The Empty Category Principle and Linearization

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

Workshop on Comparative Germanic Syntax: Leipzig
5 October 2013

Three components to a theory about what can move:

(1) a. Islands:
  i. ?? What has she asked if you'd bought *t*?
  ii. What has she said that you'd bought *t*?

b. Categorical Prohibitions:
  i. It's [$_{CP}$ how Jerry runs] that she described *t*.
  ii. * It's [$_{IP}$ Jerry runs] that she described how *t*.
  iii. * It's [$_{IP}$ Jerry to be happy] that she considers *t*.

c. The Empty Category Principle (ECP):
  i. * Who has she asked if *t* had bought something?
  ii. ?? What has she asked if you'd bought *t*?

A first approximation of the ECP.

(2) ECP
The trace of movement must be in construction with an X^0 of a certain kind.

One X^0 that satisfies the ECP is a verb.

There are certain mysteries with this system. There are interactions between the kinds of things moved and the class of X^0s that satisfy the ECP. For instance, when DPs move, their trace satisfies the ECP by being in construction with a lexical verb.

(3) What did she see *t* ?

But though VPs can move, they cannot move when they are in construction with a lexical verb.

(4) a. It's [$_{VP}$ eat more chips] that she shouldn't *t*.
  b. * It's [$_{VP}$ him jump] that she saw *t*.
  c. * It's [$_{VP}$ jump] that she saw him *t*.

Lobeck (1992) suggested that the ECP also plays a role in licensing VP ellipsis. This is animated by a near match in the environments where VPs can elide and where they can leave traces.

(5) a. It's [$_{VP}$ eat rutabagas] that Holly should *t*.
  b. Sally should eat rutabagas and Holly should ~~eat rutabagas~~ too.

(6) a. It's [$_{VP}$ eaten rutabagas] that Holly hasn't *t*.
  b. Sally has eaten rutabagas but Holly hasn't ~~eaten rutabagas~~.

(7) a. * It's [$_{VP}$ discussed] that you recall Holly being *t*.
  b. * I remember Doc Golightly being discussed, but you recall Sally being ~~discussed~~.

(8) a. * It's [$_{VP}$ discussed] that Holly is being *t*.
  b. * Doc Golightly is being discussed and Sally is being ~~discussed~~ too.

(9) a. * It's [$_{VP}$ eat rutabagas] that Holly made me *t*.
  b. * Madame Spanella made me eat rutabagas and Holly let me ~~eat rutabagas~~ too.

Lobeck's thesis is difficult to confirm empirically as the set of phrases that can elide and the set of phrases that can move have a small intersection.

Aelbrecht and Haegeman (2012), for instance, show that VP ellipsis does not track the contexts where VP topicalization is allowed, and because TPs do not move, we cannot see if the heads that license Sluicing are also those that license traces.

(10) a. Smith at something, but I don't know [$_{CP}$ what C [TP ~~Smith ate~~]].
b. * It's [$_{TP}$ Smith ate] , that I asked what *t* .

Nonetheless, I will adopt Lobeck's thesis.

(11) ECP
Traces and ellipsis sites must be in construction with certain X^0s.

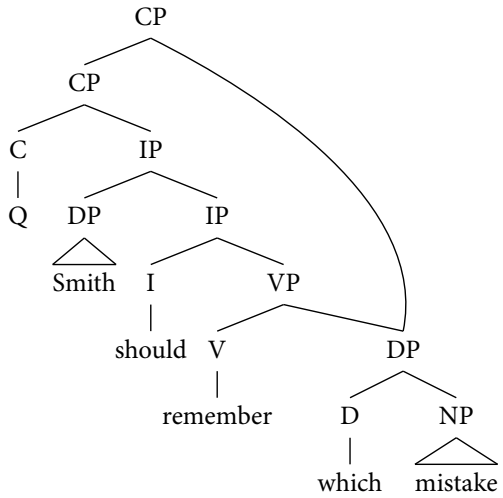
Where does the ECP come from?

- Traces and ellipsis sites are silent pronouns, and the ECP traces back to the conditions that involve licensing silent pronouns. See Rizzi (1990).
- Traces and ellipsis sites are deleted copies of the moved item/antecedent, and the ECP traces back to the conditions on deletion. See Nunes (1999), and much subsequent work.

Both of these ideas correctly explain why traces and ellipsis sites should be subject to the same condition by making traces and ellipsis sites the same thing: silent pronouns on the one hand, or deleted copies on the other.

