Studies in Chinese Linguistics
中國語文研究

Licensing Ellipsis

Kyle Johnson
University of Massachusetts at Amherst

Abstract

This paper provides an analysis of Andrews amalgams that builds on work by Marlies Kluck and Maximiliano Guimarães. It argues that Andrews amalgams involve bringing two independent sentences together by sharing a clause, where "sharing" is modeled by giving a phrase two mothers in a phrase marker. Andrews amalgams are licensed by Sluicing: they occur only when the shared clause can be sluiced. This, it is argued, shows us that the licensing conditions on ellipsis do not necessarily invoke the antecedence conditions usually attendant with ellipsis.

Keywords

Multidominant, amalgams, phrase structure, questions, sluicing, discontinuous dependencies, linearization, ellipsis

Studies in Chinese Linguistics, Volume 34, Number 2, 2013,

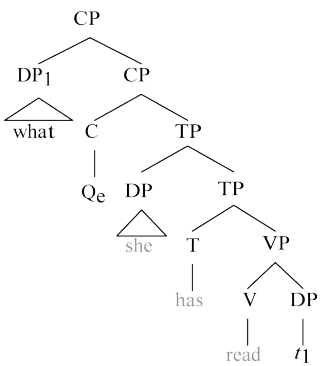
1. Introduction

In a run-of-the-mill ellipsis construction, some portion of a sentence is allowed to be unpronounced when its meaning can be recovered from spoken material in the context. There are two factors that determine when this can happen. First, the meaning-recovery must be successful: it must be possible for a hearer to determine what the meaning of the elided material is. This requires that there be an antecedent for the elided material that is both salient and fits suffciently well into the ellipsis site. I will call this factor the "antecedence condition" on ellipsis. The second factor concerns the syntax of the material to be elided. For material to be eligible to be elided it is thought that it must be a phrase whose sister is a head drawn from a small list of ellipsis licensors. I will call this the "licensing condition". Neither the licensing condition nor the antecedence condition have been figured out.

There is a possibility that the antecedence and licensing conditions are connected. Because the antecedence condition is different from all other known forms of surface anaphora, it presently seems likely that it cannot be made part of a general theory of anaphora. Instead, it looks like the antecedence condition for ellipsis belongs uniquely to ellipsis. If that is correct, then we need to build into ellipsis the particular requirements that constitute its unique antecedence condition. Because the licensing condition is responsible for defining where ellipsis is possible, it seems natural to build into the licensing condition the requirements that make up the antecedence condition.

A popular way of doing that was proposed in Merchant (2001). He suggested that the heads which license ellipsis have a denotation that requires their complement to meet the particular antecedence condition for ellipsis when that complement is unspoken. Merchant used the technology of features to express this connection. He suggested that a licensing head is one that can have an ellipsis feature (or, "e-feature"), which both specifies how to find an antecedent meaning for the head's complement and indicates that the complement can be pronounced as silence. For instance, in Sluicing – a kind of ellipsis that affiicts constituent questions – the complementizer that introduces a question can come with an e-feature, indicating that the clause that follows can be silent and that its meaning matches some previously spoken material in a way that satisfies the antecedence condition.

(1) She has read something, but I don't know



(The grayed text indicates that these words are silent.) In this way, then, we can glue the specific antecedence condition that holds of ellipsis to ellipsis — it is invoked always and only when an ellipsis is licensed.

This paper presents a counter-example to this analysis. It argues that a certain class of amalgams rely on the licensing condition for ellipsis but do not invoke the antecedence condition. If this is right, amalgams show us that the licensing condition for ellipsis and the antecedence condition on ellipsis are not linked. I believe this means we should endeavor to ground the antecedence condition on ellipsis in a more general account of anaphora, despite the problems this seems to face. That is, we should try to see the antecedence condition as something that is not specific to ellipsis, perhaps along the lines of Tancredi (1992) and Fox (2000).

In fact, what we will see is even stronger. We will see that the licensing condition on ellipsis does not even require that the relevant material be unspoken. In amalgams, that material is spoken.What the licensing condition on ellipsis does instead is allow the complement of the licensor to be linearized abnormally. I will propose that there is a condition, let's call it Contiguity, which is responsible for ensuring that phrases map onto contiguous strings. What the e-feature does is allow a lexical item to violate Contiguity. One way this can manifest itself is as ellipsis. Another are amalgams.

We start by looking at what amalgams are.

2. Andrews Amalgams

The particular sort of amalgam we will look at is what Kluck (2011) calls "Andrews amalgams", after Lakoff (1974) who first discussed them and credited their discovery to Avery Andrews. An example of an Andrews amalgam is (2).

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

The peculiarity of (2) 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 (2) a syntax that allows it to have a meaning parallel to (3).

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

Marlies Kluck and Maximiliano Guimarães wrote dissertations in the last few years that provide very complete analyses of Andrews amalgams, and my discussion will take their work as a starting point.[1] Two properties of Andrews amalgams that help steer us towards an account are, first, that the interrupting clause has the word order characteristic of root clauses. This is indicated, among other things, by the fact that they must have verb second word order in those languages, like Dutch,

[1] See Guimarães (2004) and Kluck (2011).

where that is required of root clauses.

(4) 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.'

(4) 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: 55, (14))

A companion fact is that most of the interrupting clause is not in the scope of material in the host clause. This can be appreciated by seeing that a pronoun in the interrupting clause cannot function as a variable bound by a quantifier in the host clause ((5a) is ungrammatical) and a name in the interrupting clause does not trigger disjoint reference effects with material in the host clause ((5b) is grammatical).

(5) 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:

c. * He$_1$ had been kissing many students that the professor$_1$ (himself) didn't remember.
(based on Kluck 2011: 97,(170) & 102, (195))

These facts lead to the conclusion that the interrupting clause and the clause it interrupts are, in some sense, two independent sentences. That also corresponds to the fact that the interpretation we aim for conjoins these two clauses, as (3) does.

On the other hand, Kluck shows that the interrupting clause is positioned syntactically within the host clause in just the places that it should be if it were fulfilling the semantic role that it seems to fulfill. In (2), 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.

(6) 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.

(7) 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.

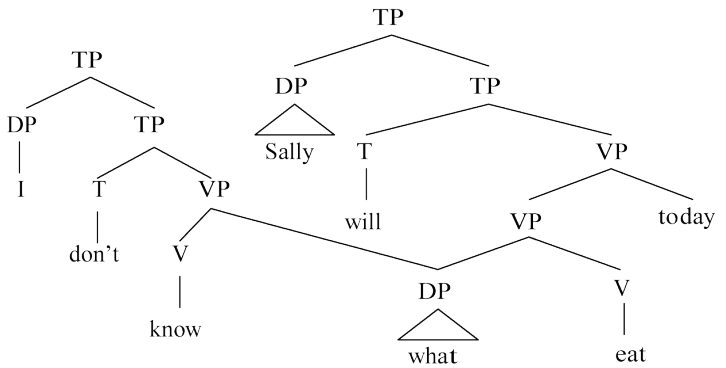
And when an interrupting clause seems to play the role of an adjective, it can only appear in the positions that adjectives can.

