The deployment of copies, aka movement, is responsible for disparities in where material is pronounced and where it is semantically interpreted. I will look at a few puzzles concerning how copies are placed, and suggest tentative solutions or advertise solutions others have proffered. To get to the puzzles requires building up some background, so that is where I shall start.

1 First Case: Speak Low/Interpret High

The classic examples of this are instances of Quantifier Raising (QR)\(^1\) and wh-in-situ\(^2\). I’ll focus on the QR cases in this talk. QR has several interesting features. It allows a quantifier to be interpreted in a position higher than where it is pronounced, as in (1), and it also allows a relative clause associated with a quantifier to be interpreted higher than where it is pronounced, as in (2).

\[
\begin{align*}
(1) & \quad \text{A syntactician } [\text{vp visited every philosopher}]. \\
& \quad = \text{every philosopher} > \text{a syntactician}
\end{align*}
\]

\[
\begin{align*}
(2) & \quad \text{A syntactician } [\text{vp visited every philosopher that a phonologist did } \Delta]. \\
& \quad \Delta = \text{visited } x
\end{align*}
\]

We know that the relative clause in (2) is interpreted outside of the vp it stands in because that is required to resolve the ellipsis it contains.

QR does not allow the NP associated with a quantifier to be interpreted higher than where it is pronounced, however. This can be seen by observing that material within that NP triggers Principle C effects with respect to the position it is spoken in.

\[
\begin{align*}
(3) & \quad * \text{She}_1 [\text{vp visited every friend of Mary’}_1\text{s}_1].
\end{align*}
\]

I’ll adapt an account of this paradigm found in Fox and Nissenbaum (1999) and Fox (2002). The account employs the bottom-up derivations proposed in Chomsky (1995a), the copy theory of movement\(^3\) and the single-output view of the syntax/semantics interface.\(^4\) For (1), QR produces a representation like (4), where the shaded material is unpronounced.

---

\(^1\) See May (1977, 1985).
\(^2\) See Chomsky (1977) and Huang (1982).
\(^3\) See Chomsky (1995b).
This representation treats referential indices as features that are brought into the derivation by determiners and are projected onto the phrases those determiners head. Indices play an important role in interpreting these structures. Let’s start with the rules in (5); (5b) is from Fox (2003).

(5) a. Let \{n, n', \ldots, n^m\} be indices on the same projection line. Only one n needs to be semantically interpreted.

b. **Trace Conversion**
   - For \(\phi'\), interpret \(\phi\) as a function that maps an individual, \(x\), to the meaning of \(\phi[x/n]\).

   \(\phi[x/n]\) is the result of replacing the head of every constituent with the index \(n\) in \(\phi\) with the head \(the_x\), whose interpretation, \([the_x]\), is \(\lambda y. ([the] (P \cap \lambda y. y = x)).\)

Fox sees (5b) as a version of Heim and Kratzer (1998)’s method of interpreting indices as lambda converters. It lets the index on a phrase turn that phrase into a binder, in the normal way, and makes the index on a determiner convert that determiner into a kind of restricted bound variable. In a simple example like (6), for example, it would deliver an interpretation that could be paraphrased with (7).

---

(6)  

```
(6) every2 NP friend of Mary’s
```

(7) every x: friend-of-Mary’s(x) is such that she visited the y: friend-of-Mary’s(y) & y = x.

Applied to (4), (5b) will deliver an interpretation in which a is interpreted within the scope of every, but at the same time it will preserve the Principle C effect in (6).

Fox’s Trace Conversion rule bothers many. It has the nettlesome property of letting a whole class of lexical items — determiners — be ambiguous. And it resolves that ambiguity by syntactic rule, something not seen elsewhere. So let me offer a variant of Fox’s proposal that avoids these consequences before turning to explain the rest of the paradigm in (1)–(3).