(11) is a challenge, however, for remerge theories of movement because traces of movement on that view are neither pronouns nor deleted copies. Instead, traces of movement are just exactly the moved phrase, but linearized in a different position. (See, for instance, Engdahl (1986).)

(12)



One way of expressing how a structure like (12) yields the desired string – *which mistake Smith should remember* – is to rig the linearization scheme so that it positions *which mistake* as if it were only in the higher of its two positions. A standard way of ensuring that structures like (12) force the linearization algorithm to manufacture a string where the phrase in two positions acts as if it were in only one of those positions is to use Kayne (1994)'s Antisymmetry condition. (This is the technique employed by Nunes 1999.)

(13) Antisymmetry
A linearization of a phrase marker cannot cause a terminal *a* to both precede and follow another terminal *b*.

There should also be a condition that requires everything in a phrase marker to be mapped into the string that the linearization procedure produces. If that condition says that every position in a phrase marker must correspond to a position in the string, then structures like (12) will present a problem. The condition in Kayne that ensures everything in a phrase marker gets into the linearization has this effect.

(14) Totality
Every position in a phrase marker must be used to form the linearization.

We could see the ECP, then, as a description of the places where this conflict can be resolved. In the context of movement, then, we could think of the ECP as having its source in allowing a linearization of a tree with multidominance.

(15) The ECP traces back to the conditions where Totality is suspended.

What we need is a way of stating Totality so that it can be suspended in the kinds of environments that the ECP describes. Our first job is to reframe Totality for this job.

I'll adopt the familiar view that the linearization algorithm produces a set of pairwise precedence relations between the terminals in a phrase marker. For English, I'll assume that the algorithm is guided by the informal descriptive guidelines in (16).

(16) Where "<" is interpreted as "precedes" and $\mathscr{L}$ is the linearization algorithm applying to a sentence containing XP, YP and X^0:

a. If y is dominated by YP and YP is the sister of X^0 dominating x, $\mathscr{L}$ sanctions x<y.

b. If y is dominated by YP and YP is the sister of XP, dominating x, and YP is in Specifier position, $\mathscr{L}$ sanctions y<x.

c. If y is dominated by YP, and YP is the sister of XP, dominating x, and YP is in adjunct position, $\mathscr{L}$ sanctions x<y.

For (12) this will allow the set in (17).

(17)

which < mistake	mistake < Q	Q < Smith	Smith < should	should < remember
which < Q	mistake < Smith	Q < should	Smith < remember	
Q < which	Smith < mistake			
which < Smith	mistake < should	Q < remember		
Smich < which	should < mistake			
which < should	mistake < remember			
should < which	remember < mistake			
which < remember				
remember < which				

Note that this set contains the Antisymmetry violating orderings produced by *which mistake* being in two positions. $\mathscr{L}$, then, expresses the range of possible positions that terms can occupy in the string.

The job of Totality will now be done by a process that actually forms the string from a syntactic representation. This process imposes adjacency relationships on the terminals in (12) to which (16) acts as a filter.

(18) FORM STRING

Let X and Y be sisters in a phrase marker. If a terminal, x, in X is pronounced then pronounce y, a terminal in Y, so that it is adjacent to X.

FORM STRING enforces Totality by making every pronounced terminal adjacent to some other terminal. This works transitively through the tree to ensure that every terminal gets into the string. For instance, in (12) because *which* is pronounced, so must *mistake*, since the NP containing *mistake* is a sister to *which*. (17) requires that *which* precede *mistake*. Similarly, because *which* (and *mistake*) is pronounced, some terminal in the CP that is a sister to the DP *which mistake* must be pronounced adjacent to DP. We should understand "being adjacent to a phrase" to mean "being adjacent to a terminal in that phrase." In this situation, (16) only allows this to be satisfied by putting Q right-adjacent to *mistake*. Because Q is pronounced (even though its pronunciation is silence), so must something in the TP that is its sister, and this is satisfied by putting *Smith* right-adjacent to it. This repeats until every terminal is in the string.

The entire linearization procedure, then, has two parts. There are the precedence relations that $\mathscr{L}$ sanctions, and FORM STRING which generates the string in compliance with those sanctions. So:

(19) The linearization procedure is a mapping from syntactic structures to strings such that:

a. The language particular function $\mathscr{L}$ applies to a root node and forms the set of pairwise precedence relations between terminals that are allowed, and

b. FORM STRING generates a string in which every pair of terminals *a* and *b*: $a < b$ only if $a < b$ is in the set generated by $\mathscr{L}$.