(8) 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 which Riemsdijk (1998) suggests is 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. I'll adopt that view as well. To see how this works, let's start with the overly simplified representation in (9).

(9)



In (9), *what* is simultaneously the object of *know* and *ate*. Otherwise, these are two independent sentences: *Sally will eat what today,* and *I don't know what*.

What's needed now is a linearization algorithm that correctly translates the phrase-marker in (9) into the string in (2). Rather than speaking these two sentences independently, they are joined by *what* and this should have the consequence of forcing these two sentences to form a single string. What's novel about (9), of

course, is its crossing branches, and so to the usual linearization algorithm we will need to add a way of evaluating phrases that have two mothers. I'll begin by sketching how a linearization algorithm works in more standard cases, and then turn to engineering a way for it to apply in the desired way to (9).

Let a linearization algorithm be a relation from phrase markers to strings. We can understand it to be part of the interface condition that translates a syntactic representation into a phonological representation, then. It is the component of that interface condition that orders the words. There are a variety of linearization algorithms available—perhaps the best known is Kayne(1994)'s "Linear Correspondence Algorithm". Most of them, including Kayne's, will have to be modified slightly to allow multidominant representations like (9). To do that fully requires more space than I have here, so instead I will merely sketch an outline of how something like Kayne's algorithm could be modified.

Assume that phrase markers X^0s and XPs, where "X" ranges over the morphosyntactic categories supported by a language. X^0 will dominate just a terminal matching the morphosyntactic category of X^0, and XP will dominate a string of terminals, including one dominated by X^0, its "head". The linearization algorithm operates on the symbols X^0 and XP and forms a set of ordered pairs, x<y, of the terminals they dominate. For English, the linearization algorithm operates with the following result.

(10) 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, $\mathscr{L}$ can produce x<y.
   b. If y is dominated by YP and YP is the sister of XP, dominating x, and in Specifier position, $\mathscr{L}$ can produce y<x.
   c. If y is dominated by YP, and YP is the sister of XP, dominating x, and in adjunct position, $\mathscr{L}$ can produce x<y.

This does nothing more than devolve the question of how phrase markers are linearized to the questions "what is in Specifier position" and "what is in adjunct position"? A real linearization algorithm would answer these questions and provide an explanation thereby for word order typology. Kayne's does so by defining Specifier as a phrase that asymmetrically c-commands another phrase or head, and defines adjunct as a Specifier past which another phrase has moved. For our purposes we can rely on the standard conventions presently used in the literature for Specifier and Adjunct, roughly those of Chomsky (1981).

Along with (10), which merely indicates how syntactic structures will be represented by precedence relations, we'll need something to ensure that the linearization algorithm maps all of the material in a phrase marker into the string. In Kayne's linearization algorithm, this is expressed as a constraint on the syntax-to-string mapping that gives every syntactic position in a phrase marker a linear

position in the resulting string. One of the consequences of Kayne's condition is that multidominant phrase markers are blocked. Since this is a consequence we should avoid, I'll reformulate his constraint as one that ensure that every terminal in a phrase marker is given a linear position in the resulting string. Kayne calls his condition "Totality," and I'll use that label as well.

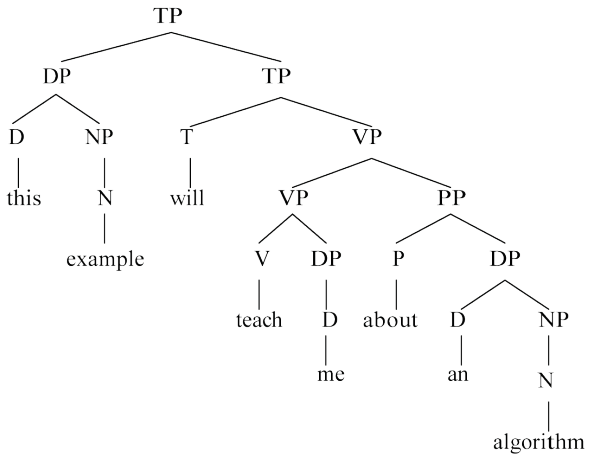
(11) Totality

If $\mathscr{R}$ is a root node, then $\mathscr{L}$ must put every terminal in $\mathscr{R}$ in an ordered pair with every other terminal in $\mathscr{R}$.

We'll derive Totality in a moment.

To get a feel for how this works, let's run through how this linearization algorithm produces a string from (12).

(12)



The terminals in this sentence are *this*, *example*, *will*, *teach*, *me*, *about*, *an* and *algorithm*. Totality requires each of these terminals be ordered with all the others: we'll need a set of 28 ordered pairs. By virtue of (10a), the ordered pairs in (13) are sanctioned.

(13)

$$\left\{\begin{array}{llllll} \text{this} < \text{example} & \text{will} < \text{teach} & \text{teach} < \text{me} & \text{about} < \text{an} & \text{an} < \text{algorithm} \\ & \text{will} < \text{me} & & \text{about} < \text{algorithm} & \\ & \text{will} < \text{about} & & & \\ & \text{will} < \text{an} & & & \\ & \text{will} < \text{algorithm} & & & \end{array}\right\}$$

To this can be added the ordered pairs in (14), which are sanctioned by (10b).

(14)
$$\left\{\begin{array}{ll} \text{this} < \text{will} & \text{example} < \text{will} \\ \text{this} < \text{teach} & \text{example} < \text{teach} \\ \text{this} < \text{me} & \text{example} < \text{me} \\ \text{this} < \text{about} & \text{example} < \text{about} \\ \text{this} < \text{an} & \text{example} < \text{an} \\ \text{this} < \text{algorithm} & \text{example} < \text{algorithm} \end{array}\right\}$$

And, finally, (10c) allows for the ordered pairs in (15).

(15)
$$\left\{\begin{array}{ll} \text{teach} < \text{about} & \text{me} < \text{about} \\ \text{teach} < \text{an} & \text{me} < \text{an} \\ \text{teach} < \text{algorithm} & \text{me} < \text{algorithm} \end{array}\right\}$$

For Totality to be satisfied, the union of all three sets is required.

(16) (13) ∪ (14) ∪ (15) =

$$\left\{\begin{array}{lllll} \text{this} < \text{example} & \text{example} < \text{will} & \text{will} < \text{teach} & \text{teach} < \text{me} & \text{me} < \text{about} \\ \text{this} < \text{will} & \text{example} < \text{teach} & \text{will} < \text{me} & \text{teach} < \text{about} & \text{me} < \text{an} \\ \text{this} < \text{teach} & \text{example} < \text{me} & \text{will} < \text{about} & \text{teach} < \text{an} & \text{me} < \text{algorithm} \\ \text{this} < \text{me} & \text{example} < \text{about} & \text{will} < \text{an} & \text{teach} < \text{algorithm} & \\ \text{this} < \text{about} & \text{example} < \text{an} & \text{will} < \text{algorithm} & \text{about} < \text{an} & \\ \text{this} < \text{an} & \text{example} < \text{algorithm} & \text{about} < \text{algorithm} & & \\ \text{this} < \text{algorithm} & \text{an} < \text{algorithm} & & & \end{array}\right\}$$

(16) corresponds to the string in (17).

