A Remerge Theory of Movement

Kyle Johnson
University of Massachusetts at Amherst
kbj@linguist.umass.edu

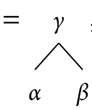
SUNY, Stony Brook
3 September 2010

We need a better theory of movement. The present theories harbor stipulations and give little traction on understanding why movement has the properties it does.

A presently popular theory of movement has the following ingredients.

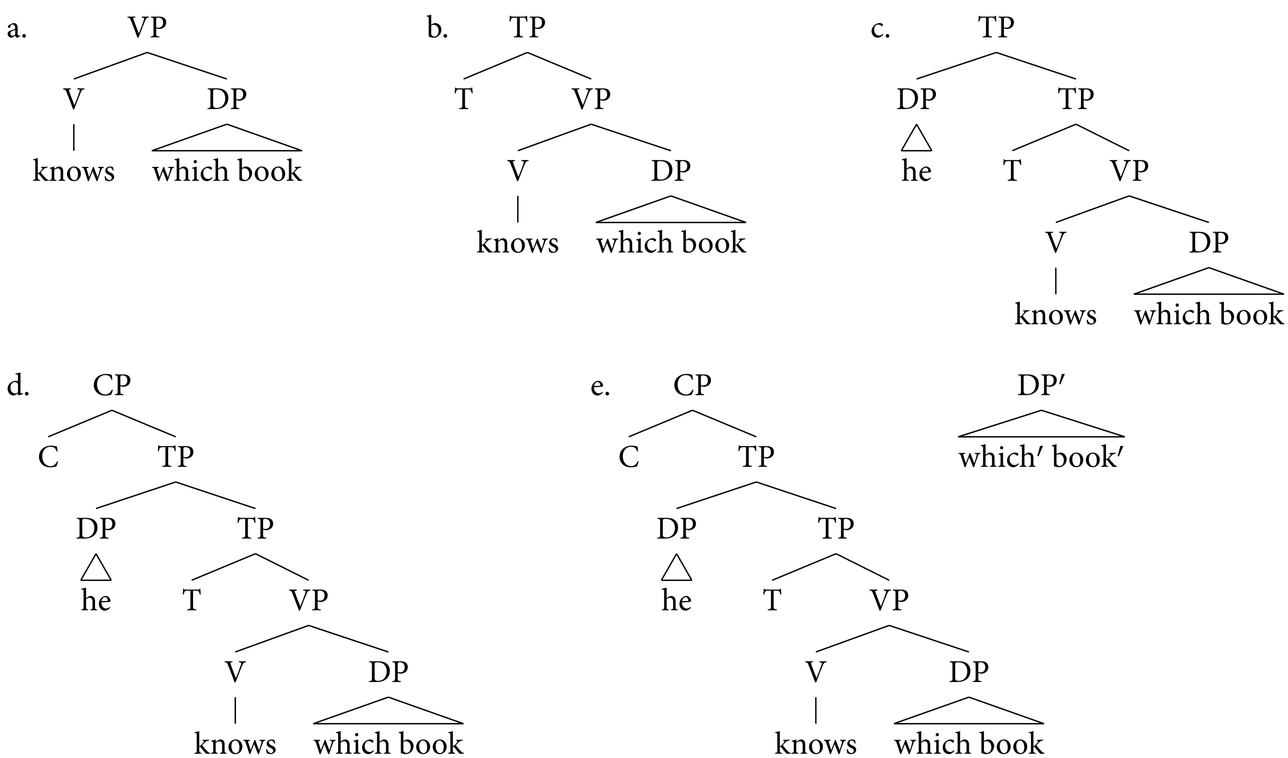
I. The Syntax

a. Copy $\alpha = \alpha'$, an exact copy of α.

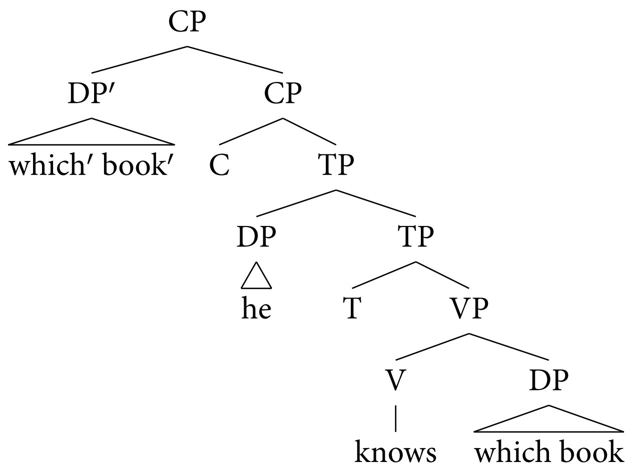
b. Merge(α,β) =  , linear order undetermined

c. Move α to β is Copy α + Merge(α',β)

(1) *illustration:* (She asked) which book he knows.



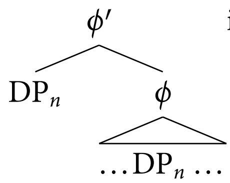
f.



II. The Semantics

(2) Trace Conversion

In

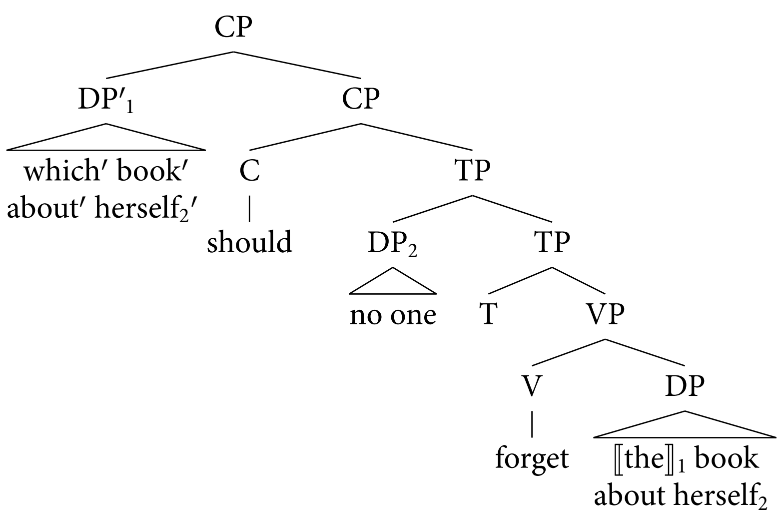


interpret ϕ as a function that maps, x, to the meaning of ϕ[x/n].

ϕ[x/n] is the result of replacing the head of every constituent bearing the index n in ϕ with the head *the*$_x$, whose interpretation, $[\![\text{the}_x]\!]$, is: $\lambda P : P(x).x$.

(adapted from Fox 2003, (52): 111)

(3)



See also Engdahl (1980, 1986) and Elbourne (2005) who suggest that the variable position has the same semantics as that found for "donkey" descriptions:

a. Every man who owns a donkey feeds the donkey.

b. Every man who owns a donkey feeds it.

III. The Phonology

Nunes (2004) argues that only one of the two copies is pronounced because of how the linearization algorithm interprets these representations.

(4) Nunes's system

a. Kayne's (1994) linearization algorithm.

b. Constraints on a Linearization

A well-formed linearization of a phrase marker p must satisfy:

i. *Totality*

The linearization algorithm applies to all vocabulary items in p.

ii. *Antisymmetry*

For all vocabulary items, a and b in p, the linearization of p cannot include both $a < b$ and $b < a$.

iii. *Transitivity*

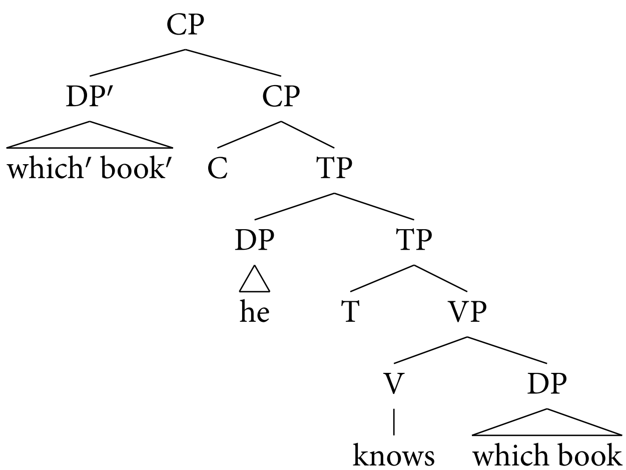
For all vocabulary items, a, b, c in p, if the linearization of p includes $a < b$ and $b < c$ then it must include $a < c$.

c. α and its copy, α', are indistinguishable for the purposes of the constraints.

d. Chain Reduction

Delete α, α is part of a copy.

