ANTECEDENT-GAP RELATIONS AND LOCALITY IN VERBAL ELLIPSIS

Elixabete Murguía

University of Deusto

This paper investigates the different locality restrictions that apply to some verbal ellipsis constructions in English; namely, Verb Phrase Ellipsis (VPE), Pseudogapping and Gapping. It is proposed that locality restrictions can be given a natural answer from the processing domain. Locality is analyzed as the result of the interaction of different factors: (i) tense presence/absence (Fodor 1985), (ii) low initial attachment of coordinates, and (iii) Spell-Out operations which render syntactic structure unavailable (Uriagereka 1999).

1. Introduction

It has been observed in the literature (i.e. Chao 1987 and Neijt 1980 on the competence side; and Berwick and Weinberg 1985, together with Fodor 1985 on the processing side) that not all elliptical constructions are subject to the same locality restrictions. The relation between the antecedent and the gap in Verb Phrase Ellipsis (henceforth VPE) and Pseudogapping can be either local (see examples 1a and 2a) or non-local (see examples 1b and 2b), while in the case of gapping, locality has to be respected (see example 3a versus 3b). If locality is not respected, then the sentence turns out to be ungrammatical:

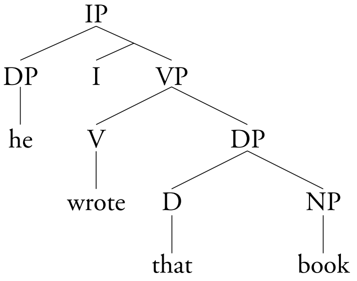
(1) a. Mary accepted the job offer, and Peter did too.
    b. Mary accepted the job offer, and I believe Peter did too.
(2) a. Tom talked to his wife, and Beth to her husband.
    b. Tom talked to his wife, and I heard Beth did to her husband.
(3) a. Susan prepared lunch, and John dinner.
    b. *Susan prepared lunch, and I think John dinner.

The analysis of locality effects advanced in this work is based on the minimalist framework (Chomsky 1993, 1995); in particular, on the economy principle that governs minimalism. Locality in coordinate elliptical structures is determined by Spell-Out operations, in the sense of Uriagereka (1999) —we will see that an antecedent remains in the local context of the gap if the former has not been spelled out. We assume Weinberg's (1999) human sentence processing algorithm (defined below in 6) and extend it to coordination and ellipsis. Before getting into the analysis of the ellipsis facts, a brief comment on some of the theoretical assumptions taken should be included first. This is done in the next two subsections.

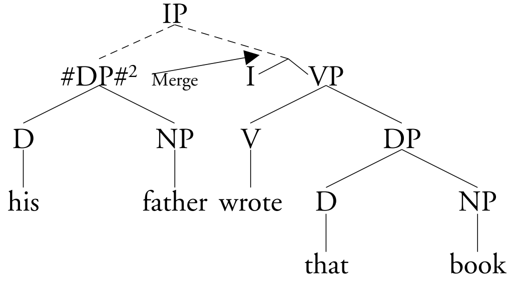
1.1. Multiple Spell-Out Theory (MSO) (Uriagereka 1999)

Uriagereka's (1999) Multiple Spell-Out (MSO) theory is an attempt to reduce Kayne's Linear Correspondance Axiom (LCA) to a more minimalist basis.[1] According to Uriagereka, once the D- and S-structure level have been abandoned, there is no reason to restrict Spell-Out to one unique application. He presents a dynamically split model, in which multiple application of Spell-Out applies, accessing PF and LF in separate derivational cascades. Following Epstein (1999), he proposes that command is a reflex of merge, and also that this command relation codes precedence relations: command maps to precedence in simple Command Units (CUs), because it is the simplest state of affairs. Consider example (4) and (5):

(4) He wrote that book.



(5) His father wrote that book.



Sentence (4) has been assembled through the monotonic application of the operation Merge —the word *that* is merged with *book*, the resulting object is merged with *wrote* and the object of this last operation is in turn merged with *he*— thus, constituting a so-called Command Unit. In the case of (5), however, we have a sen-

[1] Linear Correspondance Axiom (LCA):
Base step: If @ commands &, then @ precedes &.
Induction step: If $ precedes & and $ dominates @, then @ precedes &.

[2] The symbols # # mean that a category has been spelled-out.

tence which has been assembled through the non-monotonic application of merge to two separately assembled objects: (i) object A: *his* is merged with *father* constituting a Command Unit, and (ii) object B: *that* is merged with *book*, and then with *wrote*, constituting another Command Unit. Then, object A (the DP) —once it is spelled out— is merged with object B creating a single CU —which Uriagereka calls the mother CU.