(17) this example will teach me about an algorithm

Something of this kind functions presently as the linearization algorithm responsible for spelling out phrase markers as strings of terminals. Strictly speaking, of course, the formalism I've adopted doesn't produce strings. It produces sets of ordered pairs from which strings can be calculated. In modifying this algorithm so that it applies correctly to multidominant representations like those in (9), I will make that calculation from sets of ordered pairs to strings explicit.

We can see why that is necessary by considering how the linearization algorithm, as it stands, would apply to (9). Because (9) has two root nodes, Totality will apply to each of them independently, and drive the statements in (10) to produce the sets in (18).

(18) a.
$$\left\{\begin{array}{lll} \text{I} < \text{don't} & \text{don't} < \text{know} & \text{know} < \text{what} \\ \text{I} < \text{know} & \text{don't} < \text{what} & \\ \text{I} < \text{what} & & \end{array}\right\}$$

b.

$$\left\{\begin{array}{llll} \text{Sally} < \text{will} & \text{will} < \text{eat} & \text{eat} < \text{what} & \text{what} < \text{today} \\ \text{Sally} < \text{eat} & \text{will} < \text{what} & \text{eat} < \text{today} & \\ \text{Sally} < \text{what} & \text{will} < \text{today} & & \\ \text{Sally} < \text{today} & & & \end{array}\right\}$$

Unlike the relationship between (16) and (17), it is not obvious what string (18) corresponds to. If sentences can have two root nodes, as does (9), then we'll need a way of calculating the strings they produce that goes beyond what the standard linearization algorithm supplies.

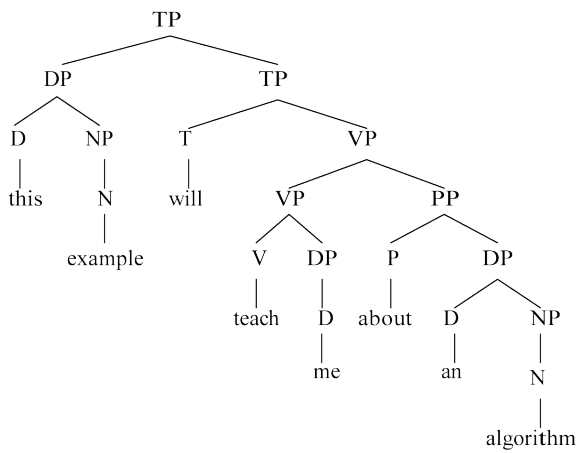
I suggest that strings are formed from phrase markers by the operation in (19).

(19) FORM STRING

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

Understand the ordered pairs that $\mathscr{L}$ creates to be a well-formedness constraint on the output of FORM STRING. These ordered pairs, then, will determine which terminal in X must be made adjacent to which terminal in Y. FORM STRING will not only, with the aid of $\mathscr{L}$, produce a string, it will enforce Totality as well. What it says is that if one terminal in a phrase marker is pronounced, then all of them must be. We can demonstrate this by looking at how it applies to (12), repeated here.

(12)



Because the two determiners are sisters to a phrase that has just one noun in them, FORM STRING will require those determiners to be adjacent to those nouns. Similarly, because the verb is a sister to a phrase that has just one terminal in it, FORM STRING will require them to be adjacent. FORM STRING will therefore require the string that is produced from (12) to contain the substrings

in (20), as these are the ways of making these terms adjacent that conform to the requirements imposed by $\mathscr{L}$ (see (16)).

(20) *this example*, *teach me*, and *an algorithm*

The preposition is a sister to the DP made up of *an* and *algorithm*, and so FORM STRING puts *about* adjacent to one these two. Because $\mathscr{L}$ requires that *about* precede these two words, this can be achieved only by forming the substring *about an algorithm*. Because the VP containing *teach me* is a sister to the PP mapped onto *about an algorithm*, FORM STRING requires one of the words in each of these strings to be adjacent. Because $\mathscr{L}$ requires the terminals in VP to all precede the terminals in PP, this gives us *teach me about an algorithm*. Because T is a sister to the VP that contains the terminals in *teach me an algorithm*, FORM STRING requires them to be adjacent, and this gives us *will teach me an algorithm*. And, finally, the subject DP is a sister to TP, and all of its terminals must, $\mathscr{L}$ requires, precede everything in that TP, so FORM STRING produces *this example will teach me about an algorithm*.

In general, then, FORM STRING forces a terminal in a phrase marker to be in the string that the linearization algorithm produces, if it has a sister. Because every terminal has a sister, every terminal will be put into the string that results from linearizing a phrase marker. This is Totality.

Let's now see how FORM STRING will apply to (9). It will force this two-rooted phrase marker to map onto one contiguous string because the two TPs share a node, and the terminals in that node will therefore have to be adjacent to material in each of the TPs. In fact, when combined with the constraints imposed by $\mathscr{L}$, the shared material, *what*, will have to be simultaneously right-adjacent to *know* and *ate*. That is impossible, of course, and so we can see that multidominant structures force a relaxation of the normal linearization's requirements. The punchline of this paper is what that relaxation is. At this point, however, let us imagine that we could find a way of letting FORM STRING avoid making *ate* and *what* adjacent. Running the linearization algorithm—that is, FORM STRING constrained by $\mathscr{L}$—will produce just the string we want from (9). In particular, FORM STRING will require *know* and *what* to be adjacent, and it will force all the rest of the TP containing *know* and *what* to form a contiguous string as well. Consider, by way of illustration, the strings in (21).

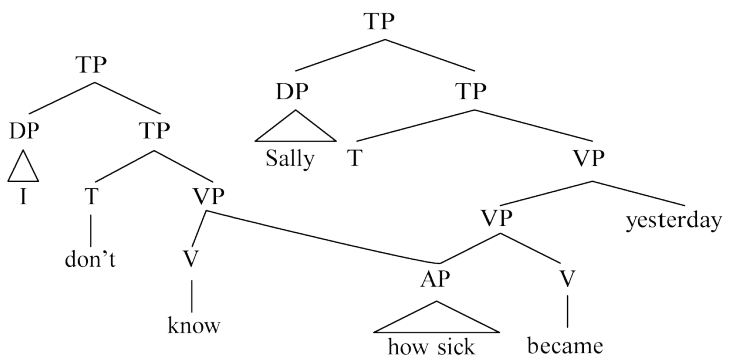
(21) a. Sally will eat I don't know what today.
b. I don't know Sally will eat what today.
c. Sally will I don't know eat what today.
d. Sally will eat what I don't know today.
e. Sally will eat what know I don't today.

