problem.

If that were possible, (9) would have the interpretation that She will read the book about every problem. We need to ensure that the universal quantifier is semantically associated with the NP that every is syntactically associated with. Something must ensure (11).

(11) Q must merge with a projection of the noun that heads the complement to Q's exponent.

(11) requires that ∀ in (9) merge with a projection of book, the noun that heads the complement to every.

QR and Wh Movement differ, then, in how the exponent of the operator that gives scope to the relevant DPs is expressed. They also differ in how the linearization algorithm produces a string. We must force the linearization algorithm to map a QRd DP into a string position that corresponds to the lower of the two positions it occupies. Wh Movement, by contrast, engages the linearization algo-
arithm in English to map that phrase onto the string position assigned to the higher of the two positions it occupies. We’ll return to how to frame this difference.

QR is thought to be responsible for resolving Antecedent Contained Deletion (ACD).

(12) Joan accused everyone that Sherlock did \( \triangle \).
\( \triangle = \text{accused} \ x \)

That’s because there seems to be a match between the scope of the quantifier and the size of the VP that can be elided. (see Williams (1977) and Sag (1976).)

(13) Joan wants to eat everything Sherlock does \( \triangle \).

a. *every > want
i. Everything that Sherlock wants to eat is something that is desirable for Joan to eat.
ii. *Everything that Sherlock eats is something that is desirable for Joan to eat.

b. *want > every
i. Eating everything that Sherlock eats is what is desirable for Joan.
ii. *Eating everything that Sherlock wants to eat is what is desirable for Joan.

On the copy and multidominant theories of movement, this isn’t trivial. If the NP part of a QRd phrase is semantically in both the lower and higher position, then we might expect a relative clause to be in both those positions too. That would not resolve ACD. Fox (2002) argues that the ability of merge to attach things inside other phrases – what is sometimes called “Late Merge” – is responsible for resolving the ellipsis. A picture of the syntax that achieves that, but which ignores the internal construction of the relative clause, is (14).

(14) \( \forall y. y \text{ is a person that Sherlock did <accuse>} \rightarrow \text{Joan accused } ix [x=y \land x \text{ is a person}] \)

We’ve looked a little bit at how relative clauses work on this multidominant model, and it can be combined with QR in the way illustrated by (15).
(15)  \( \forall y. [y \text{ is a human that Sherlock accused}] \to \text{Joan accused} \)

\[ \lambda Q. \forall y [y \text{ is a human that Sherlock accused}] \to Q(y) \]

\[ \lambda 2 \text{Joan accused } x = 2 \land x \text{ is a human} \]

\[ \forall y. [y \text{ is a human that Sherlock accused}] \to Q(y) \]

Remember that “X” here is the mysterious term that I invoked in giving the syntax of relative clauses. It’s required by our parallel merge style derivations. I’ve made the relative clause be the result of “head movement” – that is, a relative in which the so-called “head” of the relative (the NP) moves from inside the relative. This is not something easy to achieve in a copy theory of movement, but it’s within reach of a multidominant theory. The boxed VP deletes in (15) in order to produce (12).
Fox and Nissenbaum (1999) argues that QR provides essentially the same syntax that produces “extraposition from NP,” a construction type illustrated by (16).

(16)  John accused everyone yesterday that Sherlock did $\Delta$.

In this construction, the relative clause seems to have “moved” rightwards away from the DP it modifies. In the analysis in Fox and Nissenbaum (1999), the relative clause hasn’t moved. Instead the entire DP has moved by QR. But unlike the normal situation with QR, the relative clause is linearized according to the higher position that QR assigns to the DP it affects. This position, apparently, gets mapped onto a string in English that makes it follow everything it is a sister to. Thus, the higher position of QR is to the right of the lower position, and when the relative clause is linearized in that position it necessarily follows the lower position as well. The rest of the QRd DP is linearized in the normal way: according to its lower position. Fox and Nissenbaum (1999) argue that the reason the relative clause has this exceptional position is because it is only in the higher of the two positions that QR assigns to the DP. There is no relative clause in the lower position. Below there are two parses of this scenario: one includes the details of the internal structure of the relative clause (18), and in the other those details have been obscured (17).

(17)  $\forall y. y$ is a person that Sherlock did <accuse> $\rightarrow$ Joan accused $\forall x [x = y \land x$ is a person]
(18) \[ \forall y. [y \text{ is a human that Sherlock accused}] \rightarrow \text{Joan accused } x = y \land x \text{ is a human} \]

\[
\begin{array}{c}
\text{TP} \\
\text{TP*} \\
\lambda \exists \text{ Joan accused } \bigwedge \text{ x is a human} \\
\lambda Q. \forall y. [y \text{ is a human that Sherlock accused}] \rightarrow Q(y) \\
\end{array}
\]

The arguments for this analysis in Fox and Nissenbaum (1999) rest on a correlation they find between the position of the extraposited relative and the scope that the modified DP takes when it is headed by a quantifier. The relevant data are somewhat tricky, however, so let me report here other arguments on behalf of this analysis.

Overfelt (2015b) points out that this analysis solves the mystery of why relative clauses do not seem to be movable otherwise:

(19) * That Sherlock accused, John accused everyone.

And, as Overfelt (2015a) demonstrates, this also accounts for why an extraposited relative behaves semantically as if it is part of the restrictor for the quantifier with respect to NPI licensing. The NPI ever is licensed when it is in the restrictor of the quantifier every, but not when it is in the nuclear scope of every.

(20) a. We met every biker who has ever ridden on these trails.
    b. * We met every biker while he has ever ridden on these trails.

This is a fact about every, and not a fact about relative clauses more generally, as we can see from the contrast in (21).

(21) a. We met every biker who has ever ridden on these trails.
    b. * We met some biker who has ever ridden on these trails.
This is preserved under extraposition, just as the account by Fox and Nissenbaum (1999) predicts.

(22) a. We met every biker yesterday who has ever ridden on these trails.
    b. * We met some biker yesterday who has ever ridden on these trails.

Fox (2002) argues, following Baltin (1987), that relative clauses which have ACD in them are always extraposed. That is, he argues that the position a relative clause is pronounced in is the position that resolves its ellipsis. One piece of evidence on behalf of that idea is (23), which is from Tiedeman (1995).

(23) a. * I said that everyone you did \( \Delta \) arrived.
    b. I said that everyone arrived that you did \( \Delta \).