I've been explicit that $\mathscr{L}$ applies to root nodes because that will do some work for us later. It becomes relevant, as we shall see, when a sentence has more than one root node. (And, yes, they can.)

Here, then, is the ECP.

(20) ECP

Certain X^0s are exempt from FORM STRING.

I'm going to argue that (20) is correct, and that the thesis that the ECP governs both the positions that traces can be and the positions that ellipses can be is also correct. The argument is based on giving an account for Andrew's amalgams, a construction discussed (first?) by Lakoff (1974). (21) provides an example of an Andrews amalgam.

(21) Sally will eat [I don't know what] today.

The peculiarity of (21) that makes it interesting is how the bracketed clause (the interrupting clause) manages to function as the object of the verb, *ate*, in the

hosting clause. We need to find a way of giving (21) a syntax that allows it to have a meaning parallel to (22).

(22) Sally will eat something today but I don't know what.

Marlies Kluck and Maximiliano Guimaräes wrote dissertations in the last two years that provide very complete analyses of Andrews amalgams. I will suggest an account here very similar to that in Guimarães (2004), but using the general approach to the construction found in Kluck (2011). In this account the *what* in (21) will be in both interrupting and hosting clause. In the interrupting clause it will be interpreted as the interrogative *what*; in the hosting clause it will be interpreted as "something." I'll call this term, the shared item.

Two properties of Andrews amalgams that help lead us to an account are.

(23) The interrupting clause behaves like a root clause.

a. Bob heeft [je raadt nooit hoeveel koekjes] gestolen.
Bob has [you guess never how many cookies] stolen
(Bob has stolen you'll never guess how many cookies.)

b. * Bob heeft [je nooit hoeveel koekjes raadt] gestolen.
Bob has [you never how many cookies guess] stolen
(Bob has stolen you'll never guess how many cookies.)

(Kluck 2011, (14): 55)

(24) Most of the interrupting clause is not in the scope of the host clause

a. * Almost every student$_1$ kissed [he$_1$ didn't even remember how many classmates].

b. ? He$_1$ had been kissing [the professor$_1$ (himself) didn't even remember how many students].

compare:

* He$_1$ had been kissing many students that the professor$_1$ (himself) didn't remember.

(based on Kluck 2011, (170): 97 & (195): 102)

This leads to the conclusion that the interrupting clause and the clause it interrupts are, in some sense, two independent clauses.

On the other hand, Kluck shows that the interrupting clause is positioned syntactically within the host clause in just the places that it should if it were fulfilling the semantic role that it seems to fulfill. In (21), for instance, it seems to provide the object of *ate*, and it can be in just the places that objects can be in this sentence.

(25) a. i. Sally will eat [I don't know what] today.
ii. Sally will eat [the rutabagas] today.
b. i. Sally will eat today [I don't know what].
ii. Sally will eat today [the rutabagas].
c. i. * Sally will [I don't know what] eat today.
ii. * Sally will [the rutabagas] eat today.

And when an interrupting clause seems to provide the object of a preposition, it can only be in the position that objects of prepositions can be.

(26) a. i. Sally stepped into [I won't say what] on her way home.
ii. Sally stepped into [the nattoo pot] on her way home.
b. i. * Sally stepped into on her way home [I won't say what].
ii. * Sally stepped into on her way home [the nattoo pot].
c. i. * Sally stepped [I won't say what] into on her way home.
ii. * Sally stepped [the nattoo pot] into on her way home.

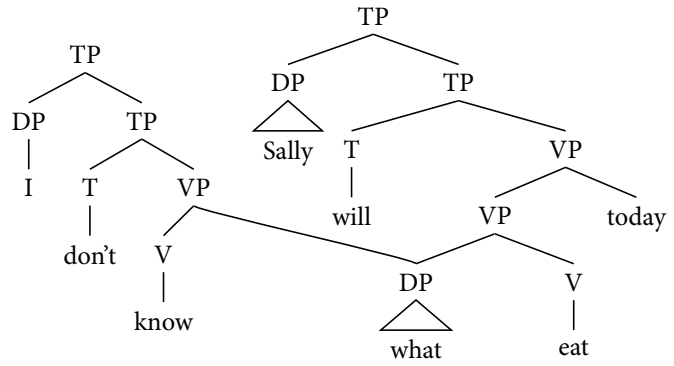
And when an interrupting clause seems to play the role of an adjective, it can only appear in the positions that adjectives can.