This variant builds on ideas many have had about the syntax of quantification.6 Perhaps it’s closest to ideas in Beghelli (1993, 1995), Sportiche (2003), Butler (2004) and Adger and Ramchand (2005). Imagine, as in Matthewson (2001), that Quantificational expressions make use of two functional heads. One has the denotation of quantifiers,7 and the other is a choice function that provides the domain for the quantification. For concreteness, we can assume this choice function to have just the meaning given to determiners by Fox’s Trace Conversion rule. Unlike Matthewson, but like those cited above, let’s separate these two functional heads, putting the term that expresses the quantification in the position where its scope is computed, while the choice function is in construction with the NP. The morphological form of the choice function varies depending on the quantificational term. Let’s follow Adger and Ramchand (2005) and let this dependency be mediated by agree. agree will determine the morphological form of the choice function part, and make both heads share an index. This proposal would give to (3) the representation in (8) before the object qrs.


7 In her syntax, quantifiers combine directly with the DP headed by the choice function, and so she adjusts the denotation of quantifiers to be of semantic type <e,<<e,t>,t>>. My syntax will differ, allowing them to be of the more traditional <<<e,t>,<<e,t>,t>>, type.
This variant of Fox’s proposal preserves his syntax and semantics, but does not require a syntactic rule that rewrites the semantics of lexical items. It also straightforwardly derives (10).

(10) The restrictor for a quantifier must be interpreted within the scope of that quantifier.

Other semantic treatments are possible with only slight differences in the syntax. What I will say here fits a family of accounts, then.

Back to the paradigm in (1)–(3). The account of (1) vs. (3) requires that a spoken copy be interpreted semantically and this, in turn, requires that the relative clause holding the ellipsis in (2) not be pronounced within the VP that serves as the antecedent for the ellipsis. Fox and Nissenbaum (1999) argues that this is correct, producing an analysis that builds on Baltin (1987)’s proposal that extraposition produces the surface syntax that licenses antecedent contained deletions. There is independent support for this conjecture from alternations like those in (11) (originally from Tiedeman (1995)).
(11) a. * I said that everyone you did ∆ arrived.
    b. I said that everyone arrived that you did ∆.

    (Fox 2002, (35b), (36b): 77)

The relative clause in (2) will have string-vacuously extraposed. Fox and Nissenbaum argue that relative clause extraposition arises by merging the relative clause to the unpronounced copy of the DP it modifies. Thus, (2) will have a representation like (12).  

Our semantics gives this representation the right interpretation and correctly resolves the ellipsis. Interesting confirmation that this account is on the right track comes from the alternation in (13), discovered by Fiengo and May (1994).

(13) a. * You introduced him₁ to every friend of John’s₁ that I wanted you to ∆.
    b. * You introduced him₁ to everyone that John₁ wanted to you to introduce him to.
    c. You introduced him₁ to everyone that John₁ wanted you to ∆.
    d. ?? You introduced him₁ to everyone yesterday that John₁ wanted you to introduce him to.

In (13a) we see the familiar Principle C effect for material within the NP, and in (13b) we see that it holds for names within the relative clause as well. In (13c,d) we find that the Principle C effect for material within the relative clause is lifted when that relative clause has an ellipsis to resolve, and that it is weakened when that relative clause extraposes. I don’t know why there is a difference between (13c) and (13d), but that they should both fall on the side of grammaticality is expected on Fox’s proposal. They will get a representation like (14), which does not produce a Principle C effect with him and John as they are not in a c-command relation.  

\[ 8 \text{ I’ve not represented the functional head that binds a.} \]

\[ 9 \text{ This VP is artificially flat as a favor to the trees.} \]
But as the ungrammaticality of (13b) highlights, we must find a way of preventing string-vacuous extraposition of the relative clause unless an ellipsis requires it. We need to find something that generally prevents string-vacuous extraposition, and then something about ellipsis that overrides this prohibition. To prevent extraposition, I suggest that there is a transderivational constraint which favors the quickest derivation path to a given spell-out. This is (15).

(15) **Spell Out Soon**

Let $D = \{ D_1, D_2, \ldots, D_n \}$ be the well-formed derivations from a fixed Numeration, $N$, to a fixed Spell Out, $S$, and let $|D|$ be the number of applications of operations in $D$. All $D \in D$ are ungrammatical if there is a $D' \in D$ such that $|D'| < |D|$. 