(5) *illustration:* (She asked) which book he knows.



(6) Linearization of (5) is:

which′ < book′	book′ < C	C < he	he < T	T < knows	knows < which	which < book
which′ < C	book′ < he	C < T	he < knows	T < which	knows < book	
which′ < he	book′ < T	C < knows	he < which	T < book		
which′ < T	book′ < knows	C < which	he < book			
which′ < knows	book′ < which	C < book				
which′ < which	book′ < book					
which′ < book						

This violates Antisymmetry under (4c). Chain Reduction applies to produce either one of:

a. which book she knows

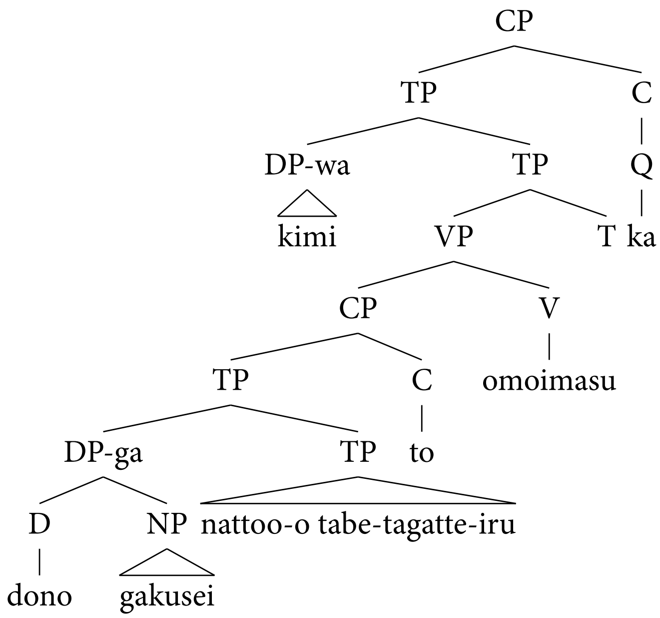
b. she knows which book

English chooses the first.

The picture I'd like to replace these with is one that says that wh-movement involves putting together a definite description of the sort that we see in donkey anaphora, with a Q morpheme that produces the question and binds the variable in the definite description. The model for English questions, then, are languages like Japanese.

(7) (Kimi-wa) dono-gakusei-ga nattoo-o tabe-tagatte-iru-to omoimasu-ka?
(you-top) which-student-nom natto-acc eat-desirous-be-C think-Q
(Which student do you think wants to eat natto?)

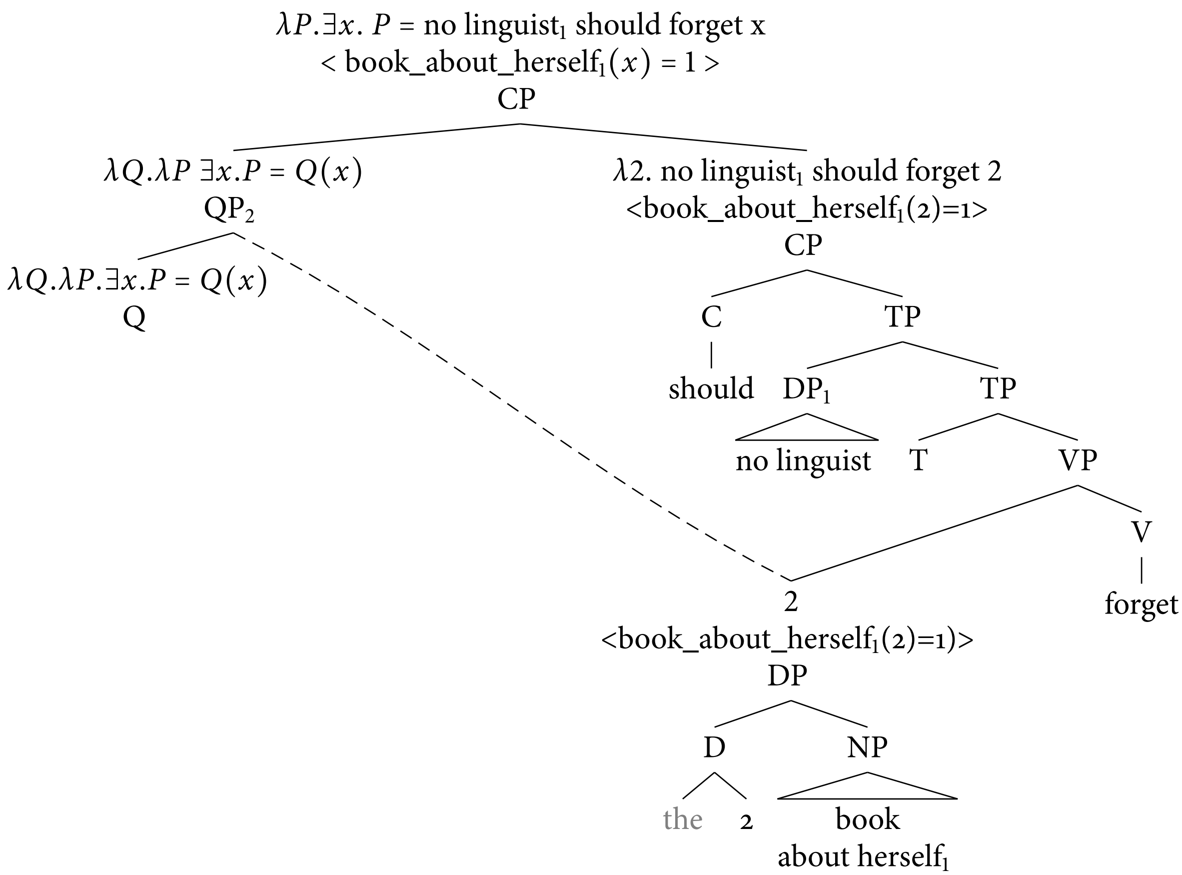
(8)



I want to suggest that English questions are built from the same DP and Q.

Rather than use copy operation, I'll adopt 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), Kobele (2006) and de Vries (2007). That idea is that movement gives an expression two positions by re-merging it. Putting these two together, then, I suggest something like (9). (The semantics here gives a Kartunnen-style account of questions, and uses ideas in Chierchia (1992), Rullmann and Beck (1998) and a class handout from Irene Heim.)

(9)



The material enclosed in "< >" is a presupposition.

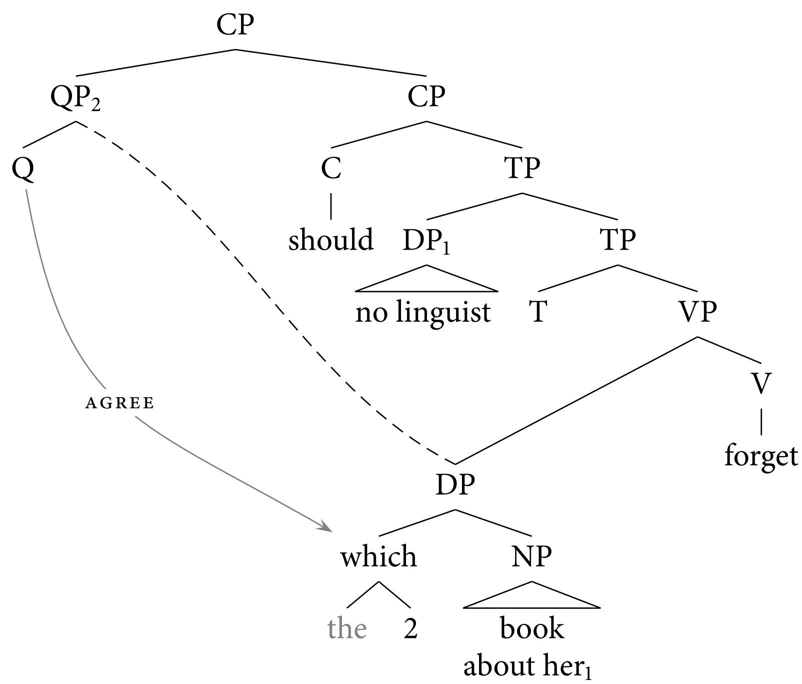
(10) $[\![\text{Q}]\!] = \lambda Q.\lambda P.\exists x.P = Q(x)$

(11) $[\![\text{the}]\!] = \lambda x.\lambda p.x$, when $p(x) = 1$

On this view, technically what has moved is just the DP portion that is interpreted as a variable. This denotation it supplies to the object position of *forget*. That DP has merged with the higher Q, which is the binder of the index within the DP in object position. It has merged with that Q, but its denotation is not computed there. I've indicated that with the dotted line. As a consequence, the QP in the higher position has the same meaning as the Q which heads it.