(21a) is the desired linearization, and it puts heads and Specifiers first and obeys FORM STRING, except with respect to *eat* and *what*, which is the violation we are sanctioning. Note in particular that because *what* is in the VP headed by *know*, this

VP and its sister (i.e., *today*) must satisfy FORM STRING. They do because *what* is adjacent to *today*. All the other strings in (21) violate FORM STRING, and also, sometimes, the ordering constraints imposed by $\mathscr{L}$. (21b) violates FORM STRING by separating *know* and *what*. (21c) violates FORM STRING by separating *know* and *what* as well as *will* and *eat what today*. (21d) also violates FORM STRING twice, by separating *know* and *what* as well as *eat what* and *today*. Moreover, it fails to put all heads first, since *know* does not precede *what*. And finally (21e) violates FORM STRING twice, by separating *know what* from *don't* and *eat what* from *today*. And like (21d), (21e) also fails to put all heads first, as neither *don't* precedes *know what* nor does *know* precede *what*.

With these assumptions, then, we can ensure that the interrupting clause is linearized in the right spot if we let the wh-phrase it contains also be part of the hosting clause. FORM STRING will ensure that the rest of the interrupting clause lines up adjacent to that shared wh-phrase. Further, relaxing FORM STRING for the term in the hosting clause that is the sister to the shared wh-phrase will permit the interrupting clause to interrupt the host clause at that position. (21) rehearses how that works when the wh-phrase is a direct object, but the same ingredients work for all the other examples. So, for instance, the example in (8c) would get the representation in (22) and if FORM STRING were relaxed to let *become* and *how sick* be non-adjacent, the correct string (=(8ci)) will be manufactured.

(22)



A complete account of how the interrupting clauses get positioned, then, needs only an explanation for why FORM STRING is relaxed in the particular way that it is. We come back to that shortly, but for now it's sufficient to see that the multidominant representation in (9) provides the means for such an account.

Adopting (9), then, provides a way of accounting for where the interrupting clause gets linearized. There are other facts which can be construed as evidence for (9) as well. For one thing, the shared material, unlike the rest of the interrupting clause, does seem to be within the scope of stuff in the hosting clause. Unlike what we saw in (5), a quantifier in the hosting clause can bind a pronoun in the shared material ((23a) is grammatical), and a name in the shared material does

invoke disjoint reference effects with material in the hosting clause ((23b) is ungrammatical).

(23) a. Almost every student$_1$ kissed [you can't imagine how many of his$_1$ classmates].
(based on Kluck 2011: 97, (169))
b. * He$_1$ kissed [I don't know how many of the professor$_1$'s students].

That follows from (9) because (9) puts the wh-phrase inside the hosting clause, and therefore within the scope of the material it contains. Indeed, as a reviewer points out, there are examples in which the subjects of both host and interrupting clause can bind a term in the shared material, thereby forcing those subjects to corefer even in apparent violation of the disjoint reference effect on names. (24) is such an example.

(24) John likes John doesn't know how many pictures of himself.

The reason (9) is over-simplified is that it fails to capture the fact that the shared material – the wh-phrase – seems to make different semantic contributions to the hosting clause and the interrupting clause. In the interrupting clause, the wh-phrase is part of a sluice. The *what* in (9) 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.

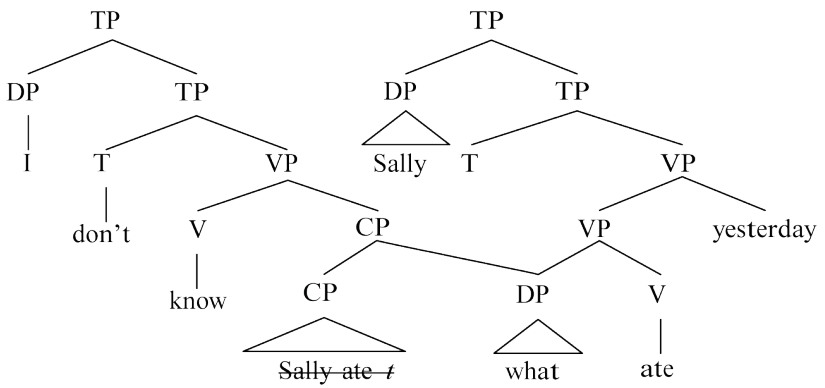
(25) a. Er will jemandem schmeicheln, aber sie wissen nicht wem.
he will somone-Dat flatter, but they know not who-Dat
'They will flatter someone, but they don't know who.'
b. * Er will jemandem schmeicheln, aber sie wissen nicht wen.
he will somone-Dat flatter, but they know not who-Acc
'They will flatter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

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

(26) a. Bea wollte [ich weiss nicht mehr wem] schmeicheln.
Bea wanted [I know not anymore who-Dat] flatter
'Bea wanted to flatter I don't know who.'
b. * Bea wollte [ich weiss nicht mehr wen] schmeicheln.
Bea wanted [I know not anymore who-Acc] flatter
'Bea wanted to flatter I don't know who.'
(Kluck 2011: 184, (83))

Putting the sluice into (9), produces something like (27).

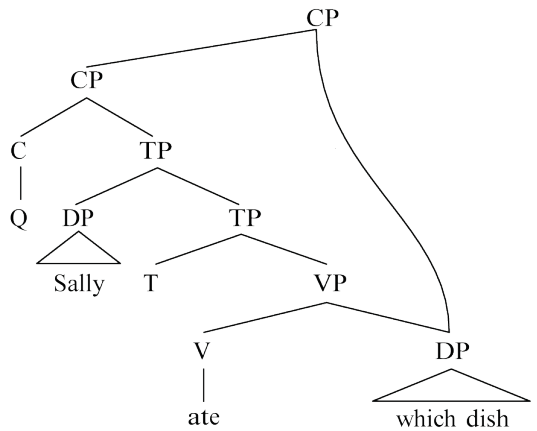
(27)



We want the hosting clause to mean something like "Sally ate something." So *what* must be interpreted as "something" in the host clause. But in the interrupting clause, that very same *what* must be interpreted as, well, "what". We need to understand how *what* can be interpreted in these two ways in this structure.[2] That means that the shared wh-phrase gets a different interpretation in the hosting and interrupting clauses. What we need to do is understand how a wh-phrase is capable of getting different interpretations in amalgams.

Let's start by considering more narrowly the interpretation a wh-phrase gets when it moves in a constituent question. If we adopt the remerge theory of movement, a constituent question gets a representation like that in (28).

(28)



The wh-phrase is in two positions, though pronounced in English only in the higher of these. We want it to introduce a variable in the lower position that is

[2] See Kluck (2011) for an alternative approach.

bound by something near the higher position. In many languages, we can see that the variable that a wh-phrase invokes is bound by a functional head. In Japanese, for instance, where wh-phrases are pronounced in their lower position, we can see that what binds the variable introduced by that wh-phrase depends on what nearby functional heads there are. In (29), for example, *dono gakusei* ('which student(s)') functions as the variable-part of a constituent question.