(27) a. i. Sally ate [I don't know how smelly] a dinner.
ii. Sally ate [so smelly] a dinner.
b. i. * Sally ate a dinner [I don't know how smelly].
ii. * Sally ate a dinner [so smelly].
c. i. Sally became [I don't know how sick] yesterday.
ii. Sally became [so sick] yesterday.
d. i. * Sally became yesterday [I don't know how sick].
ii. * Sally became yesterday [so sick].

We therefore want the interrupting clause to be enough a part of the hosting clause syntactically that we can use the normal syntax to determine its position.

A way of doing that would be to let a part of the interrupting clause be in the hosting clause, but leave all the rest of it outside. Riemsdijk (1998, 2000, 2006), Wilder (1998), and Guimarães (2004) suggest doing that with multidominant phrase markers. This would give to (21) a representation something like (28).

(28)



There are facts which could be construed as evidence for this geometry. The shared material, unlike the rest of the interrupting clause, does seem to be within the scope of stuff in the hosting clause.

(29) a. Almost every student$_1$ kissed [you can't imagine how many of his$_1$ classmates].

(based on Kluck 2011, (169): 97)

b. * He$_1$ kissed [I don't know how many of the professor$_1$'s students].

That follows from (28) because (28) puts the wh-phrase inside the hosting clause, and therefore within the scope of material it contains.

However, (28) leaves out that the interrupting clause involves sluicing. The *what* in (28) is not the object of *know*, but the remnant wh-phrase of an embedded question that has been sluiced. Kluck (2011) makes an excellent case for this. Among other things, we find the characteristic "form matching" facts that are indicative of ellipsis. The morphological form of the wh-phrase in a sluice matches that which would be found from its putative source.

(30) a. Er will jemandem schmeicheln, aber sie wissen nicht wem.
he will somone-dat flatter, but they know not who-dat
(They will flatter someone, but they don't know who.)

b. * Er will jemandem schmeicheln, aber sie wissen nicht wen.
he will somone-dat flatter, but they know not who-acc
(They will flatter someone, but they don't know who.)

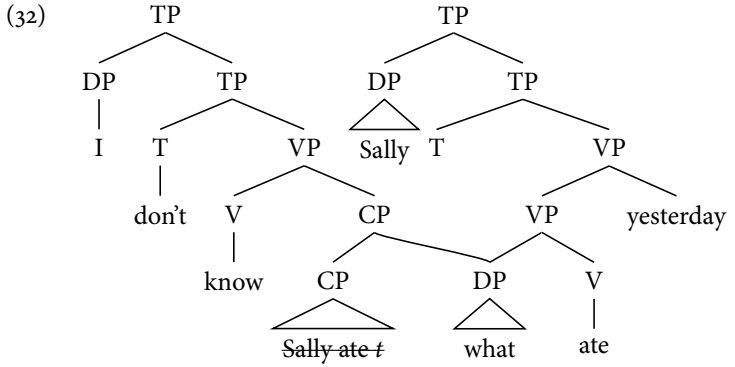
And the same is true in precisely the same way of Andrews amalgams.

(31) Bea wollte [ich weiss nicht mehr wem/*wen] schmeicheln.
Bea wanted [I know not anymore who-dat/who-acc] flatter
(Bea wanted to flatter I don't know who.)

(Kluck 2011, (83): 184)

To put the sluice in (28), we'd have to embrace something like (32).

(32)



To capture the meaning, we'll have to let *what* be interpreted as an interrogative in the interrupting clause and as an non-interrogative in the host clause. We must assume, then, that the denotation of *what*, and other question words, is neutral between these two meanings, and that whether it gets a question or declarative interpretation is determined by material elsewhere in the clause it is within.