We need to determine where in (9) the question word *which* is inserted. I am going to assume that it is the D position of the DP that gets matched to the question word. But I want the form this D has to reflect the fact that there is the question morpheme: Q. So I suggest that we let the D get the features responsible for spelling it out as *which* from the Q morpheme under Agreement.

(12)



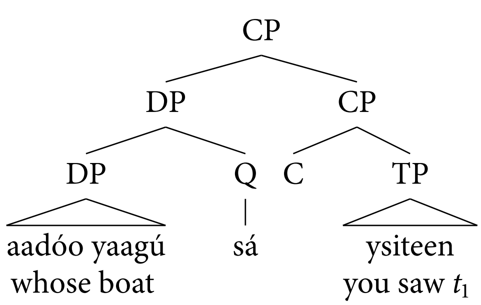
That would make Q parallel to the Tlingit *sa* morpheme that Seth Cable studied in his dissertation.

(13) Aadóo yaagú sá ysiteen?
who boat Q you.saw.it?
'Whose boat did you see'

(Cable 2007, (212)–(213): 155-6)

In (13), the Q particle, *sá*, has merged with a DP, inside of which lies the wh-word: *aadóo*. The whole thing has moved to the left edge of the sentence.

(14)



Moreover, Cable argues that the Q morpheme is in an Agreement relationship with the wh-word, and that there are locality conditions on that agreement relation that determine where the Q particle can be merged. Very roughly, that locality condition can be described with (15).

(15) Q can agree with D only if there is no lexical item that c-commands D but not Q.

The Q morpheme in our system is also in an agreement relationship with the wh-determiner, and so we should expect that, like Tlingit, it will have to merge in a position that does not take it too far from the wh-determiner it is agreeing with. This is what forces Q to merge to a phrase that it does not semantically combine with.

And, indeed, as Cable emphasizes, the range of phrases that the Tlingit Q morpheme can merge with are very close to the range of phrases that Pied-Pipe in wh-questions in English.

(16) a. Aadóo yaagú sá ysiteen?
who boat Q you.saw.it?
'Whose boat did you see'

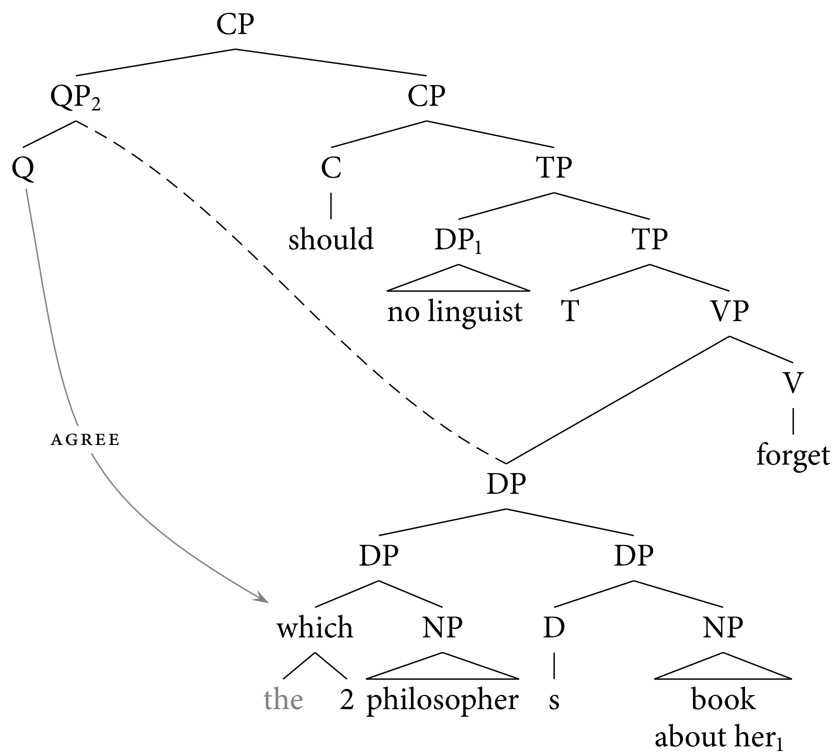
b. Aadóo teen sá yigoot?
who with Q you.went
'With whom did you go?'

c. Daa sá ax éesh aawaxá
what Q my father ate
'What did my father eat?'

d. * Daa aawaxáa sá ax éesh
what ate Q my father
'what ate my father?'

He argues, therefore, that Pied-Piping in English arises because the phrase that moves in English has a Q morpheme merged with it in just the way that Tlingit *sá* does.[1] (17) illustrates.

(17) Which philosopher's book about her should no linguist forget?



This gives us, then, a simplified syntax and removes altogether any special about the semantics of movement. We're left with the phonology.

We can recast Nunes's project, taking advantage of the lack of copies to remove his stipulation that copies are indistinguishable and his specialized deletion rule. I suggest reframing Totality, but otherwise keeping the Kaynean constraints.

[1] See also Cable (2008).

(18) Constraints on a Linearization

A well-formed linearization of a phrase marker p must satisfy:

a. *Totality*
All vocabulary items in p must be in the linearization of p.

b. *Antisymmetry*
For all vocabulary items, a and b in p, the linearization of p cannot include both $a < b$ and $b < a$.

c. *Transitivity*
For all vocabulary items, a, b, c in p, if the linearization of p includes $a < b$ and $b < c$ then it must include $a < c$.

Then I suggest adopting a much freer linearization scheme than Kayne's. Assume that the linearization algorithm, I'll call it *lin*, simply generates every conceivable ordering of vocabulary items — including those that violate the Constraints.

(19) Let L(p) be a set of ordered pairs, $x < y$, where x and y are vocabulary items dominated by p, and "<" means "precedes."
lin(p) $=_{\text{def.}}$ the set consisting of every L(p).

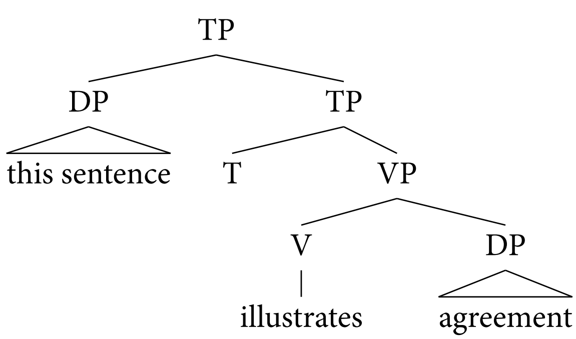
We'll let the constraints weed out those products of *lin* that are ill-formed. Totality will ensure that every vocabulary item in the phrase marker ends up in the string, and Antisymmetry and Transitivity will make sure the linearization is consistent. We'll also need to add a condition that prevents too much scrambling. I'll adopt:

(20) *Contiguity*

Let A be the set of vocabulary items dominated by A and b be a vocabulary item not in A. If b precedes something in A, then b precedes everything in A, and if b follows something in A, then b follows everything in A.