(29) (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?'

That is because the question complementizer *ka* binds that variable. In (30), however, *dono gakusei* is a variable bound by the universal quantifier *mo*.

(30) [dono gakusi-ga syootaisita sensi]-mo odotta.
[which student-Nom invited teacher]-MO dance
'The teacher that every student invited danced.'
(Shimoyama 2006: 139, (1b))

A nice way of unifying English with these languages is to let the silent Q complementizer bind the variable introduced by *which dish* in (28). This means that *which dish* cannot be semantically interpreted in its higher position, and the denotation that wh-phrases have in English is, like the interpretation of *dono*-phrases in Japanese, just an open variable which gets closed by some local operator.[3]

There is a difference between English and Japanese style languages, and that's that the wh-phrases of English can only show up when they are bound by the Q morpheme. Unlike Japanese, wh-phrases in English are not capable of getting their quantificational force from some other operator. We can describe this difference between English and Japanese with (31).

(31) The Question morpheme in English must be pronounced as the determiner of the DP it binds.

Whether (31) holds in a language or not might correlate with whether the functional heads which act as binders are overtly pronounced.[4] They are in Japanese, but they are not in English. If so, (31) might be seen as a reflex of the more general Principle of Full Interpretation.[5]

(32) Principle of Full Interpretation
Every terminal in a phrase marker must have a morphological reflex.

In English, the Q morpheme has its morphological reflex in the determiner. I'll

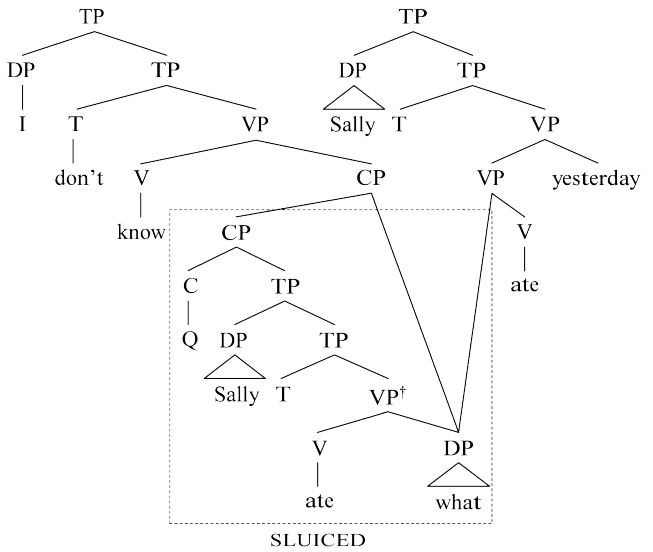
---

[3] See Hagstrom (1998, 2000), Kishimoto (2005) and Cable (2007) among others.
[4] See Cheng (1991), but also Cable (2010) for some qualifications.
[5] This is inspired by Chomsky (2000).

follow Adger and Ramchand (2005), Kratzer (2005) and Cable (2010) and take this to happen through the agency of Agreement.

I will assume, then, that the wh-phrases in English introduce a kind of restricted variable that is bound by the Q morpheme to produce a question. There are several ways of implementing this idea, and I won't choose among them here. (See Reinhart (1997, 1998), Hagstrom (1998), Beck (2006) and Cable (2007) for some choices.) Putting together the multidominant approaches to Wh-movement that (28) illustrates with the multidominant analysis of amalgams yields the phrase structure in (33).

(33)



In (33), *what* is in three places. I'll step through these positions and describe how it is evaluated in each.

(34) Daughter of VP†
   a. Not pronounced here because (single) wh-phrases must be pronounced closer to Q (that is: English allows only the higher copy of a moved wh-phrase to be pronounced.)
   b. Semantically interpreted as a variable bound by Q that functions as the object of *ate*.

(35) Daughter of CP
   a. Pronounced here because English requires the Specifier of a question CP to be phonologically filled by a wh-phrase.
   b. Not semantically interpreted here.

(36) Daughter of VP
   a. Pronounced here for whatever reason the objects of transitive verbs must be overt in English.
   b. Semantically interpreted as an open variable that functions as the object of *ate*.

The open 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 term that generates the variable from which a question denotation is derived.

In the host clause, however, something different happens. Here, the wh-phrase cannot be interpreted as part of a constituent question, as there is no Q morpheme in the host clause. Instead, we want it to be interpreted as a simple existential. I will assume that the restricted variable that wh-phrases introduce can be bound off by an existential quantifier, thereby producing the correct interpretation. The exact method for doing this will depend on how the relationship between the Q morpheme and the wh-phrase is expressed, since this will fix what the denotation of the wh-phrase is. I won't do that here; there are several possibilities.

What this analysis claims, then, is that a wh-phrase is a kind of open variable which can be used either to form a question or to form a declarative existential sentence. In the case of Andrews amalgams, both of these possibilities are exploited: the question meaning forms the sluice of the interrupting clause, and the declarative existential meaning is delivered by the host clause. If this is correct, it must be ensured that the existential meaning is not produced by wh-phrases when they are not in an amalgam. It is not the case that a wh-in-situ, for instance, can be interpreted as anything but an interrogative phrase. Sentences like (37) do not have an interpretation like that produced for (38).

(37) What did Sally give to whom?

(38) What did Sally give to someone?

What makes amalgams different from cases like (37) is that the wh-phrase in the amalgam is in two different sentences, only one of which is as an interrogative.[6] To ensure that wh-phrases are not capable of being used as plain existentials outside amalgams, we should adopt the view that the interrogative determiners — *which*, *how*, *when*, *who*, etc. — are morphologically dependent on being in the scope of the Q morpheme. That is, we should strengthen (31) so that it not only requires the

---

[6] An anonymous reviewer points out that this system doesn't clearly apply to situations where the host clause in an Andrews amalgam can be an interrogative, as in (i).

(i) Who ate I won't tell you which sundae?

I'm not certain of the grammaticality status of (i), but if it has an interpretation, that interpretation is something like "who ate I won't tell you which sundae that person ate." More work is needed to understand what is going on in this example.

Q morpheme's exponent to be the wh-determiner, but allows wh-determiners only as the exponent of a Q morpheme.

(39) A Q morpheme is pronounced as the wh-determiner on the phrase(s) it binds, and a wh-determiner only arises when it is bound by a Q morpheme.

This will force *whom* in (37) to be bound by a Q morpheme, and since there is only one Q morpheme in this sentence, *whom* will be construed as part of the question formed by that Q morpheme. In (33), by contrast, *what* is bound by the Q morpheme in the interrupting clause, and necessarily therefore understood as part of the question that Q forms, but simultaneously able to get bound by a different operator, the existential quantifier, in the host clause. (39), then, forces all wh-phrases to be interpreted as part of a constituent question in English, and allows them to get another interpretation only when they are simultaneously part of an independent declarative sentence. That can only happen in multidominant phrase-markers. That will narrow the range of places where wh-phrases can be interpreted as non-interrogative existentials to, hopefully, just amalgams.