Spell-Out applies to every CU in a derivation, linearizing the elements that compose them—or in other words, establishing the command-precedence relation among terminals. So, in example (4) the CU which is assembled monotonically through subsequent merge operations is spelled out and precedence relations established on the basis of command relations. Thus, the base step of the LCA is accounted for.

In the case of (5), however, the situation is somehow more complicated. The CU constituted by the DP *his father* is spelled out and the precedence relation between these two elements is established. After Spell-Out, what remains is not a phrase marker any longer. The resulting element (the DP) is frozen, it is a lexical compound, so the syntax cannot operate with it any longer —its syntactic structure cannot be altered. However, it can associate further up: the DP can be merged as a unit with the mother CU *wrote that book*. The command-precedence relation between the DP and the elements in this other CU is established in the following way: (i) the node DP commands the elements that constitute the mother CU, since the DP has been merged to those, and (ii) the elements that the label DP dominates should act as the label DP does within its mother CU —this is a consequence of the fact that they have been spelled out separately from that CU the DP has been attached to (they have been spelled out in a different derivational cascade); their place in the structure is frozen, so they cannot interact with the rest of the elements in the mother CU. Thus, the induction step of the LCA is also deduced.

1.2. A Minimalist Theory of Human Sentence Processing (Weinberg 1999)

Weinberg (1999) assumes the minimalist program (Chomsky 1993, 1995), and applies minimalist operations—Merge, Move and Spell-Out (as defined by Uriagereka 1999 above)—together with minimalist principles —economy principles— to parsing. She defines a minimalist algorithm for human sentence processing which not only accounts for some attachment preferences observed in the literature, but also offers a theory of reanalysis. Her algorithm definition is included below:

(6) A derivation proceeds left to right. At each point in the derivation, merge using the fewest operations needed to check a feature on the category about to be attached. If merger is not possible, try to insert a trace bound to some element within the current command path. If neither merger nor movement is licensed, spell out[3] the command path. Repeat until all terminals are incorporated into the derivation.

[3] Spell-Out operations are carried out in a phrase-by-phrase manner. In other words, when a command unit is spelled-out, the whole command unit is not linearized at once, but rather each of the phrasal nodes are spelled-out one by one.

Weinberg (1999) assumes a MSO theory for performance in order to account for the mapping between precedence and dominance relations without the need for the LCA in parsing as well.[4] The base step of the LCA (in footnote 4) is deduced from the fact that it is the simplest mapping relation between precedence and dominance (a one-to-one mapping relation). The induction step is not necessary if MSO applies. Spell-Out applies whenever two categories cannot be merged together (see algorithm definition above): if neither Merge nor Move can apply, then the category being built is spelled out; linearized —or in other words, turned into an unstructured string. For this spelled out string the only important precedence relations are those already established. Precedence does not need to be established between the elements in this string and the rest of the items in the structure.

The algorithm defined above accounts for certain parsing preferences, e.g. Argument-over-Adjunct attachment (Pritchett 1992 and Gibson 1991) and Minimal Attachment Principle (Frazier and Rayner 1982) —see Weinberg (1999) for discussion. It also offers a theory of reanalysis: we are going to discuss this last point in some detail, since it is crucial for the analysis of verbal ellipsis and locality that follows. In order to do so, we look at two examples: in the first one reanalysis is possible, but in the second one reanalysis is blocked by the prior application of Spell-Out.

Reanalysis to a different reading remains possible within a domain where Spell-Out has not applied. Consider sentence (7) below. A verb like *believe* subcategorizes both for a DP and an IP (see examples in 8):

(7) The man believed his sister to be a genius.
(8) a. He believed [$_{DP}$ his sister].
b. He believed [$_{IP}$ [$_{DP}$ his sister] to be clever].

At the point where the determiner *his* is encountered in sentence (7), the parser has two possibilities for attachment: (i) attach the DP as the object of the verb *believe* (as in 8a), or (ii) attach it as the subject of the embedded IP (as in 8b). The parser goes for the first option, since not only is it the most economical one (fewest nodes), but also it allows feature checking (case and theta-role) for the DP. Attachment as the subject of the embedded clause does not allow any feature checking at this point, since the head of the IP has not been processed yet.

When the embedded verb *to be* is processed, the parser needs to reanalyze the syntactic structure it assigned to sentence (7) so as to accommodate the new input items (it needs to reanalyze the attachment of the DP from object into subject of the embedded clause). Thus, however, in does not present any problem, because

[4] The LCA as proposed by Kayne (1994) derives linear precedence from dominance relations. Weinberg (1999) inverts the claim so as to make it relevant for parsing purposes:

(i) Linear Correspondance Axiom (LCA)
Base step: If α precedes β then α dominates β
Induction step: If γ precedes β, and γ dominates α, then α precedes β.