These constraints – Contiguity, Totality, Antisymmetry and Transitivity – are sufficiently draconian that they manage to constrain the structure-to-string mapping almost enough to ensure reasonably accurate outcomes. Imagine that the linearization algorithm did nothing more than generate all possible orderings of vocabulary items and submit them to the constraints. The strings produced would include the correct one and a small number of alternatives. For instance, a linearization algorithm of this sort would produce from (21) a collection of sets that, once filtered through the constraints, would result in those listed in (22). (I will indicate the linearizations with the (more compact) strings they correspond to, rather than with the full sets of ordered pairs.)

(21)



(22)
a. this sentence T illustrates agreement
b. sentence this T illustrates agreement
c. this sentence illustrates agreement T
d. sentence this illustrates agreement T
e. this sentence T agreement illustrates
f. this sentence agreement illustrates T
g. sentence this T agreement illustrates
h. sentence this agreement illustrates T
i. T illustrates agreement this sentence
j. T illustrates agreement sentence this
k. illustrates agreement T this sentence
l. illustrates agreement T sentence this
m. T agreement illustrates this sentence
n. T agreement illustrates sentence this
o. agreement illustrates T this sentence
p. agreement illustrates T sentence this

The ill-formed linearizations in (22) are, many of them, well formed in other languages. For instance, (22f) corresponds roughly to how German would linearize this structure, and (22i) corresponds roughly to how Nuiean would. While not all of these outcomes are ones that we might want to permit cross-linguistically,[2] I will nonetheless treat them all as language-particular possibilities. The step from this range of linearizations to the one that is correct for English, then, engages that component of the theory which models word order variation. There are a variety of proposals in the literature on how to model word order variation. One of those is built into Kayne's linearization scheme. We don't need to choose among them, though, and it will be convenient (and harmless) to avoid engaging the details. In what follows, therefore, I will leave open how the choice from the possibilities allowed by the constraints to the one appropriate for English is made. I will call that portion of the linearization procedure that makes the language particular choice, the "language particular component."

The linearization algorithm will have three parts, then: the function *lin* that produces orderings among the vocabulary items in a sentence, a set of constraints, and a procedure that steers how those constraints choose the linguistically viable ones from that set. We can take all the constraints to be absolute, except Contiguity, which is violated in every case where there is (non-vacuous) movement.

(23) Constraint Evaluation
a. Only Contiguity is violable.
b. Every element of *lin*(P) which incurs *n* violations of Contiguity is ungrammatical if there is an element of *lin*(P) that incurs fewer than *n* violations Contiguity.

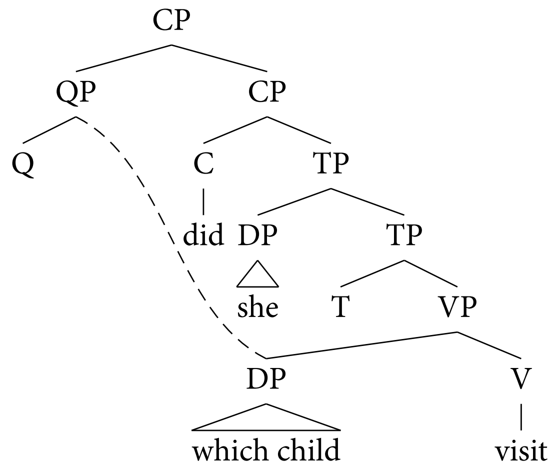
Because *lin* is defined so that it generates every possible set of orderings, including those that are incomplete, it need not produce linearizations that will violate Antisymmetry to begin with. It's the job of the constraints, and the language particular component, to determine which of the sets of orderings offered by *lin* survive. Because the version of Totality in (4b) only requires that every vocabulary item within a structure be mapped onto a position in the resulting string, it will allow orderings that do not take into account all of the positions a vocabulary item might occupy. For these reasons, then, *lin* need not produce

[2] (22n,o) are vanishingly rare according to Dryer (1996).

an ordering that makes a vocabulary item that has two (or more) positions fall into more than one spot in the string. Since this is what Antisymmetry requires, this is how Terseness arises.

To see this, consider some of the orderings that *lin* will produce for (24). These include those in (25).

(24) Which child did she visit?



(25) a. = which child did she visit

which < child	child < did	did < she	she < T	T < visit
which < did	child < she	did < T	she < visit	
which < she	child < T	did < visit		
which < T	child < visit			
which < visit				

b. = did she visit which child

did < she	she < T	T < visit	visit < which	which < child
did < T	she < visit	T < which	visit < child	
did < visit	she < which	T < child		
did < which	she < child			
did < child				

Both of these sets of orderings obey Totality, since every vocabulary item shows up in the resulting strings. They also obey Antisymmetry and Transitivity. However, they differ with respect to Contiguity and that part of the language particular of English that requires wh-phrases to be spoken in their higher position. I will call that the Wh Criterion; we can formulate it with (26).

(26) The Wh Criterion

If just one wh-phrase is merged to CP, then *lin* must position that wh-phrase so that it precedes everything else in that CP.

(25a) obeys the Wh Criterion, but violates Contiguity. (It violates Contiguity because *she* (for example) precedes *visit* but not the other vocabulary items in the VP.) (25b), by contrast, obeys Contiguity but violates the Wh Criterion. The language particular component is inviolable and therefore of these two orderings, (25a) is the better. Further, there is no way of avoiding a violation of Contiguity if the Wh Criterion is to be satisfied. In particular, there are no elements of *lin*((24)) that by virtue of violating Totality, Antisymmetry or Transitivity manage to satisfy the Wh Criterion and also avoid violating Contiguity. There are no candidates that beat out (25a) by relying on violations of one of the other of our constraints on linearization because these will all involve either additional violations of the Wh Criterion or additional violations of

Contiguity. For instance, (27) is equivalent to (25a) in its goodness, except that it has an additional violation of Contiguity.

(27) = which did she visit child

$$\left\{\begin{array}{lllll} \text{which} < \text{did} & \text{did} < \text{she} & \text{she} < \text{T} & \text{T} < \text{visit} & \text{visit} < \text{child} \\ \text{which} < \text{she} & \text{did} < \text{T} & \text{she} < \text{visit} & \text{T} < \text{child} & \\ \text{which} < \text{T} & \text{did} < \text{visit} & \text{she} < \text{child} & & \\ \text{which} < \text{visit} & \text{she} < \text{child} & & & \\ \text{which} < \text{child} & & & & \end{array}\right\}$$

What I've illustrated with Wh Movement, I would like to suggest is more general. In particular:

(28) Movement of DPs involves merging [$_{DP}$ the NP] with an operator.

Let me show how this will play out in cases of QR. It provides a way of understanding some otherwise mysterious ways in which QR differs from Wh Movement.

An obvious difference in Wh Movement and QR is how they get spelled out. The syntax of wh movement allows the moved expression to be spelled out in its higher position, and the language particular constraints of English pick that possibility for "single" constituent questions. For multiple questions, of course, English allows, in fact forces, the wh phrase to be spelled out in its lower position, as in (29).

(29) Which story should you tell which child?

In the case of QR, however, the moved phrase is always spelled out in its lower position. So, that's one difference we would like to capture.

(30) A wh-moved phrase can be spelled out in either of its two positions, but a QR'd expression can only be spelled out in its lower position.

Another difference concerns how QR and Wh Movement are semantically interpreted. We've seen that the NP part of a moved wh-phrase need not be semantically interpreted where it is spoken. QR is different, though. The NP part of QR must be interpreted in the position it is spoken. One way that can be seen is by considering the disjoint reference effect in (31).

(31) every book about Julie she likes every book about Julie.

If *book about Julie* could be interpreted in just its higher position, then we should expect *Julie* and *she* to be able to corefer. But they cannot.

(32) QR'd material must be semantically interpreted where it is spoken, but Wh moved material is able to be semantically interpreted in only its unspoken position.

One of the situations in which QR can be seen arises when a quantificational object contains an elided VP whose antecedent is the VP the object sits in. This can happen when a quantificational object comes with a relative clause, as in (33).

(33) She read every book that I did △.