What makes ellipsis special, I conjecture, is that it invokes an unusual requirement on Spell Out. To know whether a $\text{vp}$ can be Spelt Out as silence, the denotation of that $\text{vp}$ must be computed and matched with an antecedent. Think of ellipsis as a Spell Out instruction that arises when a particular semantic condition is met — this is a way of conceiving of ellipsis that can be found in Merchant (2001). We might do this by letting the syntax determine the placement of a feature on certain phrases, let’s call this an “e-feature.” An e-feature has a Spell Out interpretation along the lines of (16).

(16) **Spell Out( XP$_e$ ) = silence and is defined only if [XP] has an ellipsis antecedent.**

An e-feature on a phrase prevents Spell Out from interpreting that phrase in its “normal” way, and instead makes Spell Out render it as silence. It also makes Spell Out a partial function, requiring that XP have an antecedent of the sort that is appropriate for elided phrases. This will require Spell Out to have the ellipsis resolved, and this in turn will require a derivation that puts the phrase bearing an e-feature outside of its antecedent.

This is the background theory I will assume. In addition to the semantics I have sketched, the sos, and the interpretation of ellipsis in (16), it requires the following.

---

10 See Pesetsky (1989) "earliness principle."
(17) **Things Needed**

a. A mechanism for producing copies of terms in syntactic representations. This could be achieved by letting merge fashion identical phrases that share an index, as in Chomsky (1995b), or by letting movement be understood as giving a single phrase more than one syntactic position, as in Engdahl (1986).

b. A principle that allows some copies to be ignored by the phonological interpretation of sentences. This requires a principle that forces certain copies to be silent. Perhaps a combination of the sos with a phase-based timing of Spell Out, such as is proposed in Nissenbaum (2000), will do the trick.

c. A principle that forces all copies to be semantically interpreted.

I will not have much to say about how these needs are satisfied. What I will try to show is that not all instances of movement can satisfy (17).

2 Second Case: Speak High/Interpret Low

The system I’ve adopted works not only for all cases where a phrase is interpreted higher than where it is pronounced, but it also works for some cases where a phrase is interpreted lower than where it is pronounced. The examples in (18), for instance, get the correct interpretations from the representations indicated.\(^{11}\)

(18) a. A student of hers seemed to Jane to be in the room.

\(^{11}\) I take shortcuts whenever it doesn't matter and reduces arboreal girth.
b. Which picture of Jane's did she sell?

In (18a), a student of hers is interpreted outside the infinitival clause it originates in, and binds a restricted variable within that infinitival clause. In (18b), which picture of Jane's has wide scope, but the restriction on its variable provided by the unspoken copy induces a Principle C effect between her and Jane. These work just as expected.

But there are many cases where these mechanisms fail to deliver the right interpretations. Here are a few.\(^\text{12}\)

\((19)\)  
\begin{align*}
a. & \text{Spätzle isst Thilo regelmäßig.} \\
& \text{spätzle eats Thilo regularly} \\
& \text{‘Thilo eats gobs of spaetzle.’} \\
b. & \text{A student of hers seemed to every professor to be in the room.} \\
c. & \text{A student of Jane's seemed to her to be in the room.} \\
\end{align*}

In (19a), the meaning of isst is not computed at the position it is spoken; instead, it is only semantically interpreted at its underlying position.

There are two interpretations of (19b) in which a student of hers is similarly not interpreted in its spoken position. On both of these interpretations, her is bound by every professor and the existential quantifier associated with a is interpreted within the scope of the universal quantifier associated with every. If the student-DP is interpreted outside the infinitival complement, for instance, what we require from our present system is a representation like (20).\(^\text{13}\)

\(^{12}\) The cases in (19b) and (19c) are discussed in Fox (2002) and Hornstein (1995).

\(^{13}\) I've assumed that the index ‘t’ that every brings with it has projected up to the PP to every professor.
If her is bound by every professor, then the present system requires a representation in which, like (20), there is no copy in the position we speak one.

Similarly, (19c) has an interpretation in which the student-DP is interpreted within the infinitival complement: its world variable is bound by seem. On this interpretation Jane and her invoke a Principle C effect, and the existential quantification associated with a has narrow scope. Our present system requires a syntactic representation like (21) for such a meaning.

As with (20), this representation does not put a copy in the position where one is spoken.

These are instances of “total reconstruction.” They flout (17c). We need to explain:

(22) If there are two or more copies of α, then the pronounced copy must be semantically interpreted unless it is the highest one.