\( \Delta = \text{said that } x \text{ arrived} \)  
(Fox 2002, (35b), (36b): 77)

The word order in (23a) requires that the relative clause be inside the VP headed by \( \text{said} \), whereas in (23b), it’s possible for the relative clause to be outside this VP. If QR moves things rightwards, as these examples would suggest, it is often very difficult to tell when the relative clause is in or outside the VP. In (24), it could have moved string-vacuously out of the VP, and in (25), the PP could have itself moved farther to the right of the relative clause.

(24) She \( [\text{VP read every book } \uparrow] \) that you did.

(25) She \( [\text{VP read every book } \uparrow] \) that you did to me.

It is therefore often difficult to tell where the surface position of a relative clause is, and this, Fox argues, oftentimes gives the illusion that the relative is inside a VP that antecedes the ellipsis. It is only in carefully controlled examples, like (23), that we see that when a relative clause contains an elided VP, its surface position must be outside of that ellipsis’ antecedent.

As noted above, we need the string part of our system to assign to (26) the string in (27), and to (28) the string in (29).

(26)

(27) She said that everyone arrived that you did \( \Delta \)

(28)

(29) She said that everyone that you did \( \Delta \) arrived

If we find a way of ensuring that the NP which is a sister to the D always gets linearized according to the path that contains DP, then this contrast does, indeed, emerge. (Though the fact that the QRd material shows up to the right of its sister, rather than to the left, does not.) We’d like to derive (30).

(30) a. \( p([DP \text{ every}]) \) (the path of every) must be included in the path of \( N^0 \), the head of \( D^0 \)’s complement.
    b. The sister of QP precedes QP
Nissenbaum (2000) has an idea about (30b). The idea rests on the English particular fact that relative clauses are linearized so that they follow the phrases they modify. We can express this in the terms of our linearization algorithm with (31).

\[ \forall x, y, x \in d(NP) \land y \in d(CP), x < y, \text{ where CP and NP are sisters} \]

If (31) holds of relative clauses that "late merge," then this will derive the fact that relative clause are linearized to the right of the material they are "extraposed" from. As Nissenbaum notes, this also derives the fact that DP internal material that English linearizes so that it precedes the head noun cannot extrapose. For instance, certain APs cannot follow the NP they modify, but must instead precede them. English adopts (33).

\[ \forall x, y, x \in d(NP) \land y \in d(AP), y < x, \text{ where AP and NP are sisters} \]

If we tried to Late Merge an AP of this kind, we'd have a structure like (34).

\[ \forall \, x, y, \, x \in d(NP) \land y \in d(AP), \, y < x, \text{ where AP and NP are sisters} \]

Finally, let's examine how the Fox and Nissenbaum (1999) account of extraposition captures a fact reported in Rochemont and Culicover (1990). Consider a case in which extraposition has affected a PP, as in (40).

\[ \forall \, x, y, \, x \in d(D^0) \land y \in d(NP), \, x < y, \text{ where } D^0 \text{ and NP are sisters} \]

This condition blocks (36).

This idea also accounts for the relative order of more than one phrase that has extraposed. As Keller (1995) notes, when a PP and a relative clause extrapose, they must show up in the same order that they would if they hadn't.

\[ \forall x, y, \, x \in d(D^0) \land y \in d(NP), \, y < x, \text{ where } D^0 \text{ and NP are sisters} \]

(38) a. A man with blond hair who was smiling just came in.

b. * A man who was smiling with blond hair just came in.

(39) a. A man came in with blond hair who was smiling.

b. * A man came in who was smiling with blond hair.

(Keller 1995, (13), (15): 301)

(40) a. Mary saw an alleged mouse from Mars yesterday.

b. Mary saw an alleged mouse yesterday from Mars.

Rochemont and Culicover (1990) reports that from Mars is necessarily outside the scope of alleged in the second example, but can fall within the scope of alleged in the first example. This would follow if there was something about extraposition that required the extraposed phrase not be too deeply embedded inside the DP it modifies. On traditional accounts, that involve moving the extraposed phrase, this can be captured by adopting a locality condition on movement that prevents the extraposed phrase from moving through too much of the DP it modifies. If PP extraposition involves the syntax Fox and Nissenbaum (1999) suggest holds for relative clauses, then another way of expressing this locality condition will be required.

To see how this can be achieved, consider the representation (= (41)) assigned to (40b) by our syntax.
(41) \( \exists y. y \) is a mouse in \( w \) \& from Mars in \( @ \) \& Mary saw \( ix.x = y \) \& \( y \) is a mouse in \( w \)

\[
\lambda Q. \exists y. y \) is a mouse from Mars \& Q(y)
\]

By contrast, the case in (40a) has the structure in (42), which would allow from Mars to be part of what is alleged.

Making the many necessary translations, this is basically Culicover and Rochemont (1990)’s account. Let’s work through how these structures get semantically interpreted.

An important ingredient necessary for characterizing the meaning of these examples is Keshet (2010)’s condition in (43).

(43) Intersective Predicate Generalization
Two predicates interpreted intersectively may not be evaluated at different times or worlds from one another. (Keshet 2010, (10): 388)
The Intersective Predicate Generalization is supported by the fact that sentences like (44) are anomalous.

(44)  

a. # Mary thinks the married bachelor is confused.  
b. # Mary thinks the professor in college is too young to teach.

(Perhaps a better example for (44b) would be Mary thinks the professor in his freshman year is too young to teach.) Without (43), it should be possible to interpret married in (44a) in the worlds of the speaker of (44a), and bachelor in the worlds of Mary’s belief. That would deliver a non-contradictory interpretation of (44a). It would report that the married man that Mary thinks is a bachelor is thought by Mary to be confused. That isn’t a meaning (44a) has, however, and it’s blocked by (43) because the worlds in which married and bachelor are evaluated are different. Instead, the meanings that (44a) has are contradictory. It reports either that the speaker has a contradictory belief or that Mary does. That contradictory belief is that the individual being described as confused is both married and a bachelor (hence, unmarried). The example in (44b) illustrates a similar effect, but with respect to the times at which professor and in college are evaluated.