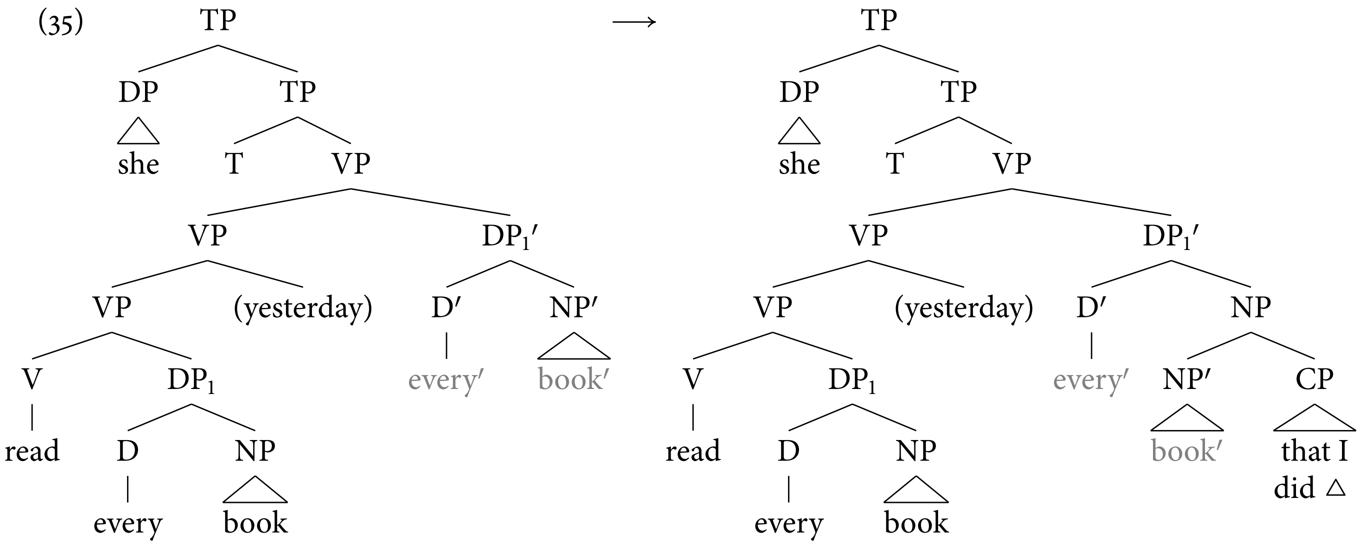
I'll adopt the account of these so-called Antecedent Contained Deletions (ACD) in Fox (2002). On Fox's analysis, ACD is licensed when the relative clause containing the ellipsis has extraposed from the antecedent for that ellipsis.[3]

[3] The original argument for using extraposition structures as the source for ACD is Baltin (1987).

(34) She [$_{\text{VP}}$ read [$_{\text{DP}}$ every book]] (yesterday) [$_{\text{CP}}$ that I did △]

Fox (and Fox and Nissenbaum 1999) argue that these sorts of extraposition operations are the result of "late merging" a clause into a QR'd DP.

(35)



Movement produces a copy of the object DP and then merges that copy into a position outside the VP which serves as antecedent to the ellipsis. Unlike Wh Movement, this higher copy goes unpronounced and, instead, the lower copy is spoken. However, into the higher copy is merged the relative clause containing the elided VP, and this relative clause is pronounced in the position occupied by the higher copy of QR. A reason for using movement, and the copy theory of movement in particular, to model QR is that it provides a simple account of extraposition from NP and, with it, a good account of ACD. It also captures a fact about ACD that Sag (1976) established: the position where an ACD in a DP is resolved is the same position that the quantifier heading that DP is interpreted.

We can now see another difference between QR and Wh Movement: when material in the higher copy in Wh Movement is pronounced, that material gets linearized so that it precedes everything else in the clause it is dominated by. But when material in the higher copy of QR is pronounced, it follows everything else in the phrase it is dominated by. Wh Movement goes to the left, but QR goes to the right. This difference too should be derived.

(36) Unlike Wh-moved material, when QR'd material is spelled out in its higher position, it shows up to the right of the phrase it is merged to.

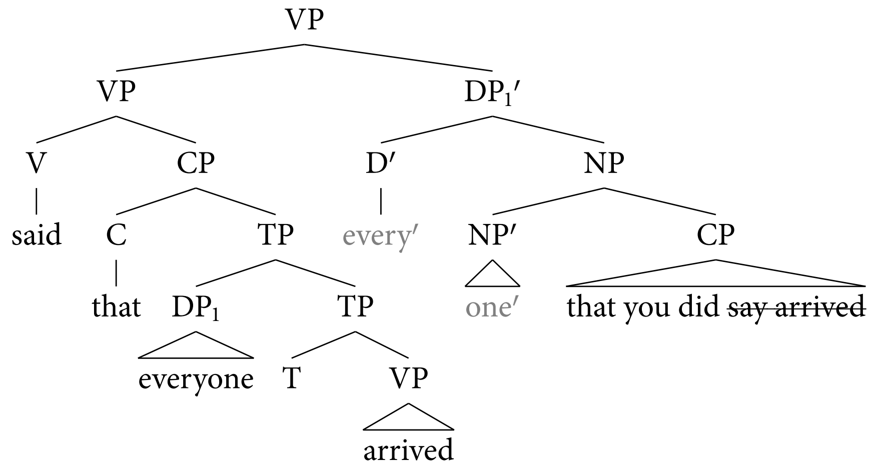
One of the most interesting pieces of support for Fox's analysis of ACD is the contrast in (37), from Tiedeman (1995).

(37) a. * I said that everyone you did △ arrived.
b. I said that everyone arrived that you did △.
△ = said that *x* arrived

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

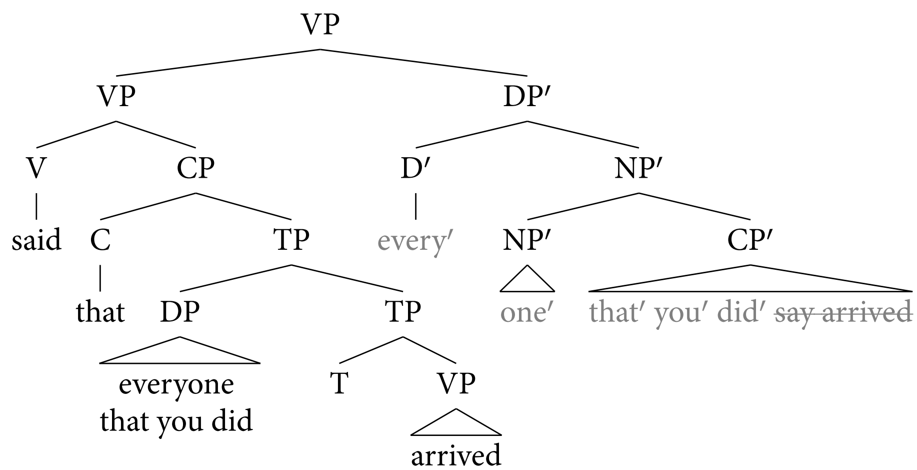
The difference is credited to extraposition being able to generate the string in (37b) but not (37a). The representation in (38) is only available for (37b). *lin* will put the material in the embedded subject between the complementizer and the embedded VP, and linearize the extraposed relative clause so that it follows everything the higher copy has merged with.

(38)



Hidden in this example, however, is yet another illustration of the difference between QR and Wh Movement semantics. To see this difference, consider a derivation in which the relative clause is part of the lower copy, and therefore maps onto the string in (37a). This derivation will QR the subject and its relative clause together into a higher position, one that puts the elided VP outside of its antecedent, as in (39).