This is the analysis of Andrews amalgams I'll adopt. An Andrews amalgam puts two independent sentences together by letting them syntactically share material. This allows the wh-phrase that is always the shared thing in an Andrews amalgam to get two different interpretations: as part of the sluiced question in the interrupting clause and as an existential in the hosting clause. The standard linearization algorithm will put the hosting and interrupting clauses together into a single string in the right way, if FORM STRING is relaxed for the shared material in the hosting clause.

3. Sluicing

There are two problems with the account just sketched. First, it requires a violation of FORM STRING in a strategic place for the correct strings to be manufactured, but offers no explanation for that violation or its placement. Second, it does not force the interrupting clause to have a sluice in it which, as we will see, is another requirement of the construction. This section offers a solution to the second of these problems that has the consequences for licensing ellipses that were advertised at the beginning of the paper. It also provides a suggestion about the first problem.

An Andrews amalgam must have a sluiced clause in it. If (33) is pronounced with the sluiced material restored, the result is ungrammatical.[7]

---

[7] This predicts that Andrews amalgams should not be found in languages without Sluicing. A reviewer notes that Andrews amalgams are not found in Chinese.

(i) * Shali jintian dagai yao chi wo bu zhidao shi shenme.
Shali today probably will eat I not know be what
'Shali will probably eat I don't know what today'

Because Chinese is a wh-in-situ language, it is predicted to not have Sluicing, and so falls in line

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

Moreover, If neither of the sentences that are brought together in an Andrews amalgam have material that can be sluiced, the result is ungrammatical. An Andrews amalgam only happens when a sluice can, and does, occur.

To see that Andrews amalgams are licensed only in environments where Sluicing happens, we need to know first what those environments are. Sluicing elides the TP part of questions, and nothing else. It distinguishes TPs embedded in indirect questions from TPs embedded in free relatives, for instance. That is responsible for the contrast in (41).

(41) a. Mary ate something, and I'll tell you what ~~Mary ate~~.
b. * Mary danced some time last week, and I danced when ~~Mary danced~~.

As we've seen, interrupting clauses can contain indirect questions from which a TP has been dropped. But if the interrupting clause contains a free relative, then the TP inside it cannot elide, and that prevents it from being grammatical.

(42) * Mary danced I'm pleased when.

This is perfectly general. Andrews amalgams only arise when the host clause matches a clause in the interrupting clause that has sluiced. What we're looking for is something that will allow Andrews amalgams only where sluices are permitted.

Deriving this property is one of the central goals of Guimarães (2004). His proposal has two parts. He suggests a structure, slightly different from (33), that will ensure that the standard linearization schemes fail. Then he devises a linearization scheme purpose-built for that structure which has the right outcome. I will adopt his structure, but suggest a different way of resolving the linearization problem it creates.

Guimarães's structure aims to explain why (40) is ungrammatical. What is needed is something that prevents the sluiced material from being spoken. Note that in Andrews amalgams, the sluiced material is part of the same string which holds its antecedent. We could use this feature of amalgams to force sluicing if we could find a way of preventing the sluiced material from being pronounced in the same string as its antecedent. One way of achieving that goal in frameworks which allow multidominant phrase markers is to let the material which can only be pronounced once be the <u>same</u> material given two positions. Under normal circumstances, when a phrase has two positions in a phrase marker, the linearization algorithm cannot let

---

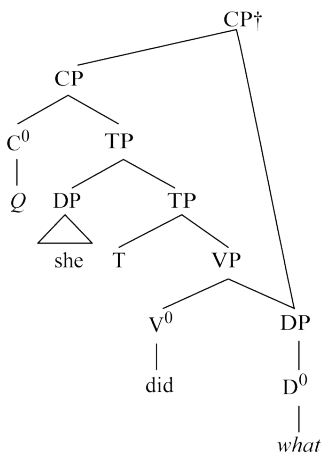
with this prediction. On the other hand, the reviewer points out that Chinese does have examples that look like Sluicing:

(ii) Shali jintian dagai yao chi shenme, dan wo bu zhidao shi shenme.
Shali today probably will eat what but I not know be what
'Shali will probably eat something today, but I do not know what.'

More needs to be discovered about the language variation of amalgams.

the material associated with that phrase show up twice in the string. For instance, consider how the structure in (43) gets linearized.

(43)



We have adopted a remerge theory of wh-movement, and this example involves the structure this theory would assign to the indirect question: *what she did*. We want to design the linearization algorithm so that it allows (43) to produce either of the strings in (44), but prevent it from generating (45).

(44) a. *what she did*
b. *she did what*

(45) *what she did what*

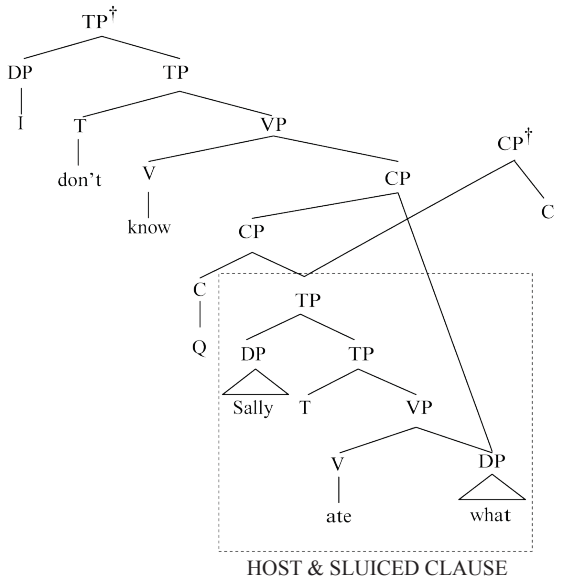
(45) is prevented by Kayne's (1994) Antisymmetry.

(46) Antisymmetry
If α precedes β in a linearization, then β cannot precede α in that linearization.

In (45), *what* both precedes and follows *she*, among other things, and this is what Antisymmetry forbids. In general, then, there must be a way for structures like (43) to avoid a violation of Antisymmetry, and instead offer one of the linearizations in (44). The focus of this paper is not movement, and so I will not explore how that is achieved. (Note that the good linearizations in (44) both violate FORM STRING, and so movement must somehow permit a departure from its normal operation.) All we need see here is that Antisymmetry plays a crucial role in preventing a phrase that has two mothers from being mapped into two positions in a string.

Guimarães exploits this effect of Antisymmetry to derive the ungrammaticality of (40). He suggests that the host clause and the clause that is sluiced are the same clause in two positions in the phrase marker. We can incorporate his suggestion into the analysis here with (47).