Keshet’s condition forces predicates that are all inside the scope of alleged in our example to be evaluated at the same worlds: the worlds that characterize the allegation. This condition will only allow from Mars to be evaluated at the actual world only if it is out of the scope of alleged. It will therefore have to be merged to a phrase that includes alleged. Note that the NP formed from alleged is evaluated at the actual worlds – that is, the worlds that characterize the state of affairs that contain the allegation, not the worlds that characterize the allegation – and therefore any modifiers that combine with that NP will have to be evaluated at those, non-alleged, worlds. If from Mars merges to this NP, it will have to be evaluated at the actual worlds. In (41) and (42), I’ve put subscripts on the relevant material to indicate at which worlds they are evaluated: from Mars\textsubscript{w} indicates that from Mars holds in the actual worlds, whereas from Mars\textsubscript{w} indicates that from Mars holds in the worlds of the allegation.

On an account that treats extrapolation of PP as movement of the PP, rather than an outcome of QR, the fact that an extrapolated PP cannot fall within the scope of modifiers of the NP it is related to could be credited to a locality condition on movement. This is what Rochemont and Culicover (1990) suggest for the contrast in (40). The contrast in (40) arises because movement cannot span the distance in (45a), though it can in (45b).
I’m skeptical about such a constraint on movement, however, because it does not show up with leftward movement. In (46), it is possible for *from Mars* to be associated with the position marked by the trace in (45a).

(46) It’s from Mars that Mary saw an alleged mouse.

That is, *from Mars* can be part of what is alleged to be the thing Mary saw in (46). On the syntax proposed here, the fact that *from Mars* cannot be related to a position in the scope of *alleged* when it extrapolates follows from the condition on how the Q and its exponent are related. The condition in (11) requires that Q merge with a projection of the complement of its exponent. Because the complement of D⁰, which is where Q’s exponent lives, is always an NP, this means that Q will have to merge with an NP. Because it is the D⁰ that combines with the NP projected by *mouse* that expresses the exponent of ∀ in (40), the parse in (41) is the only one that satisfies (11) and puts *from mouse* in a position where it can be linearized farther to the right of the DP it modifies. I therefore regard the contrast in (40) as evidence for the multidominant-based syntax of QR sketched here. It builds in a locality condition between the quantifier and its exponent that derives the contrast in (40).

It also, incidentally, provides evidence for the restriction to parallel merge derivations that I’ve adopted as well. Consider what would be possible if remerge derivations were possible instead. We should be able to produce representations like (47).

\[
\exists y. y \text{ is a mouse in } w \land \text{ from Mars in } w \land \text{Mary saw } ix.x = y \land y \text{ is a mouse in } w 
\]

This would manufacture the reading that I think is absent. This is blocked, however, if the Extension Condition is enforced in the way that it is here. Recall, the Extension Condition requires that merge only be able to join terms that are elements (i.e., root nodes) in a Stage. That derives the constraint in (48).

(48) If α merges with β, then α cannot be (recursively) dominated by β.

What (48) proscribes is precisely what has happened in (47): *alleged* has merged with an NP that dominates *alleged*.
1 Hydras

In a QR-based account of Extraposition, then, we find evidence for some of the ingredients of my analysis of movement. There is a specialized instance of Extraposition that I think provides evidence for that part of my proposal that says that what is in two positions in QR is not the entire QP/DP, but the NP inside it instead. That specialized instance of Extraposition can be found in (49).

(49) A man entered the room and a woman went out who were quite similar.


In (49) is an extraposited relative clause: who were quite similar. This relative clause must modify a plurality, and yet in (49) there are only singular DPs. Somehow, what has happened here is that the relative clause manages to modify a plurality that is created by the two singular DPs. Sometimes this construction is said to give the relative clause “split antecedents.”

One approach to these examples would be to assume that associated with the extraposited relative is a silent plural pronoun whose antecedent is the plurality invoked by the previously occurring indefinites. Schematically, something like (50).

(50) A man1 entered the room and a woman2 went out pro1,2 who were quite similar.

We might expect the meaning on this analysis to be roughly that in (51).

(51) A man entered the room and a woman went out and they were quite similar.

See, for instance, the introduction in Webelhuth, Sailer, and Walker (2013), where something of this kind is sketched. I think this is a likely source for (49).

But it won’t spread to examples with better quantification:

(52) a. Every man smiled and every woman frowned who had met each other at the open house.

≠ Every man smiled and every woman frowned and they had met each other at the open house.

b. No man smiled and no woman frowned who had met each other at the open house.

≠ No man smiled and no woman frowned who had met each other at the open house.

The relative clause serves to restrict the quantification in these examples. (52) have interpretations parallel to (53).

(53) a. Every man and woman who met each other at the open house smiled and frowned respectively.

b. No man and woman who had met each other at the open house smiled and frowned respectively.

That the relative clauses are serving the role of restricting the quantification might be better appreciated by considering an example like:

(54) Every triangle is small and every circle is big that are connected to each other by a line.

This sentence is true in the situation portrayed by the drawing below.

(55) [Diagram]

We want a syntax, then, that ensures that the relative clause combines semantically with the NP parts of each quantifier so that it can contribute to restricting the quantification.

Every English speaker I have confronted with examples like these finds that (52) and (53) do not have the same status as (49). There is uniform and easy consensus on the grammaticality status of (49) that is not found with (52) and (53). Some I have consulted classify (52) and their ilk as ungrammatical, while others – the majority in my informal poll – grant them grammaticality but mark a clear degradation compared to (49). My suspicion is that this is because they have a different syntax – (49) gets a parse like (51), for instance – and that these different syntaxes are not equally accessible. Perhaps, for instance, the syntax associated with (52) presents complexity for the online processor that is absent in the syntax that lies behind (49), and this contributes to their differing grammaticality perceptions. The model I will offer in a moment requires of (52) and (53) a dependence on movement – namely QR – whereas (49) might arise through anaphora. It is not far-fetched, I believe, to imagine that the discontinuous dependencies that movement invoke have a larger tax on the resources required of online processing than do anaphora. Whatever the source of the contrast between (49) and (52), the English speakers I’ve consulted have no hesitation assigning to (52) and (53) the meanings I’ve described. I will set aside the uncertain grammaticality status of (52) and (53), and here focus on what seems clear: that (52) and (53) get interpretations that cause the relative clause to restrict the quantifications involved.
Zhang (2007) argues that this must be achieved by syntactically relating the extraposed relative clause to each of the DPs in these constructions, pointing to the ungrammaticality of (56).

(56) a. * Mary met him and John met a woman who knew each other well.
    b. * Mary met Bill and John met a woman who knew each other well.