(39)



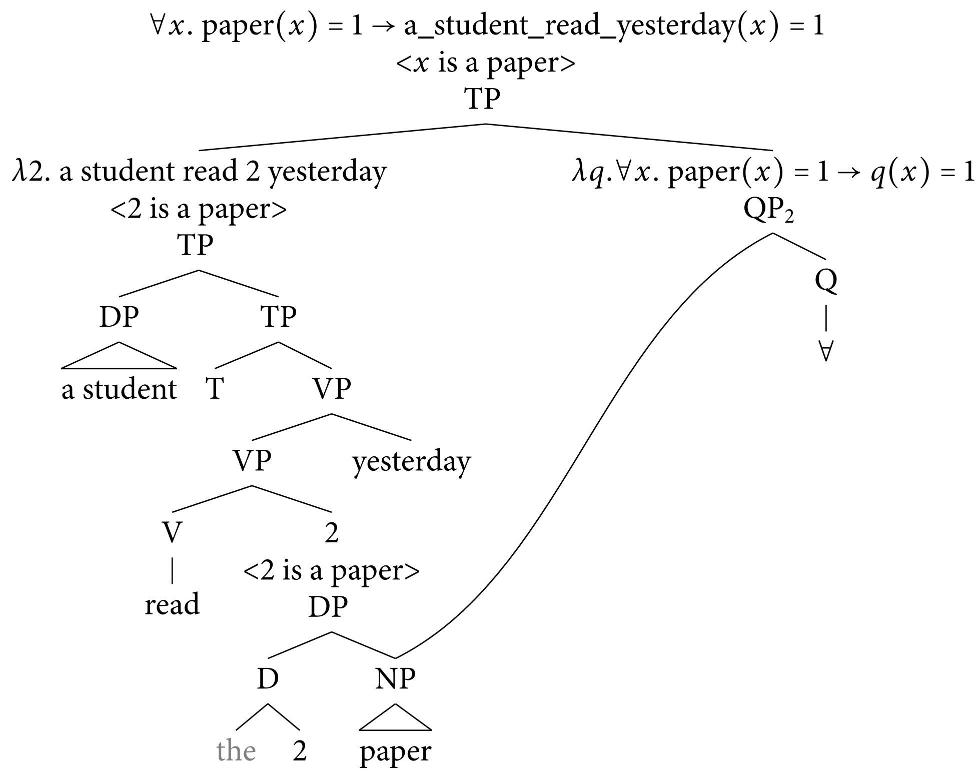
This representation is ill-formed only if the relative clause in the lower copy must be semantically interpreted. Here, then, is another illustration of (32).

Let's now consider how QR can be modeled in our remerge theory of movement. As with Wh Movement, let's let there be a definite description in the lower position that is bound by an operator in the higher position. Unlike the case with questions, however, the determiner in the lower position and the quantifier in the higher position will both combine semantically with the NP. That is because the denotation of quantifiers requires that they relate, semantically, the meaning of the NP they combine with with the meaning of the clause they are in. Quantifiers like *every*, for instance, have a meaning something like that indicated for ∀ in (40).

(40) $[\![\forall]\!] = \lambda p.\lambda q.\forall x.p(x) = 1 \rightarrow q(x) = 1$

QR will therefore produce a representation like that in (41).

(41) A student read every paper yesterday.



As with questions, we want the form of D to be determined by the quantifier that is in the higher position. When the quantifier is ∀ we'll want D to be spelled-out as *every*, when it's "¬ any" we'll want D to be spelled-out as *no*, and so on. We can't do this with Agree, as Q does not c-command D and Agree only holds between things that are in a c-command relation. I suggest instead that D+Q are brought together by the morphology. Morphology is equipped with processes that allow two terminals to be mapped onto one vocabulary item. These processes show up in a variety of ways. They are responsible for mapping a preposition and determiner onto one lexical item in certain contexts in German (and other IndoEuropean languages), for instance.

(42) an dem Tisch → am Tisch
on the table → on.the table

These processes go by various names, and there seem to be slight differences in the conditions under which they may operate that depends on the case being modeled.[4] But across all these cases, there is a similar locality condition on the two terminals that combine: they must be so close together that, under normal circumstances, they would show up adjacent in the string. Let us simply adopt this descriptive consequence as a well-formedness condition on "fusion," as I will call the process that bring D and Q together into one word.

(43) X and Y can fuse only if *lin* assigns them adjacent positions.

The language particular component of English causes Q and D both to come before everything else in the phrases they head.

(44) *lin*(QP) puts Q before everything else in QP
lin(DP) puts D before everything else in DP

4 See Pranka (1983), Marantz (1988, 1984), Halle and Marantz (1993), Bobaljik (1995), Embick and Noyer (2001), Matushansky (2006) and references cited therein.

If D and Q were to be linearized in non-QR contexts, then, they would show up in adjacent positions. They are, therefore, possible fusers.

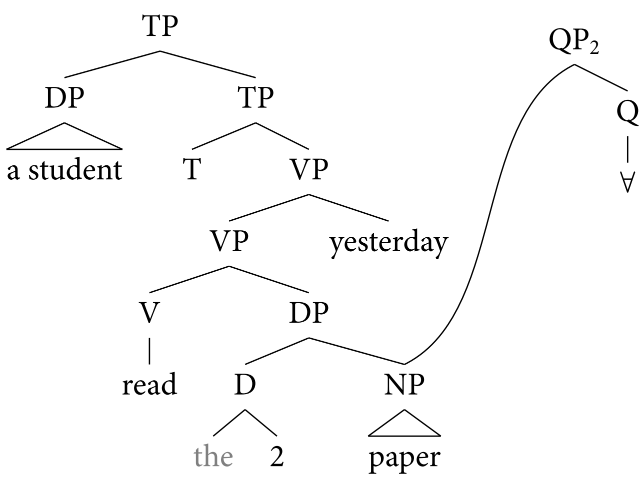
However, if D and Q are to try to fuse in structures of QR, then the condition in (43) will stand in the way. In (41), for example, *lin* will put things between D and Q and prevent adjacency. There are two strings that *lin* could produce from (41). If the NP containing *paper* is linearized in the spot assigned to QP, *lin* will deliver (45a), and if this NP is linearized in the spot assigned to DP, *lin* will deliver (45b).

(45) a. a student read D yesterday Q paper
     b. a student read D paper yesterday Q

Both of these linearizations violate Contiguity, but they only violate Contiguity to the extent required by the multidominant representation. These, then, are the minimal violators of Contiguity and therefore the candidate winners. In both of them, D and Q are separated by *yesterday* and so fusion is blocked. If we assume that the English lexicon does not provide vocabulary items for the D or Q in these structures, we will have a sentence that cannot be spelled out.

I suggest, then, that fusion is achieved before the entire QR structure is built. Let's imagine the stage in the derivation that leads to (41) immediately before the QP has merged with TP. This will look like (46).

(46)



This representation has two root nodes: TP and QP. If we define *lin* so that it runs on root nodes, then in this case it will apply to TP and QP independently, and produce the ordered pairs in (47).

(47)

lin(TP) = {

a < student	student < T	read < D	D < paper	paper < yesterday
a < T	student < read	read < paper	D < yesterday	
a < read	student < D	read < yesterday		
a < D	student < paper			
a < paper	student < yesterday			
a < yesterday				

}

lin(QP) = ∀ < paper

This linearization puts nothing between D and ∀, and so they can fuse. Once they've fused and been mapped onto *every*, it is *every* that will occupy the positions assigned to D and ∀ in (47). After QP and TP have merged, no new ordering statements need to be added to meet the requirement of Totality. That's because Totality only requires that every vocabulary item in a phrase marker be assigned a position relative to every other vocabulary item, and that will be achieved by (47) (after D and Q fuse) for all the vocabulary items that will be matched to the terminals in (46). So (47) will be the linearization for the final representation (=(41)); (47) corresponds to the string in (48), which is precisely the right outcome.

(48) a student read every paper yesterday

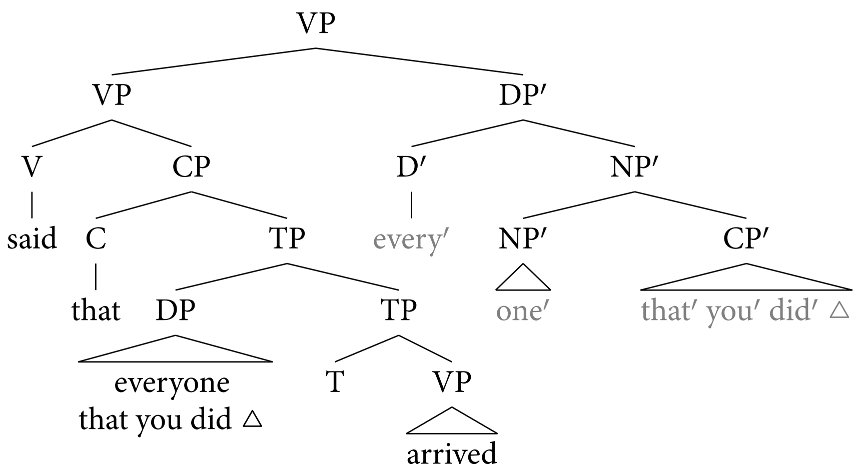
Once QP has merged with TP to form the representation that is semantically interpreted, one of two options are possible: *lin* can be run again, or the ordered pairs in (47) can simply be combined. In either case, the resulting set of ordered pairs will meet Totality, Transitivity and Antisymmetry, and it will violate Contiguity no more than is made necessary by QR. We have just derived the fact that QR'd phrases get spelled out in their lower position.