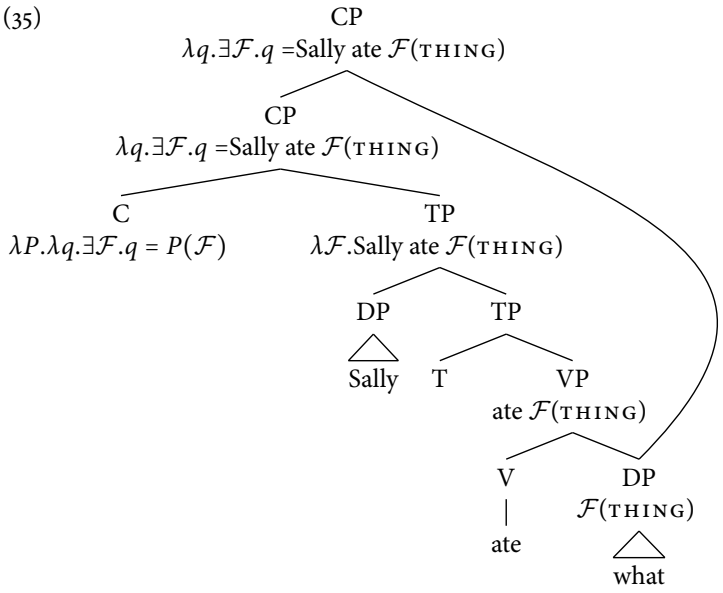
Imagine, for instance, that wh-phrases include a choice function, $\mathcal{F}$, which picks a member from the predicate denoted by the other part of the wh-phrase. So:

(33) a. $\mathcal{F}(\mathrm{P}_{<e,t>}) = a$, such that $\mathrm{P}(a)$.
b. $[\![\text{what}]\!] = \mathcal{F}(\textsc{thing})$

The wh-phrase is pronounced in English in the higher of the two positions it occupies and is semantically interpreted in the lower of the two positions. I'll assume, somewhat parochially, that the remerged wh-phrase introduces at the TP level a λ binder for the choice function it introduces in its "trace" position. Let the question morpheme, Q, have the denotation in (34), and we'll derive a Kartunnen-style meaning for questions, as indicated in (35).

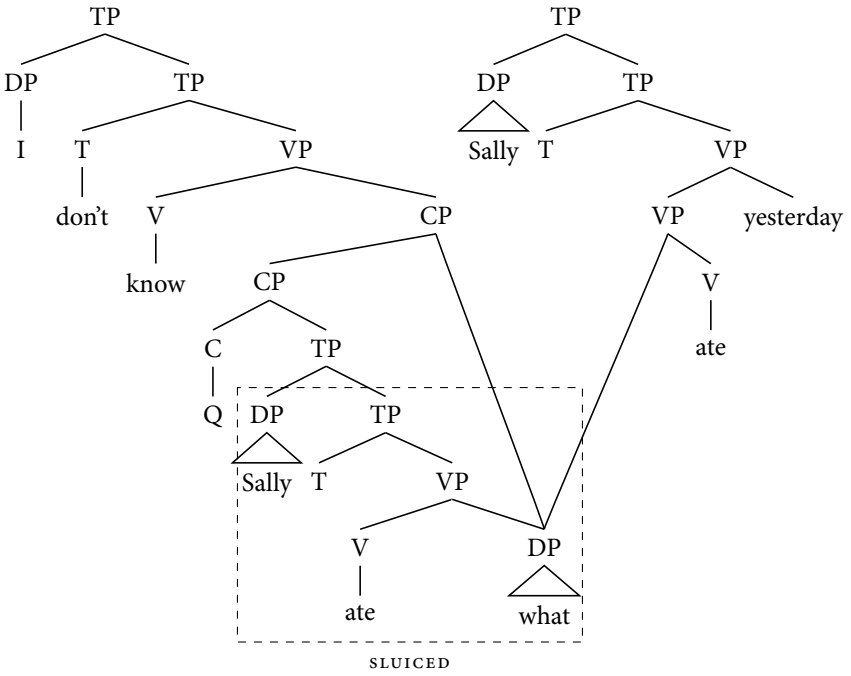
(34) $[\![\text{Q}]\!] = \lambda P.\lambda q.\exists \mathcal{F}.q = P(\mathcal{F})$

(35)



Putting the remerge theory of movement into (28), we get (36).

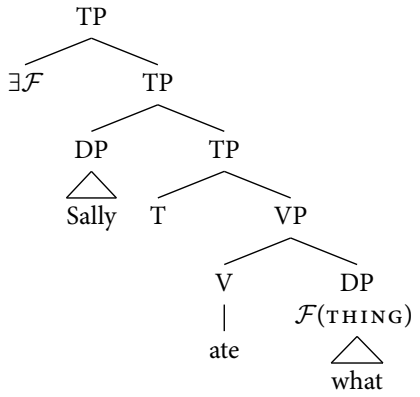
(36)



The variable that *what* provides is used by Q to make the complement of *know* a question. Its semantic contribution to the interrupting clause, then, is as the "indefinite" part of the material that makes the complement CP a question. In the host clause, however, the variable that *what* introduces is interpreted as existentially closed off. That seems to be a general process available to open variables in English.[1] Existentially closing the variable in *what* will make it equivalent semantically to *something*.