This can be related to the fact that restrictive relative clauses cannot be in construction with pronouns or names in English.

(57) a. * He who left arrived later.
    b. * John who left arrived later.

Might these constructions involve Right Node Raising? Maybe not, because Right Node Raising doesn't seem to support the plurals that these constructions somehow manufacture.

(58) a. * Sue's proud that Bill and Mary's glad that John have finally met.
    b. * Bill is proud that Hilary and Michelle is glad that Barack support each other.
    c. * Bill was glad that Sue, George was relieved that Jane and Jim was reassured that Mary outnumbered the gangsters.

(Grosz 2015, (21): 11)

compare:

(59) a. Sue's proud that Bill and Mary's glad that John has/?have left.
    b. Bill is proud that Hilary and Michelle is glad that Barack are/?is happy.

Right Node Raising is hard to diagnose, and its properties are spooky, so it's difficult to know whether this really tells us anything about our examples. I'll tentatively assume that Right Node Raising isn't involved.

In any case, we want an account which correctly allows the relative clause to restrict the quantification in each of the separate conjuncts. With just two wrinkles, this can be achieved with our present system. We'll need to assemble a few ingredients first, though.

First, because we are now suddenly dealing with plurals we have to become cognizant of some of the meshugas about plurals. Let us assume, with Link (1983), that NPs denote the set of individuals that are formed by taking a set of atomic individuals and closing them under sum formation. So, for instance, if the domain of women is just Sheila, Marcia and Pat, we have:

(60) a. [woman] = {Sheila, Marcia, Pat}
    b. [women] = {Sheila, Marcia, Pat, Sheila®Marcia, Sheila®Pat, Marcia®Pat, Marcia®Pat®Sheila}

or:

a. [woman] = λx woman(x)=1.
    b. [women] = λx ∀y[y ≤ x ∧ y atomic] → woman(y)=1

where α ⊕ β is an individual with parts α and β and ≤ means “is a reflexive part of.” With plurals we also have to recognize that definite descriptions build in a maximality operator. In the present context, for instance, (61) is a claim about all the women in this class.

(61) The women in the class are alert.

So, ignoring the presupposition that the definite determiner invokes, we can give the denotation of the with (62).

(62) [the] = λP. max(P).
    max(P) = ix [P(x) ∧ ∀y[P(y) → y ≤ x]]

We need the same maximality operator for our indexed definite determiners, the one that makes definite descriptions that can vary. We can see that from examples like (63).

(63) A child of [every man and woman]i keeps [the man and woman]j awake all night.

(A less prolix way of saying the same thing (65) does is to replace the man and woman with them.) In this example, every man and woman must scope above a child for the man and woman to be understood as bound to it. This is (probably) not an instance of e-type anaphora, then. It requires every man and woman to QR out of the subject DP it is spoken within and bind the man and woman. This is evidence that the definite DP the man and woman contains a variable that is bound by every man and woman, rather than the machinery of e-type anaphora, since the presence of that index inside the man and woman would require every man and woman to c-command it for the bound variable interpretation to result.

Next, we have to consider how coordinations work. We'll only look at coordinations of NPs, and we'll start with simple cases where the NPs that are coordinated form sums. That's the case in (64).

(64) The man and woman met in the park.

We want and to give us a plural individual that is made up of a man and a woman in this case. We want and to have the denotation in (65).
(65) \[ \llbracket \text{and} \rrbracket = \lambda P. \lambda Q. \lambda x. x = y \oplus z \land P(y) = 1 \land Q(z) = 1 \]

So:

(66) \[ \llbracket \text{the man and woman} \rrbracket = \]
\[ \max(\llbracket \text{man and woman} \rrbracket) = \]
\[ \max(\lambda x. x = y \oplus z \land \text{woman}(y) = 1 \land \text{man}(z) = 1) = \]
\[ \lambda x. x = y \oplus z \land \text{woman}(y) = 1 \land \text{man}(z) = 1 \land \]
\[ \forall w[\llbracket w = y \oplus z \land \text{woman}(y) \land \text{man}(z) \rrbracket \rightarrow y \leq x] \]

We can learn from (67) that we need to change how the maximality operator works in the case of the indexed definite determiner.

(67) A friend of \[ \llbracket \text{every man and woman in the class} \rrbracket \_] thinks that the man\(_i\) dances better than the woman\(_i\).

First, notice that we need \textit{the man} and \textit{the woman} to both be bound by the quantifier. The quantifier ranges over plural individuals that are made up of a man and a woman, but \textit{the man} and \textit{the woman} are variables that refer to singular individuals. What our present denotation for \textit{the man} and \textit{the woman} will do is require that the value the index gets is a man or is a woman. The values the index will have, however, is man@woman pairs, because that is what \textit{every man and woman} quantifies over. What we need is for \textit{the man} and \textit{the woman} to refer to just the part of those pairs that satisfies the \textit{man} or \textit{woman} predicates. So we change our denotation for indices to (68).

(68) \[ \llbracket n \rrbracket : \lambda x \ x \leq n \]

This gives us (69) for (67). (Forgive the departure from multidominance.)
(69) \[ \forall y \text{ man} \oplus \text{woman}(y) \rightarrow \text{a friend of } y \text{ thinks that} \]
\[ \max(\lambda x \leq y \land \text{man}(x)=1) \text{ dances better than } \max(\lambda x \leq y \land \text{woman}(x)=1) \]

TP

QP

every man and woman

TP

\[ \lambda \text{2 a friend of } 2 \text{ thinks that } \max(\lambda x \leq 2 \land \text{man}(x)=1) \]
dances better than \[ \max(\lambda x \leq 2 \land \text{woman}(x)=1) \]

TP

DP

T

VP

\[ a \text{ friend of } 2 \text{ think } \]

C

TP

max (\[ \lambda x \leq 2 \land \text{man}(x)=1 \]

T

D

P

V

AP

\[ \text{the } 2 \text{ man } \text{ dance } \]

better

P

\[ \text{max}(\lambda x \leq 2 \land \text{woman}(x)=1) \]
than

DP

\[ \text{the } 2 \text{ woman } \]
A paraphrase of the interpretation this syntax provides is (70).

(70) For every man-woman pair, \(x\), there is a friend of \(x\) that thinks that the maximal man-part of \(x\) dances better than the maximal woman-part of \(x\).