Let's consider next how this system derives (32).

(32) QR'd material must be semantically interpreted where it is spoken, but Wh moved material is able to be semantically interpreted in only its unspoken position.

We've already seen how the case of Wh Movement works. An interrogative DP can be semantically interpreted entirely in its lower position, and yet be part of a phrase that *lin* puts in a different position. Let's now consider why something parallel is not possible for QR. One of the cases I used to demonstrate (32) is (39). Under a copy theory account, this gets the representation indicated.

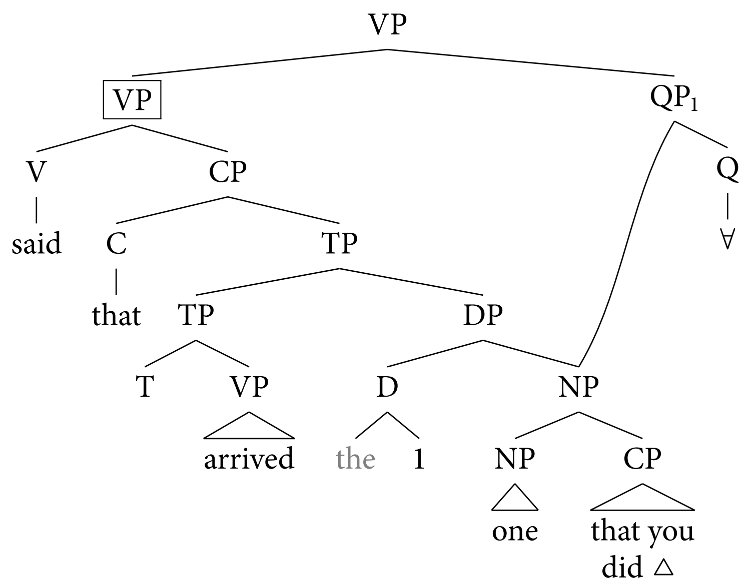
(39) * I said that everyone you did △ arrived.



Because this representation cannot resolve the ellipsis it contains we have concluded that the relative clause in the lower copy must be semantically interpreted.

If QR is modeled with the multidominant representations proposed here, however, the string in (39) can only get the representation in (49). Like the representation in (39), The structural requirements for resolving the ellipsis are not met in (49), either, since the ellipsis is inside its antecedent: VP.

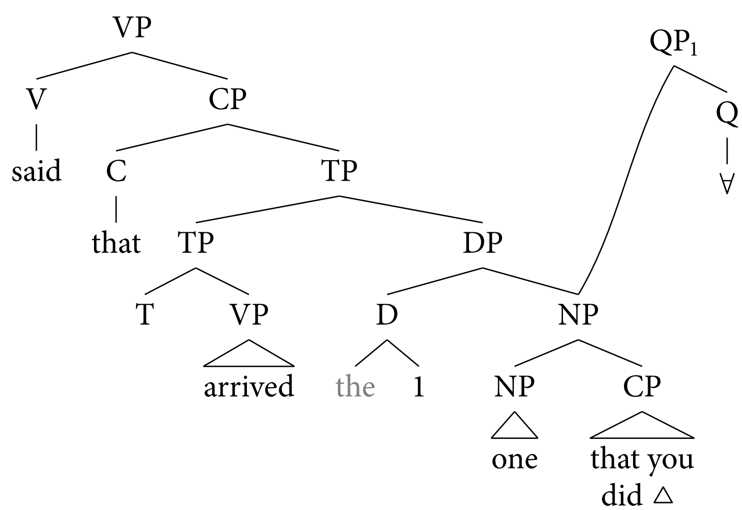
(49)



To see why it's this structure that corresponds to the string in (39), consider how *lin* will deliver the linearization of (49).

As we've seen, *lin* must run before the QP is merged to VP in order to fuse D and Q into *every*. So, the string associated with (49) will be built upon the output *lin* produces from applying to (50).

(50)



That linearization is (51).

(51)

lin(VP) =

said < that	that < D	D < one	one < that	that < you	you < did	did < T	T < arrived
said < D	that < one	D < that	one < you	that < did	you < T	did < arrived	
said < one	that < that	D < you	one < did	that < T	you < arrived		
said < that	that < you	D < did	one < T	that < arrived			
said < you	that < did	D < T	one < arrived				
said < did	that < T	D < arrived					
said < T	that < arrived						
said < arrived							

lin(QP) =

∀ < one	one < that	that < you	you < did
∀ < that	one < you	that < did	
∀ < you	one < did		
∀ < did			

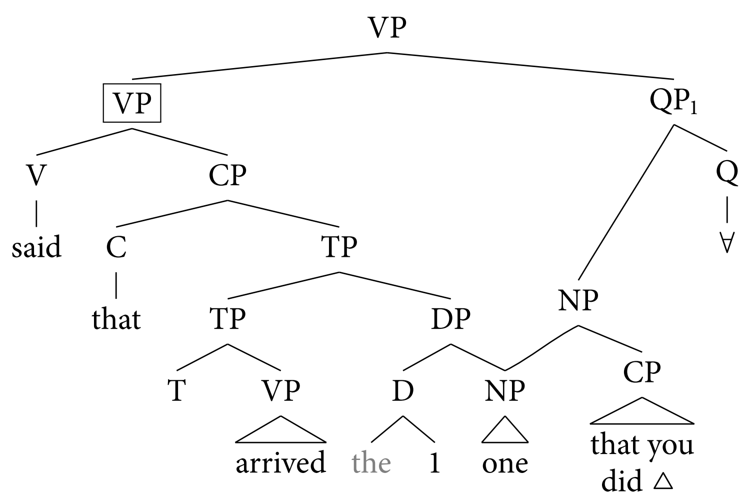
When *lin* runs again, subsequent to merging QP and VP, no new ordered pairs will be introduced, and so we will get the simple union of *lin*(VP) and *lin*(QP). That corresponds to the string in (52).

(52) said that everyone that you did arrived

This string therefore corresponds to a structure that does not resolve the ellipsis: just what we want.

To resolve the ACD, we must have a representation that involves "late merge" of the relative clause. Under the present proposal, this will look like (53).

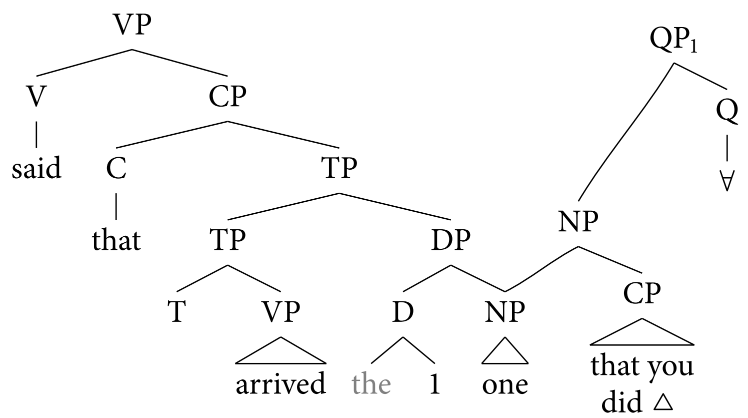
(53)



In this structure, the relative clause is not within the VP that serves as antecedent, and so the ellipsis can be resolved. To see that this structure does not correspond to the string in (39), consider how *lin* will manufacture a linearization for it.

As always, *lin* will be forced to apply before the QP has merged into the larger structure. In this case, that will look like (54).

(54)



From (54), *lin* will produce (55).