[1] See Heim (1982).

(37)



The host clause, then, uses *what* to make something that means "Sally ate something yesterday." All of this gives *what* a meaning that will allow it to have different effects depending on what kind of sentence it finds itself in. In (36), it finds itself in two different kinds of clauses, and therefore has different effects in each.

Why can't a wh-phrase be interpreted as "something" outside of amalgams? I suggest that this is because the morphology of a wh-phrase indicates that it is bound by a Q morpheme. (See Adger and Ramchand (2005) and Cable (2010) for precedents.)

(38) Every wh-phrase must be bound by a Q morpheme.

The intuition behind (38) is that the morphological form of the wh-phrase is the exponent of the Q morpheme. This will force every wh-phrase to get interpreted as part of a question. It is only when a wh-phrase can do this and also contribute its meaning to another sentence that we see it having the meaning of "something." That environment is what defines an amalgam.

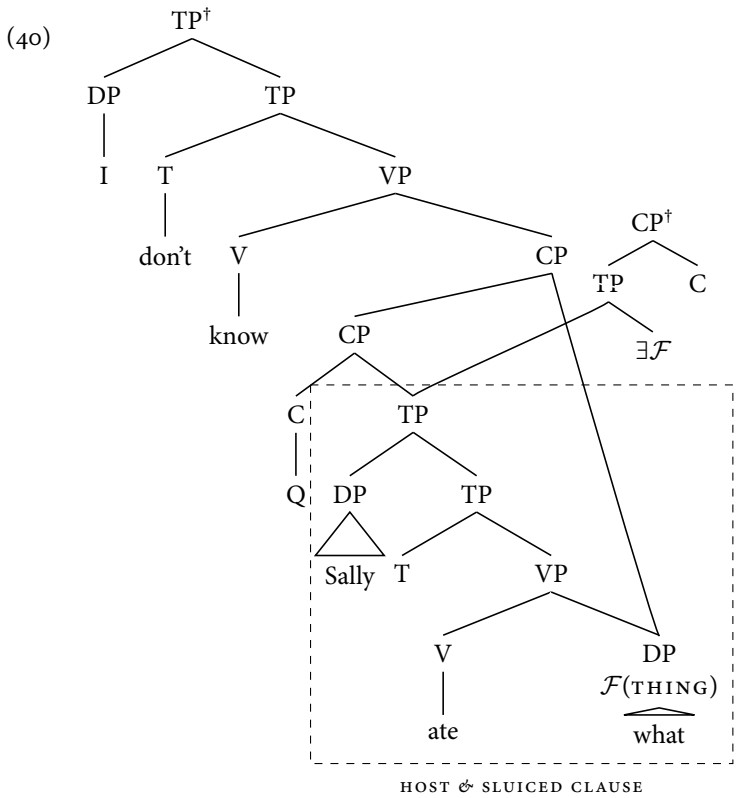
The sluicing in Andrew's amalgams is obligatory.

(39) a. Mary likes you'll never guess what.

b. * Mary likes you'll never guess what Mary likes.

To account for this, I'm going to follow the leading idea in Guimarães (2004), which is to use Antisymmetry and to adopt a representation like that in (40). That is, we assume that the clause that sluices is the hosting clause.

(40)



It's not just the wh-phrase that is shared between the two independent clauses; it is the TP that the wh-phrase resides in that is shared between the two clauses. The semantics works the same way that we developed earlier: the shared TP has the same meaning in each of the independent sentences, but the $\mathcal{F}$ variable gets bound by C in the question and by $\exists\mathcal{F}$ in the hosting clause. (40) makes (41) a violation of antisymmetry.

(41) * Sally ate I don't know what Sally ate.

If the TP of the embedded question is linearized according to both of its positions, then (41) violates Antisymmetry.

The proposal, then, is (42).

(42) An amalgam arises when an embedded clause is also the daughter of a root CP.

Andrews amalgams, then, will only arise in contexts where there is some way of avoiding the Antisymmetry violation that (41) incurs. That is what the ECP does. Let's see how.

Our linearization algorithm is repeated here.