Now all we need to add to these background assumptions is a particular theory of the syntax of coordination. The idea is that coordination in English is just the usual syntax and semantics of intersective modification. That is, I will propose that (71) are completely parallel, except that (71b) has the special Spell Out rule in (72).

\[
(\lambda x \big(x\big) = 1 \land \text{woman}(x) = 1) \rightarrow \text{man}(x) = 1 \land \text{woman}(x) = 1 \land x \text{ came in together}
\]

Remember that (75).

(75) When Q and D share an NP, the exponent of Q is expressed in D.

The geometry in (74) predicts that the form of the determiners in each of the coordinated clauses should be the same, since they are both in the relation that (75) describes with the same Q. This seems to be true. (This fact is first, to my knowledge, reported in Moltmann 1992.)

With this assumption our model of QR provides the right syntax and semantics. What we envision is a derivation that takes the NP parts of each DP and merges them together before combining them with the quantifier. That is precisely the analysis of these constructions that Zhang (2007) proposes, and what I’ve done here is to make this a consequence of QR. (This is what is done in Fox and Johnson (2016), so the “I” here includes Danny Fox.)

\[
\forall y \big[y = w \land \text{man}(z) = 1 \land y \text{ came in together}\big]
\]

Remember that (75).

(75) When Q and D share an NP, the exponent of Q is expressed in D.
c. * Every woman is smiling and some man is frowning who came in together.
d. Some woman is smiling and some man is frowning who came in together.
e. * Most women are smiling and every man is frowning who came in together.
f. ? Most women are smiling and most men are frowning who came in together.
g. * Every woman is smiling and few men are frowning who came in together.
h. Few women are smiling and few men are frowning who came in together.
i. * No woman is smiling and every man is frowning who came in together.
j. No woman is smiling and no man is frowning who came in together.

There are examples, like (77) suggested to us by Ming Xiang, where determiner-like material in each of the subjects differs and the result is fully grammatical.

(77) One woman is smiling and other women are frowning who came in together.

I suspect that the differing material in this, and similar examples, is not the relevant quantifier, however. In (77), for example, the subject DP's plausibly involve the same existential quantifier and therefore also fit the prediction. That is, a fuller picture for (77) would provide a syntax whose denotation could be paraphrased with (78).

(78) There is an x who came in together, such that y, the maximal part of x that is one woman, is smiling and z, the maximal part of x that isn't y, is frowning.

In a syntax that produces (78), we can imagine a single existential quantifier being exponed in each of the subjects. This account also predicts that examples like (79) should be ungrammatical.

(79) Every woman is smiling and every woman is frowning who came in together.

This should have the same status as (80).

(80) Every woman and woman who came in together are smiling.

There is something that prevents two NPs from merging together (sometimes with the aid of and) if they are identical. Because that is the syntax that our account gives to (79), it should violate that constraint.

What hydras teach us, if this is correct, is not only that every, and perhaps other quantificational determiners, aren’t semantically where they are pronounced. This is the just the conclusion reached by any example calling for QR (or its ilk). We also learn that the NP part of the DP headed by every is shared with the unspoken higher term that does the quantification in the way that the treatment of QR sketched here says. It’s not, as standard theories have it, that the DP headed by every moves to a higher position and there binds a variable in its lower position. It’s the NP part of the DP headed by every that moves. That’s a result expected on a parallel merge version of movement, rather than a remerge version of movement.

References


Overfelt, Jason. 2015a. Extrapolation of NPIs from NP. Lingua 164:25–44.
Let's look at where we've arrived. I started by laying out the following goals.

1. a. Explain violations of Contiguity, i.e. derive semantic displacement.
   b. Explain Terseness and its exceptions:
      - When a term is displaced, it is pronounced in only one of the positions it occupies, except possibly:
        i. If the term is a verb, or
        ii. The term is a DP, in which case a resumptive pronoun is possible.
   c. Make (1a) and (1b) have the same source.

Multidominance has been our answer. With adjustments to the semantic pieces that make up movement examples, we've gotten closer to (1a). And when this is wedded to a linearization algorithm that can handle multidominant trees, we can implement Jairo Nunes’ idea that (1b) follows from the Antisymmetry condition. The particulars of the story have two parts: one that determines how phrases are built and another that determines how those phrases are mapped onto strings.

The phrase part is:

2. Derivation

   \( \mathcal{D}(S_i) \), a Numeration, is the series of Stages \( (S_i, S_{i+1}, S_{i+2}, \ldots, S_m) \), where:
   a. Each \( S_{k+1} \) is derived from \( S_{k-1} \), by at least one application of Merge.
   b. \( S_m \) is a singleton set (i.e., a root node)

\[ \text{MERGE}(\alpha, \beta) = \begin{array}{c}
          \gamma \\
        \alpha \quad (\beta)
        \end{array} \]

where:
   a. \( \alpha \) and \( \beta \) are root nodes in (i.e., elements of) a Stage.
   b. If one of the arguments of \text{MERGE} is a word, then \( \gamma \) must be a phrase.
   c. If \text{MERGE} has a sole argument, it must not be a phrase.
   d. \( \gamma \) is the category of \( \alpha \) or \( \beta \).
   e. \( \alpha \) and \( \beta \) must be certain categories.

3. Principle of Full Interpretation

   If \( \alpha \) is a member of a Numeration, \( S_i \), then \( [\alpha] \) must be part of the denotation of the output of \( \mathcal{D}(S_i) \).

   The string part is:

4. \( \mathcal{L}_{pc}(P) = \{ S : S = [\alpha < \beta, \alpha \text{ and } \beta X^0 \text{ s in } P] \} \), such that \( S \) satisfies Path Contiguity, Totality, Antisymmetry, and the language particular component.

   a. Let \( p(w) = (\text{XP}_1, \text{XP}_2, \ldots, \text{XP}_n) \), a path, be the phrases that dominate \( w \), an \( X^0 \), and includes the root phrase, where every \( \text{XP}_1 \) is dominated by every \( \text{XP}_{>1} \).
   b. \( \Pi(P) \) is a set of paths formed from the words in \( P \).
   c. \( \text{d}(\text{XP}) \) is the set of \( w \)s such that \( \text{XP} \) is in \( p(w) \).
   d. Path Contiguity

      If \( p \in \Pi \), then for every \( \text{XP} \in P \) and \( \text{YP} \), sister of \( \text{XP} \):
      i. \( \beta \in \text{d}(\text{YP}) \rightarrow \forall \alpha \in \text{d}(\text{XP}) \, \alpha < \beta \), or
      ii. \( \beta \in \text{d}(\text{YP}) \rightarrow \forall \alpha \in \text{d}(\text{XP}) \, \beta < \alpha \).
   e. Totality

      For every \( w \) in \( P \), \( \Pi(P) \) must contain \( p(w) \).
   f. Antisymmetry

      \( \neg(\alpha < \beta \land \beta < \alpha) \)

Our formula for the relation between the phrase structure of a sentence and its string is (6).