(55) = said that everyone arrived & everyone that you did

$$
lin(\text{VP}) = \left\{ \begin{array}{lllll}
\text{said} < \text{that} & \text{that} < \text{D} & \text{D} < \text{one} & \text{one} < \text{T} & \text{T} < \text{arrived} \\
\text{said} < \text{D} & \text{that} < \text{one} & \text{D} < \text{T} & \text{one} < \text{arrived} & \\
\text{said} < \text{one} & \text{that} < \text{T} & \text{D} < \text{arrived} & & \\
\text{said} < \text{T} & \text{that} < \text{arrived} & & & \\
\text{said} < \text{arrived} & & & &
\end{array} \right\}
$$

$$
lin(\text{QP}) = \left\{ \begin{array}{llll}
\forall < \text{one} & \text{one} < \text{that} & \text{that} < \text{you} & \text{you} < \text{did} \\
\forall < \text{that} & \text{one} < \text{you} & \text{that} < \text{did} & \\
\forall < \text{you} & \text{one} < \text{did} & & \\
\forall < \text{did} & & &
\end{array} \right\}
$$

Notice that because the relative clause is not yet inside VP, it is not included in the string associated with VP. As a consequence, only *lin*(QP) has information about where the relative clause will be positioned: it will follow everything else in QP. It is only after QP has merged with VP — to form (53) — that *lin* can order the material in the relative clause with the material in the VP. In order to satisfy Totality, *lin* will therefore have to apply again after (53) is formed.

The ordered pairs that this second run of *lin* will add to (55) must not only satisfy Totality, but they must also satisfy Antisymmetry and Contiguity. The best satisfaction of Contiguity will be linearizations that keep the material in the relative clause together with Q and put it either all before the VP, or all after the VP. If *lin* puts the relative clause before the VP, however, the ordered pairs it will generate will include (56).

(56) that < every, that < one, you < every, you < one, …

If these are added to the ordered pairs in (55), a violation of Antisymmetry will ensue. For these reasons, the ordered pairs *lin* will generate when applied to (53) will add to (55) those in (57): ones in which the relative clause follows the VP.

(57)

$$
\text{The new outputs from } lin(\,(53)\,) = \left\{ \begin{array}{llll}
\text{said} < \text{that} & \text{that} < \text{that} & \text{T} < \text{that} & \text{arrived} < \text{that} \\
\text{said} < \text{you} & \text{that} < \text{you} & \text{T} < \text{you} & \text{arrived} < \text{you} \\
\text{said} < \text{did} & \text{that} < \text{did} & \text{T} < \text{did} & \text{arrived} < \text{did} \\
\text{said} < \forall & \text{that} < \forall & \text{T} < \forall & \text{arrived} < \forall
\end{array} \right\}
$$

This corresponds to the string in (58).

(58) ...said that everyone arrived that you did △

For the relative clause to be positioned outside the VP that is serving as the antecedent for the ellipsis it contains, it will necessarily be positioned linearly outside the string that corresponds to that VP. This result is perfectly general. We derive (32): QR cannot put spoken material in a position where it is not semantically interpreted.

In fact, this also sketches how (36) is manufactured.

(36) When QR'd material is spelled out in its higher position, it shows up to the right of the phrase it is merged to.

In stepping through how *lin* positions the late merged relative clause in (55), we saw not only that it must put that relative clause outside the phrase to which the QP merges, but also that it must put that relative clause to the right of the phrase to which the QP merges. This arises because the relative clause is forced to follow everything else within the QP that dominates it, and, at the same time, all that other material in the QP gets linearized within the phrase to which the QP later merges. Thus, when the QP merges with some phrase, the relative clause it contains will have to linearize itself with respect to that phrase in the same way that it is linearized with respect to the rest of the QP's material inside that phrase.

References

Baltin, Mark. 1987. Do antecedent-contained deletions exist? *Linguistic Inquiry* 18:579–596.

Bobaljik, Jonathan. 1995. Morphosyntax: The syntax of verbal inflection. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge.

Cable, Seth. 2007. The grammar of Q: Q-particles and the nature of wh-fronting, as revealed by the wh-questions of Tlingit. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Cable, Seth. 2008. There is no such thing as pied-piping. Unpublished paper, University of Massachusetts at Amherst.

Chierchia, Gennaro. 1992. Questions with quantifiers. *Natural Language Semantics* 1:181–234.

Citko, Barbara. 2005. On the nature of merge: External merge, internal merge, and parallel merge. *Linguistic Inquiry* 36:475–496.

Dryer, Matthew S. 1996. Word order typology. In *Handbook on syntax*, ed. J. Jacobs, volume 2, 1050–1065. Walter de Gruyter.

Elbourne, Paul D. 2005. *Situations and individuals*. MIT Press.

Embick, David, and Rolf Noyer. 2001. Movement operations after syntax. *Linguistic Inquiry* 32:555–595.

Engdahl, Elisabet. 1980. The syntax and semantics of questions in Swedish. Doctoral Dissertation, University of Massachusetts, Amherst, Massachusetts.

Engdahl, Elisabet. 1986. *Constituent questions*. Dordrecht, The Netherlands: D. Reidel Publishing Company.

Fox, Danny. 2002. Antecedent-contained deletion and the copy theory of movement. *Linguistic Inquiry* 33:63–96.

Fox, Danny. 2003. On logical form. In *Minimalist syntax*, ed. Randall Hendrick, 82–123. Oxford: Blackwell Publishers.

Fox, Danny, and Jon Nissenbaum. 1999. Extraposition and scope: a case for overt QR. In *Proceedings of the West Coast Conference on Formal Linguistics 18*, ed. Sonya Bird, Andrew Carnie, Jason D. Haugen, and Peter Norquest, 132–144. Somerville, Massachusetts: Cascadilla Press.

Frampton, John. 2004. Copies, traces, occurrences, and all that: Evidence from Bulgarian multiple *wh*-phenomena. Unpublished manuscript, Northeastern University.

Gärtner, Hans-Martin. 1997. Generalized transformations and beyond. Doctoral Dissertation, University of Franfurt/Main.

Halle, Morris, and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In *The view from building 20*, ed. Kenneth Hale and Samuel Jay Keyser, 111–176. Cambridge, Massachusetts: MIT Press.

Kayne, Richard S. 1994. *The antisymmetry of syntax*. Cambridge, Massachusetts: MIT Press.

Kobele, Gregory Michael. 2006. Generating copies: An investigation into structural identity in language and grammar. Doctoral Dissertation, University of California, Los Angeles, Los Angeles, CA.

Marantz, Alec. 1984. *On the nature of grammatical relations*. Cambridge, Massachusetts: MIT Press.

Marantz, Alec. 1988. Clitics, morphological merger, and the mapping to phonological structure. In *Theoretical morphology: Approaches in modern linguistics*, ed. Michael Hammond and Michael Noonan, 253–270. San Diego, California: Academic Press.

Matushansky, Ora. 2006. Head movement in linguistic theory. *Linguistic Inquiry* 37:69–109.

Nunes, Jairo. 2001. Sideward movement. *Linguistic Inquiry* 32:303–344.

Nunes, Jairo. 2004. *Linearization of chains and sideward movement*. Linguistic Inquiry Monographs. Cambridge, Massachusetts: MIT Press.

Pranka, Paula. 1983. Syntax and word formation. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Rullmann, Hotze, and Sigrid Beck. 1998. Presupposition projection and the interpretation of *which*-questions. In *Proceedings of SALT VIII*, ed. Devon Strolovitch and Aaron Lawson, 215–232. Ithaca, NY: CLC Publications.

Sag, Ivan. 1976. Deletion and logical form. Doctoral Dissertation, Massachusetts Institute of Technology.

Starke, Michal. 2001. Move dissolves into merge: A theory of locality. Doctoral Dissertation, University of Geneva.

Tiedeman, Robyne. 1995. Some remarks on antecedent contained deletion. In *Minimalism and Linguistic Theory*, ed. Shosuke Haraguchi and Michio Funaki, 67–103. Tokyo: Hituzi Syobo.

de Vries, Mark. 2007. Internal and external remerge: On movement, multidominance, and the linearization of syntactic objects. Unpublished manuscript, University of Groningen.