---

14 One issue I will dodge in this talk is why Principle C effects are not invoked in all cases of Argument Movement. On the wide scope interpretation of (19c), for instance, Jane and her may easily corefer. I think Takahashi (2006)’s proposal — that the NP of a DP may merge late — provides a plausible account.

15 See May (1977) and Sauerland and Elbourne (2002).
The traditional account for (22) is to let movement create representations that are not semantically interpreted. The spoken copies of (19) could be manufactured after the representations indicated above have gotten their final semantic interpretation. This approach claims that a spoken copy will either be completely interpreted or completely ignored by the semantics. What we find instead is that parts of a spoken copy can be ignored while other parts can be interpreted. This is pied-piping. We need to allow the sentences in (23) to have the representations indicated.\(^\text{16}\)

(23)  

\[
\begin{align*}
\text{a. Into how many allegations is she supposed to look?}
\end{align*}
\]

b. How many linguists’ mistakes shouldn’t we repeat?

The *many* part of the moved phrases are interpreted outside the scope of *supposed* and *shouldn’t* in these questions, and yet the material that *how many* has pied-piped is interpreted in its underlying position. We need another explanation for (22).

I suggest that these instances of semantically vacuous “movement” are instead cases where the normal principles of linearizing a phrase marker are overrun by requirements of Spell Out. I’ll sketch now a way to do that.

The relevant Spell Out requirements.\(^\text{17}\)

\begin{align*}
(24) & \quad \text{a. A } C^0 \text{ with a } [Q]\text{ feature must be right adjacent to the string whose } [Q]\text{ feature it values.} \\
& \quad \text{b. An } I^0 \text{ with a } [+\text{case}]\text{ feature must be right adjacent to the string whose } [+\text{case}]\text{ feature it values.}\quad \text{\footnote{I ignore here those conditions relevant for forcing head movement.}}
\end{align*}

These are intended merely to code the fact that the Specifiers of CP and IP have the phrases in them that \(C^0\) and \(I^0\) agree with. Understand these to be well-formedness conditions on Spell Outs.

For the “normal” linearization of a phrase marker, I’ll adopt Kayne (1994)’s Linear Correspondence Axiom, a slightly modified version of which is in (25).

\begin{align*}
(25) \text{ Linear Correspondence Axiom (lca)} \\
& \quad \text{a. If } \gamma \text{ asymmetrically c-commands } \rho, \text{ then } [\gamma, \rho]. \\
& \quad \text{b. } \text{lca}(\{A, B\}) = \{ \langle \alpha, \beta \rangle : \alpha \text{ is dominated by } A \text{ and } \beta \text{ is dominated by } B \}.
\end{align*}

\footnote{This is a better description of French than it is of English, as English lets adverbs intervene between the surface position of the subject and where we think \(I^0\) is. The right account will make sense of this language variation. What matters here is only that the EPP is a Spell Out condition.}

\footnote{I ignore here those conditions relevant for forcing head movement.}
The **LCA** produces a set of ordered pairs whose transitive closure is a string which includes all of the terminals in some phrase marker. It translates “asymmetric c-command” into precedence. It will apply to the simple νP in (26), for example, to produce the ordering in (28). (I’ve boxed the maximal and minimal projections that enter into an asymmetric c-command relation.)

(26)

```
\[\text{DP} \quad \check{\text{vP}} \quad \text{DP} \quad \text{vP} \quad \text{vP} \quad \text{V} \quad \text{VP} \quad \text{V} \quad \text{VP} \quad \text{eat} \quad \text{DP} \quad \text{the} \quad \text{NP} \quad \text{poi}\]
```

(27) asymmetric c-commanders in (26):

```
\[
\begin{cases}
[\text{DP}_{\text{she}}, \text{v}], [\text{DP}_{\text{she}}, \text{VP}], [\text{DP}_{\text{she}}, \text{eat}], [\text{DP}_{\text{she}}, \text{the}], [\text{DP}_{\text{she}}, \text{NP}], [\text{DP}_{\text{she}}, \text{poi}]
\end{cases}
\]
```


```
\[
\begin{cases}
[\text{v}, \text{eat}], [\text{v}, \text{DP}_{\text{the}}], [\text{v}, \text{NP}], [\text{v}, \text{the}], [\text{v}, \text{poi}]
\end{cases}
\]
```

```
\[
\begin{cases}
[\text{eat}, \text{the}], [\text{eat}, \text{NP}], [\text{eat}, \text{poi}]
\text{[the, poi]}
\end{cases}
\]
```