(47)



In (47), it is not the wh-phrase that is shared, but instead a TP containing that wh-phrase which is shared. This shared TP combines with the Complementizer heading the CP which *know* selects, and it combines with a Complementizer to form CP†, the host clause. The CP that *know* selects is headed by a Q morpheme, and so the wh-phrase within the shared TP gets the interrogative interpretation as described in the previous section. The CP which forms the host clause doesn't have a Q morpheme, and so the wh-phrase in the shared TP gets interpreted as an existential in this CP. The semantics sketched in the previous section, then, works the same for this representation.

So also do the effects of linearization algorithm: FORM STRING and $\mathscr{L}$. To linearize TP†, FORM STRING produces the sequence in (48).

(48) I don't know what Sally ate.

Movement is responsible for putting *what* in two positions, and this allows *what* to be linearized in the way it would if it only occupied the higher of these positions. Sluicing removes the embedded TP portion from the string, and this produces (49).

(49) I don't know what.

To linearize CP†, FORM STRING produces (50).

(50) Sally ate what.

But, again, movement has put *what* in two positions, and in English this requires that it be linearized in the way indicated in (51).

(51) what Sally ate

Here, at last, is the reason why the shared wh-phrase in an amalgam is allowed to violate FORM STRING with its sister in the hosting clause but not with its sister in the interrupting clause. The shared wh-phrase has moved away from its sister in the hosting clause, and movement overcomes violations of FORM STRING. Moreover, the wh-phrase has moved into a position that FORM STRING requires be adjacent to the verb of the interrupting clause.

Putting (51) together with (49) in a way that follows FORM STRING produces the sequence in (52).

(52) Sally ate I don't know what.

That's the correction outcome.

Consider, now, what would happen if Sluicing had not applied. Under that scenario, (50) would not have to be combined with (49) but with (48) instead. Relaxing Contiguity just enough to allow *ate* and *what* to be non-adjacent would produce the string in (53).

(53) Sally ate I don't know what Sally ate.

This violates Antisymmetry: *Sally* both precedes and follows *know* (among other things)and so does *ate*. The representation in (47), then, preserves everything in the account given in the previous section but also derives the obligatoriness of Sluicing.

Except that what's happening in (49) isn't Sluicing. There is no ellipsis in the traditional sense here. The TP that forms the embedded question is not unpronounced, with its semantic content recovered by way of the antecedent condition. Instead, the embedded TP is not being linearized in the normal way. In the terms of this paper, what is happening in (49) is that the TP is allowed to not be adjacent to the ellipsis licensor. That is, the head that normally licenses Sluicing is allowing to violate FORM STRING in this example. We should conclude that the thing which licenses Sluicing, and so ellipsis in general, is something that also allows a violation of FORM STRING.

One consequence of this conclusion is the one made at the beginning of this paper: the licensing condition and the antecedence condition should not be directly linked. In an Andrews amalgam, the licenser is obviously not requiring that its complement be subject to the antecedence condition. Its complement is neither eliding nor recovering its meaning from an antecedent. Instead, its complement is being pronounced in a position that is permitted by a selective relaxation of FORM STRING.

We can also learn something about what the licenser of an ellipsis does. It must allow either ellipsis or an unorthodox linearization. It must relax a constraint that prevents either of these outcomes. I suggest that it is FORM STRING, repeated

here, which an ellipsis licenser relaxes.

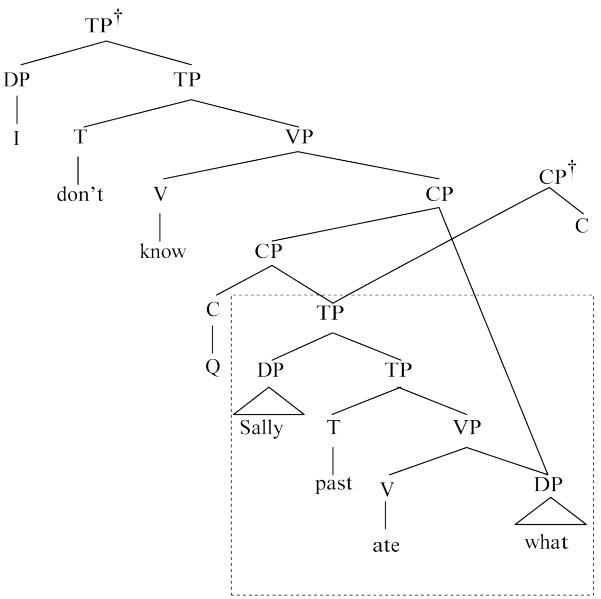
(19) FORM STRING
Let X and Y be sisters in a phrase marker. If a terminal, x, in X is pronounced then pronounce a terminal in Y adjacent to x.

FORM STRING, then, must also be responsible for preventing ellipsis, and indeed, since FORM STRING enforces Totality, that is exactly what it does. So here's the proposal.

(54) A lexical item that licenses ellipsis is one that FORM STRING can ignore.

So with everything finally in place, consider now how FORM STRING will evaluate the Andrews amalgam in (47).

(47)



Movement causes *what* to get linearized so that it is not adjacent to *ate*, but instead is the first phrase in the embedded TP. It licenses somehow, then, what would otherwise be a violation of FORM STRING. The embedded TP is simultaneously a sister to the head of CP† and the C with the Q morpheme in it. FORM STRING would require this TP, then, to be pronounced so that it is adjacent to both of these complementizers, and this is impossible.[8] But the Q morpheme is an ellipsis licenser, and so it is amnestied from being subject to FORM STRING. This allows the embedded TP to be pronounced adjacent to the root C^0, heading CP†, and non-

[8] In this particular example, the root complementizer of CP† has a silent morpheme in it. I will assume that silent morphemes are subject to FORM STRING

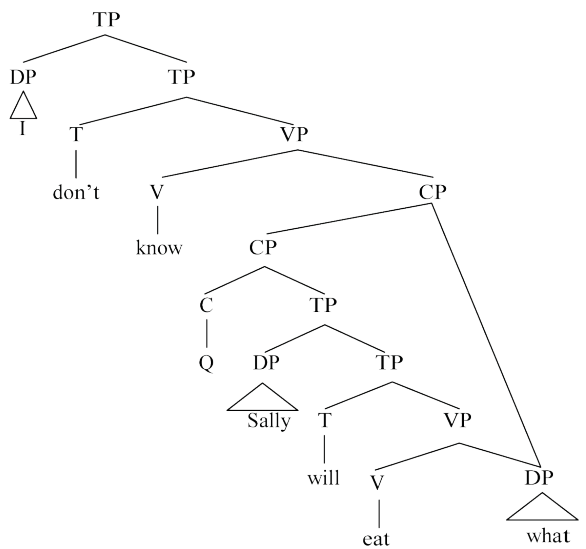
adjacent to the embedded question complementizer. FORM STRING operates everywhere else, however, and this keeps all the rest of *C Sally ate* and *I don't know what C* together. The result is the correct *Sally ate I don't know what*.

Consider next how this proposal applies to a more conventional sluice, like that in (55).