5. \( \mathcal{L}_{pc}(\mathcal{D}(S_i)) \)

From this system, in tandem with a semantics that spells out how determiners work, “movement,” with the properties in (1), almost emerges. There are two cases left unexamined however.

We have "shown" how resumptive pronouns can be produced, but only in relative clauses. They also show up in constituent questions, however, and so we're not quite done with (1b). And we've also worked out how reconstruction effects work – how variables inside a displaced phrase can be bound from the position...
the phrase is displaced from. But we haven't worked out the example that started that project:

(7) Which papers of hers that Mark brought will no woman talk to him about?

In this example, not only can hers be bound by no woman, but at the same time Mark can be interpreted as if it weren't c-commanded by him. Our model of wh-movement doesn't predict this. Because wh-moved phrases are interpreted in their lowest position, (7) is a surprise. Our analysis gives to (7) the representation in (8).

(8) 

\[ \lambda \mathbf{p} \, \exists \mathbf{p} \, p = \neg \exists x \, x \text{ is woman} \land x \text{ will talk to him about } i \, y. \, y = f(x) \land y \text{ is papers of } x \text{ that Mark brought} \]

The denotation for DP, and every phrase connected to DP by a solid line up to CP*, is defined only if there is a unique x for which \( \llbracket \text{papers of hers that Mark brought} \rrbracket(x) = 1 \) and 2(3) is that x.
The *about* phrase in this parse is unorthodox. I include it to make *to him* command the *which*-phrase. There are other, less perverse, solutions. But they will distract at this point.

Today I’m going to try to address these two leftovers. And also gesture at issues left that I do not know how to resolve.

Let’s start with the problem of resumptive pronouns. First recall how I suggested we get resumptives out of relative clauses. A putative example like (9) will get the representation in (10).

(9) A picture (of herself) that every woman is admiring it

(10)

\[
\text{DP} \\
\text{D}^0 \quad \text{XP} \\
\text{a} \quad \text{CP} \quad \text{XP} \\
\text{C}^0 \quad \text{TP} \quad \text{X} \quad \text{TP} \\
\text{that} \quad \text{DP}_1 \quad \text{TP} \\
\text{every woman} \quad \text{T}^0 \quad \text{VP} \\
\text{is} \quad \text{V}^0 \quad \text{DP} \\
\text{admiring} \quad \text{D} \quad \text{NP} \\
\text{the} \quad \phi \quad \text{N} \\
\text{picture (of herself)}
\]

Assume that a pronoun can be the spell-out of:

(11)

\[
\text{DP} \\
\text{D} \quad \text{NP} \\
\text{the} \quad \text{sing}
\]

And the structure in (10) allows for the possibility that the DP in the lower position can be spelled out as a pronoun. If the linearization procedure puts the string *pictures (of herself)* into the higher position, then the material in the lower position that could be pronounced there is just what (11) has. What’s key to this story is that in relative clauses, the determiner that heads the phrase in the lower position is not also in the phrase that occupies the higher position. For this reason, it, and any features that might remain unshared, can make up material that is a pronoun.

Our syntax of questions, however, doesn’t have that syntax. As (8) shows, the DP in the lower position is entirely in the higher position to. There is nothing in the lower position that isn’t also in the higher position. Path Contiguity will require all of that material be spoken together, blocking a resumptive pronoun.

The structure we gave to QR, however, shares the relevant properties of relative clauses in (10). Like relative clauses, QR puts the NP part of two DPs in different positions, but leaves the determiners out of the mix.

(12)

\[
\forall \ y . y \text{ is a bank} \rightarrow \text{a guard stands in front of } \lambda x . x = y \land x \text{ is a bank}
\]

\[
\text{DP} \\
\text{TP}^* \\
\lambda Q . \forall y . y \text{ is a bank} \rightarrow Q(y)
\]

\[
\text{TP} \\
\text{TP} \\
\lambda \text{a guard stands in front of } \lambda x . x = 2 \land x \text{ is a bank}
\]

\[
\text{TP} \\
\text{TP} \\
\text{a guard} \quad \text{T} \\
\text{pres} \quad \text{V}^0 \quad \text{PP} \\
\text{stand} \quad \text{P}^0 \quad \text{DP} \\
\text{in} \quad \text{N}^0 \\
\text{front} \quad \text{P}^0 \\
\text{of} \quad \text{DP} \\
\text{every} \quad 2 \\
\text{NP} \\
\text{bank}
\]

*The object DP, and everything up to TP*, has a value only if *n* is a bank.
Note that I am using the first idea we had about indices. I’m assuming that they are predicates that hold of entities which are identical to what the index is valued as. We saw from our study of hyd-53as that indices should be given a semantics that allows them to hold of entities that are part of the entity assigned to the index.

\[ \llbracket n \rrbracket = \lambda x. x = g(n) \]

I’ve also suppressed the max operator that makes up the denotation of definite descriptions, and which interacts with the “part of” definition of indices to produce result that are often the same as (13). The cases we look at today don’t require those details, and so I think they can be safely ignored.

Kotek (2014) has argued that wh-movement and QR can chain together. That wh-phrases can QR is shown independently by their ability to license Antecedent Contained Deletion (a point made in Pesetsky 2000).

\[ \Delta = \text{talk to that child} \]

We saw last time that the syntax for examples like these involves QRing the DP associated with the relative clause, so that the relative has a position to be inserted that is not inside its antecedent. The grammaticality of (14) shows that DPs headed by which have this ability as well: they too can QR.