(28) LIN(vP):

```
\[
\begin{cases}
\langle\text{she}, \text{v}\rangle, \langle\text{she}, \text{eat}\rangle, \langle\text{eat}, \text{the}\rangle, \langle\text{the}, \text{poi}\rangle
\end{cases}
\]
```

```
\[
\begin{cases}
\langle\text{v}, \text{eat}\rangle, \langle\text{she}, \text{the}\rangle, \langle\text{eat}, \text{poi}\rangle
\end{cases}
\]
```

```
\[
\begin{cases}
\langle\text{v}, \text{the}\rangle, \langle\text{she}, \text{poi}\rangle
\end{cases}
\]
```

```
\[
\langle\text{v}, \text{poi}\rangle
\]
```

The transitive closure of (28) is \textit{she }\textit{v eat the poi}, as desired.

When copies are involved, things get more complex. If we assume that copies can neither precede nor follow themselves (\textsc{=(29)}),\(^{19}\) then the **LCA** cannot both be complete and consistent in a phrase marker that has copies.\(^{20}\)

(29) \textsc{*(x, y)}\), where \textit{x} and \textit{y} are copies.

For example, consider how the **LCA** would evaluate the full sentence that (26) fits into:

---

\(^{19}\) This is easiest to derive on the view that copies are one phrase with several syntactic positions.

Because $\text{DP}_{\text{she}}$ asymmetrically c-commands $\text{she}_1$, a complete linearization of this phrase marker will include $\langle \text{she}, \text{she} \rangle$, a violation of (29). Let’s understand linearizations, then, to be as complete as possible. For any given phrase marker, let the LCA produce a set of linearizations that are as complete as (29) will allow them to be.

(31) Form all subsets, $\alpha$, of the set generated by the LCA such that:

a. $\alpha$ makes a consistent linearization (e.g., obeys (29)), and  
b. $\alpha$ orders a copy of every terminal.

We can let the well-formedness conditions on Spell Out choose from these linearizations the correct one. So, for instance, the two consistent, complete linearizations of (30) are (32).

(32) a.  

\[
\begin{align*}
\langle \text{she}, \text{can’t} \rangle & \quad \langle \text{she}, \nu \rangle & \quad \langle \text{she}, \text{eat} \rangle & \quad \langle \text{eat}, \text{the} \rangle & \quad \langle \text{the}, \text{poi} \rangle \\
\langle \text{can’t}, \nu \rangle & \quad \langle \nu, \text{eat} \rangle & \quad \langle \text{she}, \text{the} \rangle & \quad \langle \text{eat}, \text{poi} \rangle & \\
\langle \text{can’t}, \text{eat} \rangle & \quad \langle \nu, \text{the} \rangle & \quad \langle \text{she}, \text{poi} \rangle & \\
\langle \text{can’t}, \text{the} \rangle & \quad \langle \nu, \text{poi} \rangle & \\
\langle \text{can’t}, \text{poi} \rangle & 
\end{align*}
\]

$= \text{she can’t } \nu \text{ eat the poi}$

b.  

\[
\begin{align*}
\langle \text{can’t}, \text{she} \rangle & \quad \langle \text{she}, \nu \rangle & \quad \langle \text{she}, \text{eat} \rangle & \quad \langle \text{eat}, \text{the} \rangle & \quad \langle \text{the}, \text{poi} \rangle \\
\langle \text{can’t}, \nu \rangle & \quad \langle \nu, \text{eat} \rangle & \quad \langle \text{she}, \text{the} \rangle & \quad \langle \text{eat}, \text{poi} \rangle & \\
\langle \text{can’t}, \text{eat} \rangle & \quad \langle \nu, \text{the} \rangle & \quad \langle \text{she}, \text{poi} \rangle & \\
\langle \text{can’t}, \text{the} \rangle & \quad \langle \nu, \text{poi} \rangle & \\
\langle \text{can’t}, \text{poi} \rangle & 
\end{align*}
\]

$= \text{can’t she } \nu \text{ eat the poi}$

Only (32a) meets the Spell Out requirement in (24b), however, and so it’s chosen.