(55) Sally will eat something, but I don't know what.

The TP that follows *but* in (55) has the parse in (56).

(56)



As before, Movement causes *what* to be linearized as if it only occupied the Specifier of CP position: it must therefore precede and be adjacent to Q. FORM STRING, in turn, requires that *know* be adjacent to *what*, and that *don't* be adjacent to some terminal in *know what Q*, and that *I* be adjacent to some terminal in *don't know what Q*. The orderings sanctioned by $\mathscr{L}$ allow FORM STRING to achieve this requirements with just the ordering *I don't know what Q*. If FORM STRING applied to Q, then it would require that Q be adjacent to one of the terminals in the embedded TP, and $\mathscr{L}$ would make this terminal *Sally*. But Q is an ellipsis licensor, so FORM STRING does not have to apply to it, and this means that Sally need not show up in the resulting linearization. Because *Sally* need not show up in the linearization, neither must any word in the rest of the embedded TP. Indeed, if *Sally* is not pronounced in the linearization, then no other word in the embedded TP can be. Suppose, for instance, that *Sally* is not pronounced, but *will* is. Because the TP that immediately dominates *will* has its sister the DP that contains *Sally*, FORM STRING will require that *Sally* be pronounced if *will* is. Or suppose that *Sally* is not pronounced, but *ate* is. FORM STRING will require that *will* be pronounced if *eat* is, since the T dominating *will* is the sister to the VP

headed by *eat*, and this will consequently force *Sally* to be pronounced.

Formulating FORM STRING in this way, then, permits a way of unifying the condition that prevents ellipsis with the condition that enforces the mapping of phrases onto contiguous strings. And this, in turn, allows us to locate what an ellipsis licensor relaxes. It amnesties the condition that requires the linearization algorithm to spell out its sister adjacent to it. When there is an antecedent available, this allows for ellipsis. When there is a multidominant phrase marker of the sort proposed here, it allows for an Andrews amalgam. In the framework proposed here, this is expressed as letting ellipsis licensors be amnestied from FORM STRING. We are still left with the interesting, and unsolved puzzle, of why FORM STRING can ignore some lexical items and not others.

There are, no doubt, many ways in which this proposal could be improved. But if Guimarães is correct that the hosting clause in Andrews amalgams is actually embedded in the interrupting clause, and its ability to be linearized so that the interrupting clause is a string properly contained by it, traces back to it being able to be Sluiced, then the conclusion sketched at the outset holds. An ellipsis licensor does not automatically evoke the antecedence conditions on ellipsis. That is because an ellipsis licensor can license something that isn't an ellipsis: an Andrews amalgam.

Acknowledgments

I have been helped by the many comments and questions of the participants of the Workshop on Ellipsis at The Chinese University of Hong Kong in March 2013, where this paper was presented. A special thanks to Professor Sze-Wing Tang for his help, remarks and support. Marlies Kluck deserves thanks for teaching me everything about this construction. It is her hard work that I am building upon. The two anonymous reviewers pushed me to substantial revisions; they are responsible for substantial improvements.

References

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

Beck, Sigrid. 2006. Intervention effects follow from focus interpretation. *Natural Language Semantics* 14: 1–56.

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.

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

Cheng, Lisa Lai-Shen. 1991. On the typology of wh-questions. Doctoral dissertation, Massachusetts Institute of Technology.

Chomsky, Noam. 1981. *Lectures on Government and Binding.* Dordrecht, The

Netherlands: Foris Publications.
Chomsky, Noam. 2000. Minimalist inquiries: The framework. In *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, ed. Roger Martin, David Michaels, and Juan Uriagereka, 89–156. Cambridge, MA: MIT Press.
Fox, Danny. 2000. *Economy and Semantic Interpretation*. Cambridge, MA: MIT Press.
Guimarães, Maximiliano. 2004. Derivation and representation of syntactic amalgams. Doctoral Dissertation, University of Maryland.
Hagstrom, Paul. 1998. Decomposing questions. Doctoral dissertation, Massachusetts Institute of Technology.
Hagstrom, Paul. 2000. The movement of question particles. In *Proceedings of the North East Linguistic Society*, ed. Masako Hirotani, Andries Coetzee, Nancy Hall, and Ji-yung Kim, 275–286. New Brunswick, NJ: Graduate Linguistic Student Association, Rutgers University.
Kayne, Richard S. 1994. *The Antisymmetry of Syntax*. Cambridge, MA: MIT Press.
Kishimoto, Hideki. 2005. Wh-in-situ and movement in Sinhala questions. *Natural Language and Linguistic Theory* 23: 1–51.
Kluck, Marlies. 2011. *Sentence Amalgamation*. Groningen: Landelijke Onderzoeckschool Taalwetenschap.
Kratzer, Angelika. 2005. Indefinites and the operators they depend on: From Japanese to Salish. In *Reference and Quantification: The Partee Effect*, ed. Gregory N. Carlson, and Francis Jeffrey Pelletier, 113–142. Stanford: CSLI Publications.
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.
Merchant, Jason. 2001. *The Syntax of Silence: Sluicing, Islands, and the Theory of Ellipsis*. Oxford: Oxford University Press.
Reinhart, Tanya. 1997. Quantifier scope: How labor is divided between QR and choice functions. *Linguistics and Philosophy* 20: 335–397.
Reinhart, Tanya. 1998. *Wh*-in-situ in the framework of the Minimalist Program. *Natural Language Semantics* 6: 29–56.
Riemsdijk, Henk van. 1998. Trees and scions – science and trees. In *Festweb Page for Noam Chomsky*. Cambridge, MA: 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, 423-431. 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.
Shimoyama, Junko. 2006. Indeterminate phrase quantification in Japanese. *Natural Language Semantics* 14: 139–173.
Tancredi, Christopher. 1992. Deletion, deaccenting and presupposition. Doctoral

dissertation, Massachusetts Institute of Technology.
Wilder, Chris. 1998. Transparent free relatives. *ZAS Working Papers in Linguistics* 10: 191–199.

Mailing address: Department of Linguistics, University of Massachusetts at Amherst, Amherst, MA 01002 USA
Email: kbj@linguist.umass.edu
Received: May 20, 2013
Accepted: July 3, 2013

對省略的允許

Kyle Johnson
馬薩諸塞大學安姆斯特分校

提要

本文在 Marlies Kluck 和 Maximiliano Guimarães 著述的基礎上，對“Andrews 合成句”作出分析，指出兩個獨立句子共有同一子句，由此結合成“Andrews 合成句”，所謂“共有”子句，是指短語標記中的一個短語有兩個上一級成份，即一個短語獲予兩個母節點。“Andrews 合成句”通過“截省”得以允許：只有在共有的子句能被截省的情況下，這種合成句才能出現。從以上論述可見，省略時雖常見以先行語作為條件，但先行語條件並不總是允許省略。

關鍵詞

多重支配、合成句、短語結構、問句、截省、非連續依靠、線性化、省略