If Wh movement can apply to a wh-phrase that has QRd, then we would have representations like (15) and (16). In both of these parses, the wh-phrase has QRd to a position above the subject and then, from that position wh moved. The graphs differ with respect to how the φ features are deployed in the DP. The second tree represents a language which allows φ features to be within NPs of this kind, and in this case, the φ features are left in the lower position, just as in (10). In such a language, this DP could be pronounced as a resumptive pronoun.

\[ \Delta = \text{talk to that child} \]

An unfortunate consequence of this proposal is that it predicts that resumptive pronouns should be just as common in the trace position of QR as they are in
the trace position of wh-movement. In fact, I don’t know of any convincing examples of resumptive pronouns left by QR. Indeed, I’m not sure that there are convincing examples of resumptive pronouns left by wh-movement in the lowest position. The examples of resumptive pronouns in constituent questions that seem compelling are found in intermediary position. Kandybowicz (2008) is a good example. I’m uncertain how to capture these cases. My verdict is that the system I’ve sketched has the ingredients necessary to explain why violations of Terseness can involve resumptive pronouns when DPs are moved, but it does not provide an account of where these ingredients seem to be deployed.

I will maintain the view, however, that Wh Movement can apply to a QR’d DP. It has some utility in understanding other issues, which I’ll briefly mention today. If the QR+Wh-Movement combination has the syntax in (15) or (16), then we’ll need to understand how the linearization algorithm negotiates the conflicting requirements of QR and wh-movement. For QR, we have described the fact that the moved NP gets spoken in the lower of its two positions with (17).

(17) \( p(every), p(some), \ldots \) must be included in the path of the head of its complement.

For wh-movement we’ve not made an explicit statement of how overt wh-movement is determined. In English, it is only one wh-phrase that needs to be spoken in the higher position, which suggests that it’s a requirement of Q that forces the wh-phrase to be merged into its higher position. Cable (2010) suggests that there is a morphological requirement between Q and which is responsible for overt movement. That fits with our system.

(18) Q must merge with the phrase headed by its exponent (i.e., which).

These two requirements (or their equivalents in the correct analysis) conflict in (15) and (16). (18) wins the contest, and we should understand why. I believe there’s a way of connecting this to the failure of resumptive pronouns in lowest position. I can see a way that this system can capture these two things, but I don’t see much in the way of evidence that helps support that way. If you’re interested in seeing it, ask.

If which-phrases can QR, then we have to rethink what the meaning of which is. In my initial semantics, I made which have the same meaning that the definite determiner the has. And then we designed a theory of definite descriptions that allows them to function as variables. What we see from (14) is that we should instead treat which as a quantifier, and let QR manufacture the trace in the lower position. I think the common intuition is that which is a kind of existential quantifier. In questions, as we saw, the quantification is over Skolem functions – functions that take an entity-type argument are return an entity type element. In our present semantics, that function is assigned to Q, and I will maintain that decision. It has the utility of forcing the existential quantification to scope wide. I will instead make which provide a restrictor for the functions that are being quantified over and feed that restrictor to the existential quantifier in Q position. I will adopt Engdahl’s idea that the restrictor of which is converted from a predicate of individuals to a predicate of Skolem functions and closed by a universal quantifier. So, here is the denotation for which.

\[
\text{[which]} = \lambda P.\lambda W.\lambda p.\lambda f.[\forall y. P(f(y))] = 1 \land W(f)(p) = 1
\]

Key:

- \( x, y, z \) variables over entities
- \( P, Q \) variables over predicates of entities (e.g. \( \lambda x. P(x) \))
- \( f \) variable over skolem function
- \( p, q \) variables over propositions
- \( \forall W \) variable over relations between propositions and functions (e.g. \( \lambda f.\lambda p.\forall W(f)(p) \))

Let’s start with a (relatively) simple example in (20).

(20) (I wonder) which papers no woman should review?

The parts go together in the way indicated in (21) on the following page. In (21), I’ve indicated, for the first time in this class, that subject arguments undergo a movement from their underlying position (Specifier of vP) to Specifier of TP. In this instance, that movement invokes a semantics just like that we’ve given to QR: the moved phrase in the higher position binds a variable in the lower position. I’ve obscured that syntax in (21), I hope without introducing error.
Like all instances of QR, the wh-phrase has moved in a fashion that causes the NP is shared by the determiner in object position and the determiner containing which. Because both of these determiners take that NP as an argument, its denotation will be combined in both of the positions it resides in. This gives us a way of using the late merge derivations we saw in the context of relative clause exposition to create structures like that on the next page. This, then, is a syntax that allows the relative clause to fall outside the scope of to him in (7). This is just the kind of solution that we get from Lebeaux (1990).
(7) I wonder which paper that Mark brought no woman talked to him about.

(22) \[
\lambda p.\exists f. [\forall y \text{paper}(f(y)) = 1 \land \text{Mark\_brought}(f(y)) = 1] \land p = \neg \exists z. \text{woman}(z) \land z \text{ talk to him about } i.x = f(z) \land \text{paper}(x)
\]

\[
\lambda p.\lambda f. [\forall y \text{paper}(f(y)) = 1 \land \text{Mark\_brought}(f(y)) = 1] \land p = \neg \exists z. \text{woman}(z) \land z \text{ talk to him about } i.x = f(z) \land \text{paper}(x)
\]
If QR is the process that allows a relative clause to be interpreted in its higher position, then we should expect to find that things which don’t QR cannot partake in this process. One thing we’ve seen wh-move that shouldn’t be able to QR are APs. The structure for a wh-moving AP looks like (23). (I’ve given the semantics we arrived at originally.)

How rich is every heiress

\[ \lambda p \, \exists f \cdot p = \forall x \text{ heiress}(x) \rightarrow x \text{ is } (id \ d = f(x) \land d \text{ is degree})\text{-rich} \]

There is some evidence that \( \text{Deg}^0 \) – the thing that is pronounced as *how* but has the meaning of *which degree* – is capable of QRing. (see Bhatt and Pancheva (2004).) To convert these structures into one that embraces the idea that wh-phrases QR will mean changing the denotation of \( \text{Deg}^0 \) and QRing it. The semantics is straightforward, but the syntax is a little tricky because it involves pied-piping. I won’t dive into that problem here. What’s relevant is that the \( \text{DegP} \) – the thing often called an AP – doesn’t QR. And that means that a clause associated with this AP cannot late merge.

\( \text{DegPs}/\text{APs} \) don’t get modified by relative clauses, so we cannot test this prediction with relative clauses. But everything we’ve seen for relative clauses holds for complement clauses as well. They can “extrapose” as in (24).

(24) She recounted the rumor last week that the \( \text{pCC} \) is a phenomenon of human languages.

And they too can contain names that don’t produce disjoint reference effects in wh-movement contexts.