(16) Where "<" is interpreted as "precedes" and $\mathscr{L}$ is the linearization algorithm applying to a root node containing XP, YP and X^0:

a. If y is dominated by YP and YP is the sister of X^0 dominating x, $\mathscr{L}$ sanctions x<y.

b. If y is dominated by YP and YP is the sister of XP, dominating x, and YP is in Specifier position, $\mathscr{L}$ sanctions y<x.

c. If y is dominated by YP, and YP is the sister of XP, dominating x, and YP is in adjunct position, $\mathscr{L}$ sanctions x<y.

(18) FORM STRING

Let X and Y be sisters in a phrase marker. If a terminal, x, in X is pronounced then pronounce y, a terminal in Y, so that it is adjacent to X.

There are two root nodes in (40), so $\mathscr{L}$ will produce the following two independent sets of ordering statements.

(43) $\mathscr{L}(TP^{\dagger})$ =

I < don't	don't < know	know < what	what < Q	Q < Sally	Sally < T	T < ate	ate < what
I < know	don't < what	know < Q	what < Sally	Q < T	Sally < ate	T < what	
I < what	don't < Q	know < Sally	what < T	Q < ate	Sally < what		
I < Q	don't < Sally	know < T	what < ate	Q < what			
I < Sally	don't < T	know < ate	what < what				
I < T	don't < ate						
I < ate							

(44) $\mathscr{L}(CP^{\dagger})$ =

C < Sally	Sally < T	T < ate	ate < what
C < T	Sally < ate	T < what	
C < Sally	Sally < what		
C < what			

When FORM STRING applies, its results must conform to the constraints in one or the other, or both, of these orderings. If two terminals are dominated by different root nodes, then both sets of orderings apply, otherwise the ones from the single root node that dominates them apply. Let's now walk through how this procedure produces a string.

Because *I* is pronounced, FORM STRING requires something in its sister to be pronounced adjacent to it, and (16) forces that to be *don't*. Because *don't* is pronounced, FORM STRING together with (16) requires *know* to immediately follow *don't*. This assembles the string in (45).

(45) I don't know

To this point, FORM STRING has only had to conform to (43), as the terms that have been put into the string are only dominated by $TP^{\dagger}$. The next step, however, will have to conform to both (43) and (44). That's because FORM STRING must now pick a terminal within the CP that is the sister to *know* to make it adjacent to *know*. One of those terminals is *what*, and *what* is dominated by both $TP^{\dagger}$ and $CP^{\dagger}$. *what* also has two sisters, and if FORM STRING applies to both of them, then a violation of antisymmetry threatens. As it happens, English independently requires that wh-phrases that occupy Specifier of CP be linearized according to that position. This will lead to a violation of FORM STRING if it applies to *ate*. But *ate* is among the terms that do not have to conform to FORM STRING; that is what the ECP says. For these reasons, then, FORM STRING will make *what* the item that is adjacent to *know*. Because *what* is pronounced, FORM STRING will require a terminal in the sister to *what* be pronounced adjacent to it. There are two sisters to *what*, of course, but only one of them is obligatorily subject to FORM STRING, and that is the CP. Because (43) requires that Q precede everything in that CP (except *what*), FORM STRING will make Q the next item. At this point we have (46).

(46) I don't know what Q

If FORM STRING applies to Q, it, together with (16) will require *Sally* to be the next terminal, because (43) and (44) both require *Sally* must precede everything else in the TP immediately dominating it (except *what*), and this TP is a sister to Q. But the TP containing *Sally* is also a sister to the root C^0, and the morpheme in this position, whatever it is, is pronounced as well. Thus FORM STRING together with (16) will require *Sally* to be adjacent to this C^0. *Sally* cannot be (right-)adjacent to both of these complementizers, however. But the Q morpheme is one of those items that is exempt from FORM STRING, and so *Sally* need not be pronounced adjacent to Q. Because *Sally* is pronounced, T must be pronounced adjacent to it, and *ate*, in turn, must be pronounced adjacent to T. This gives us the string in (47).

(47) C Sally T ate

These are the two substrings that need to be put together. FORM STRING does not allow these substrings to be broken up, so the two possibilities are arranging them in one of the two orders in (48).

(48) a. I don't know what Q C Sally T ate.
b. C Sally T ate I don't know what Q.

Consider the portion of these strings that is dominated by both $TP^{\dagger}$ and $CP^{\dagger}$: *Sally T ate what*. $\mathscr{L}(TP^{\dagger})$ requires that all of the terminals in this string follow *know*, and $\mathscr{L}(CP^{\dagger})$ requires that all of the terminals in this string follow C. Only (48a) satisfies both of these requirements. But now consider the relative order of *what* and *ate*. $TP^{\dagger}$ allows either order of these terms, but $CP^{\dagger}$ only permits *ate* < *what*. Because *what* is in both $TP^{\dagger}$ and $CP^{\dagger}$, it must satisfy both these constraints. Only (48b) does that. We have a problem.