This, then, is the normal linearization procedure. For the cases of semantically vacuous movement that we’re interested in, I suggest there is a rule that adds linearization statements.

(33) $\text{move} =_{\text{def.}} \text{add } \langle \alpha, \beta \rangle$, $\alpha$, $\beta$ terminals.

Let’s look at how this will work for a simplified version of (23a).
(34) Into which allegations is she looking?

Let this be the representation that is semantically interpreted. An LCA-faithful linearization of it is (35).

\[
\begin{align*}
\langle \text{which, allegations} \rangle & \langle \text{allegations, is} \rangle \langle \text{is, she} \rangle \langle \text{she, looking} \rangle \langle \text{looking, into} \rangle \\
\langle \text{which, is} \rangle & \langle \text{allegations, she} \rangle \langle \text{is, looking} \rangle \langle \text{she, into} \rangle \\
\langle \text{which, she} \rangle & \langle \text{allegations, looking} \rangle \langle \text{is, into} \rangle \\
\langle \text{which, looking} \rangle & \langle \text{allegations, into} \rangle \\
\langle \text{which, into} \rangle &
\end{align*}
\]

= which allegations is she looking into

This is formed by taking the subset of the ordered pairs that the LCA produces for (34) which omits the shaded terminals. This is a perfectly good linearization, but it does not satisfy (24a), repeated here.

(24a) A C\(^0\) with a [Q] feature must be right adjacent to the string whose [Q] feature it values.

In (34), the [Q] feature which C\(^0\) values is on the pp headed by into. What (24a) requires, then, is that the C\(^0\) be right adjacent to the string in (36).

\[
\begin{align*}
\{ \langle \text{into, which} \rangle \langle \text{which, allegations} \rangle \} \\
\{ \langle \text{into, allegations} \rangle \}
\end{align*}
\]

= into which allegations

To satisfy (24a), the statements \langle into, which \rangle and \langle into, allegations \rangle need to be added to the linearization in (35). Adding those statements and leaving everything else in (35) will result in an inconsistent linearization. For instance, it will create a linearization that contains (37).
Further, there is no subset of $(35)+\langle \text{into, which} \rangle + \langle \text{into, allegations} \rangle$ that orders all copies of the terminals and is consistent. Movement, then, must do more than add the statements that provide satisfaction of $(24a)$. It must add statements that do this and achieve a complete linearization. Movement will need to add the statements in (38).

(38) Add: $\{ \langle \text{into, which} \rangle \langle \text{into, allegations} \rangle \langle \text{into, is} \rangle \langle \text{into, she} \rangle \langle \text{into, looking} \rangle \}$

Adding (38) to (35), and then taking a subset which orders all the terminals, is consistent and satisfies $(24a)$ gives us:

(39)

$$\{ \langle \text{which, allegations} \rangle \langle \text{allegations, is} \rangle \langle \text{is, she} \rangle \langle \text{she, looking} \rangle \langle \text{into, looking} \rangle \}
\{ \langle \text{which, is} \rangle \langle \text{allegations, she} \rangle \langle \text{is, looking} \rangle \langle \text{into, she} \rangle \}
\{ \langle \text{which, she} \rangle \langle \text{allegations, looking} \rangle \langle \text{into, is} \rangle \}
\{ \langle \text{which, looking} \rangle \langle \text{into, allegations} \rangle \}
\{ \text{into, which} \}$$

$$= \text{into which allegations is she looking}$$

3 Conclusion

Semantically vacuous movement is “Add $\langle \alpha, \beta \rangle$.” We are not free to add linearization statements at will, however, as the sos $(=(15))$ insists that we produce a Spell Out that is as faithful to the syntactic representation as possible. Thus, we should expect to find semantically vacuous movement only in those situations where there are Spell Out requirements at odds with the semantics.

What’s left to understand is why the movement produced by “Add $\langle \alpha, \beta \rangle$” obeys all of the constraints obeyed by producing copies.

References