(25) Which rumor that John is responsible for the fiasco does he hope you will stop?

\( \text{DegPs}/\text{APs} \) do host complement clauses, so we can test the predication with them. Compare (25) to (26).

(26) How happy that John is responsible for the fiasco does he think you are?

My informants reliably find a contrast in these examples. It is harder to get an interpretation that allows John and he to corefer in (26) than in (25).

This contrast is discussed in Heycock (1995), Huang (1993), Takano (1995) and many others.

Now we have an interesting problem. By letting wh-phrases be quantifiers of a sort that undergo QR, we have allowed relative clauses to late merge and this gives us a handle on why relative clauses need not be interpreted in the scope of the lower position that the wh-phrase occupies. But now we have lost our simple explanation for why a pronoun in a wh-phrase can be bound as if it were in the lower position. We need to do this, recall, so that we can explain why the contrast in (27) is like the contrast in (28).
(27)  a. * No woman’s father should forget pictures of herself.
    b. No woman should forget pictures of herself.

(28)  a. * Which pictures of herself should no woman’s father forget?
    b. Which pictures of herself should no woman forget?

We concluded that the *pictures of herself* material in (28) must be in the lowest position, and in that position, it is subject to the same requirements that cause *no woman* to be a possible antecedent only in (27b).

On the other hand, we have also seen that when an anaphor is in the NP part of a quantifier, it must be bound by virtue of both its positions: the lower and the higher. That is why, recall, (29a) allows for the universal quantifier to outscope the indefinite, but (29b) doesn’t.

(29)  a. A curator restored every painting of Knossos.
    b. A curator restored every painting of herself.

Our conclusion was that the presence of *herself* forced the NP to remain within the scope of *a curator* and this must bring with it the consequence that *every* also has to be within the scope of *a curator*. For wh-phrases, however, we want the existential quantifier that comes with them to take scope over the antecedent for the reflexive they contain. I’ve put that existential quantifier in Q, but the semantics require the wh-phrase to take scope just below Q. If that’s right, what we are seeing is that QR of wh-phrases is capable of bringing a reflexive out of the scope of its antecedent, but QR of other DPs doesn’t. What makes wh-phrases special?

I have a speculation which I’ll show you. First, though, we should see what it is precisely that is responsible for preventing QR from bringing a reflexive out of the scope of its antecedent. Our syntax for the inverse scope of (29) is (30).

The Binding Theory requires *herself* to have the same index as the subject of *restore*. That will cause it to be bound by that subject when we consider how the denotation of TP* is computed. But its denotation will also be part of QP, and here it is not bound: it will be a free variable. Free variables are typically interpreted by a context – they are given a value that allows them to refer to a salient, or accommodated, antecedent. One way we could block this structure is to hypothesize that an index cannot get both of these interpretations.

(31)  Every occurrence of an index, *n*, in XP must be bound in [XP] or every occurrence of an index, *n*, in XP must be free in [XP].

We might also speculate that what blocks (30) is the Binding Theory:

(32)  If α is a reflexive, then for every p(α), there must be some β ∈ p whose sister binds α.
(32) merely requires of a reflexive that it be bound in every path that contains it. 
(32), however, doesn’t extend to pronouns, as in (33).

(33) A curator_2 restored every painting of hers_2

I tentatively adopt (31). It’s because of (31), then, that (29a) can get a representation parallel to (30), but (29b) can only get the representation in (34).

(34) \( \exists y \text{ curator}(y) \land \forall z \text{ painting_of_herself}_3(z) \rightarrow \\
y \text{ restore } i x. x = z \land \text{ painting_of_herself}_3(x) \)

This is what is responsible for the fact that QR cannot bring a reflexive out of the scope of its antecedent.

Okay, with this in hand, let’s look at what our account does for (7).

(7) Which papers of hers_4 that Mark brought will no woman_4 talk to him about?

If we plug in all the parts, we’ll get the expected violation of (31).
(35)  
\[ \lambda p. \exists f. [\forall y. \text{paper_of_hers}_4(f(y)) = 1 \land \text{Mark_brought}(f(y)) = 1] \land p = \neg \exists z. \text{woman}(z) \land z \text{ talk to him about } \iota x. x = f(z) \land \text{paper_of_z}(x) \]

\[ \lambda p. \lambda f. [\forall y. \text{paper_of_hers}_4(f(y)) = 1 \land \text{Mark_brought}(f(y)) = 1] \land p = \neg \exists z. \text{woman}(z) \land z \text{ talk to him about } \iota x. x = f(z) \land \text{paper_of_z}(x) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]

\[ \lambda W. \lambda p. \exists f. W(f)(p) \]
The index 4 on hers is both bound (by no woman) and free (in the restrictor of which) in the denotation assigned to the root CP. How is this ungrammatical outcome avoided by wh-phrases?

A difference between wh-determiners and other quantificational determiners is what the quantify over. Determiners like every relate predicates of entities: they say that if some entity is a thing described by their restrictor it will be a thing described by their nuclear scope. But questions, we have seen, quantify over functions. They say that there exists some function that picks out things which their restrictor describes, and they relate that function to the propositions that characterize the answer space. Because the restrictor for which is a predicate of entities, what has to happen is that which has to convert this predicate into something that will describe Skolem functions. Engdahl’s proposal – incorporated in our semantics – is to “close” one of the arguments of that Skolem function with a universal quantifier.

\[
\lambda W. \lambda p. \lambda f. \left[ \forall y \text{papers}(f(y)) \right] \wedge W(f)(p)
\]

What which has done is take \( \lambda x. \text{papers}(s)=1 \) and say that we’re talking about Skolem functions that spit out an entity that satisfies the predicate for every entity that the function is fed. This is a way of making the restrictor blind to the input to the Skolem function; the functions are not restricted by the kinds of entities that the Skolem function applies to, only to the kinds of entities it returns. What the input to the Skolem function is will be determined by the nuclear scope. This is exactly what we want to happen to hers.

Imagine, then, that what which does to the predicate it combines with is close all of the free variables in its restrictor with a universal quantifier. This is, indeed, the solution that Engdahl suggests for our problem. This will produce (37).
This satisfies (31). If *her* were replaced by *herself* it would still satisfy (31), and, in addition, it would satisfy the requirement that *herself* has that its antecedent c-command it from a non-QRd position. It satisfies that requirement by virtue of *herself* being interpreted in the lower position as well as the higher one.

The class can now end.

### References