The thesis of this paper is that the ECP describes the licensors for both ellipsis and traces, but that does not mean that ellipsis and traces are the same kinds of things. There is a simple difference between them that I suggest resolves this problem. When a phrase moves, that is when a phrase is in two positions, the ECP relaxes FORM STRING, thereby permitting the moved phrase to be pronounced without incurring a violation of Antisymmetry. But when the ECP licenses ellipsis, it's permitting the unpronunciation of the material that escapes FORM STRING. Let's build this difference into our system now. Imagine that ellipsis allows the set of precedence relations that $\mathscr{L}$ sanctions for the elided terminals to be ignored. This amounts to saying that the linearization procedure imposes no linearization requirements on material that is elided, and this seems a natural consequence for a process – ellipsis – that does not put that material in the string. Because the ECP is licensing ellipsis with respect to Q, this means that all the linearization statements in $TP^{\dagger}$ that involve the terminals in Q's sister can be ignored. This makes (43) equivalent to (49).

(49) $\mathscr{L}(TP^{\dagger}) =$

$$\left\{\begin{array}{llll} \text{I} < \text{don't} & \text{don't} < \text{know} & \text{know} < \text{what} & \text{what} < \text{Q} \\ \text{I} < \text{know} & \text{don't} < \text{what} & \text{know} < \text{Q} & \\ \text{I} < \text{what} & \text{don't} < \text{Q} & & \\ \text{I} < \text{Q} & & & \end{array}\right\}$$

The only string in (48) that satisfies both (49) and (44) is (48b), and that, at last, is the correct result.

References

Adger, David, and Gillian Ramchand. 2005. Merge and move: *Wh*-dependencies revisited. *Linguistic Inquiry* 36:161–193.

Aelbrecht, Lobke, and Liliane Haegeman. 2012. VP-ellipsis is not licensed by VP topicalization. *Linguistic Inquiry* 43:591–613.

Cable, Seth. 2010. *The grammar of Q: Q-particles, Wh-Movement and Pied-Piping*. Oxford University Press.

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

Guimarães, Maximiliano. 2004. Derivation and representation of syntactic amalgams. Doctoral Dissertation, University of Maryland.

Heim, Irene. 1982. The semantics of definite and indefinite Noun Phrases. Doctoral Dissertation, University of Massachusetts, Amherst.

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

Kluck, Marlies. 2011. *Sentence amalgamation*. Groningen: Landelijke Onderzoeckschool Taalwetenschap.

Lakoff, George. 1974. Syntactic amalgams. In *Papers from the 10th Regional Meeting of the Chicago Linguistic Society*, ed. Michael Galy, Robert Fox, and Anthony Bruck, 321–344.

Lobeck, Anne. 1992. Licensing and identification of ellipted categories in English. In *Proceedings of the Stuttgart Ellipsis Workshop*, ed. Steve Berman and Arild Hestvik. Stuttgart.

Nunes, Jairo. 1999. Linearization of chains and phonetic realization of chain links. In *Working minimalism*, ed. Samuel Epstein and Norbert Hornstein, 217–249. Cambridge, Massachusetts: MIT Press.

Riemsdijk, Henk van. 1998. Trees and scions – science and trees. In *Festweb page for Noam Chomsky*. MIT Press.

Riemsdijk, Henk van. 2000. Wh-prefixes, the case of wäsch in Swiss German. In *Naturally! linguistic studies in honour of Wolfgang Ulrich Dressler*, ed. Chris Schaner-Wolles, John Rennison, and Friedrich Neubarth. Torino: Rosenberg & Sellier.

Riemsdijk, Henk van. 2006. Towards a unified theory of *wh*- and non-*wh*-amalgams. In *In search of the essence of language science: festschrift for Professor Heizo Nakajima*, ed. Yubun Suzuki, Mizuho Keiso, and Ken-ichi Takami, 43–59. Tokyo: Hitsuji Shobo.

Rizzi, Luigi. 1990. *Relativized minimality*. Cambridge, Massachusetts: MIT Press.

Wilder, Chris. 1998. Transparent free relatives. *ZAS Working Papers in Linguistics* 10:191–199.