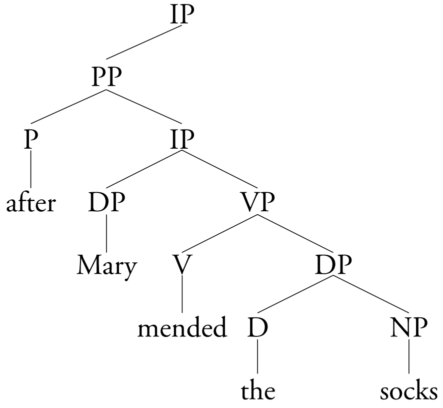
both the verb and the DP are available: Spell-Out has not applied to these categories yet.

On the contrary, if Spell-Out applies, then extraction or insertion of syntactic material is not possible: once a syntactic structure is spelled out it is frozen, and it cannot be affected by operations such as merge or move. Consider sentence (9) below:

(9) *After Mary mended the socks fell off the table.

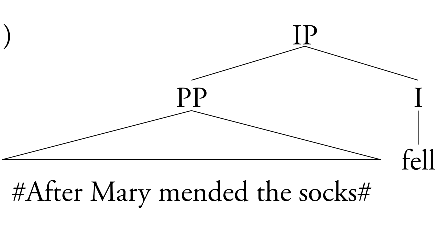
When we start processing this sentence a PP is built headed by the preposition *after*. Once we arrive to the DP *the socks*, there are two possible attachments: (i) attachment of the DP as the object of the verb *mend*, or (ii) attachment as the subject of the matrix clause. The first option is the one that the parser chooses (as in 10 below), since it allows features of the DP to be checked by the verb *mend*. Attachment as the subject of the matrix clause does not allow the checking of any feature at this point, since the head of the matrix clause *fell* is not part of the structure yet.

(10)



At this point of the parse, the next item to be attached is the verb *fell*. However, it cannot be merged with anything in the preceding clause. Thus, Spell-Out applies: the adverbial clause is linearized (c-command is established for the elements of this string) and the verb *fell* is attached to the structure. The resulting structure is that shown below:

(11)



At this point, reanalysis to the non-preferred reading (where the DP is the subject of the matrix clause) is not possible, because the domain were the DP is attached has been spelled out, and consequently, it is not possible to retrieve it in order to attach it as the matrix subject.

2. Verbal Ellipsis and Locality Restrictions

In this section, a minimalist processing account of verbal ellipsis and locality restrictions is introduced. The starting point is Weinberg's (1999) human sentence processing algorithm (defined in 6 above), which is extended to coordination and ellipsis here.

An analysis for parsing different verbal ellipsis constructions is advanced, based on an algorithm defined on the minimalist operations (Merge, Move and Spell-Out), which takes into account economy considerations and which makes use of local information.

Locality is explained as a result of the interaction of different factors: (i) Tense presence/absence, (ii) low initial attachment of coordinates, and (iii) Spell-Out operations which render syntactic structure unavailable. These last two together determine when left-context, i.e. the antecedent in these ellipsis contexts, is available.

The problem for parsing is (i) to detect the gap, and (ii) to resolve/interpret it. As a preview of what is coming, it should be mentioned that there is a contrast between VPE and Pseudogapping constructions on the one hand, and Gapping constructions on the other. For the first two, the gap is detected by the presence of an auxiliary —the auxiliary signals the gap and allows us to predict a VP (the antecedent only needs to be accessed to interpret the predicted VP)— while in the case of gapping the antecedent needs to be consulted to assign structure to the gap and for interpretation purposes.[5]

2.1. Tense Presence/Absence and Locality

As it has already been noticed, there exists a crucial difference between VPE and Pseudogapping elliptical constructions on the one hand, and gapping on the other. In the case of VPE and Pseudogapping, there is an auxiliary overtly realized in the elision site (see 12 and 13). On the contrary, in gapping sentences there is no auxiliary present (see 14):

(12) Mary is very hungry, and I am too.
(13) Peter gave his corrections to Susan, and John did to Bill.
(14) These students ate bagels, and the visitors pizza.

This auxiliary difference is crucial for detecting and resolving the gap. In the case of VPE and Pseudogapping, since there is an auxiliary, an IP can be built and a VP

[5] In this paper, I just deal with the work that is done on-line by the parser. For a further discussion of how the interpretation process is carried out in ellipsis contexts I refer the reader to Murguía (2004), where both on-line and off-line processes are discussed.

predicted (functional categories like "I" select lexical categories like "V"): the auxiliary is recognized on the basis of the input string (bottom-up), an IP is built, and a top-down prediction of a VP can be made. All this is done by using local context, i.e. the information provided by the auxiliary. There is no need to access the antecedent to detect the gap and assign structure to it. However, in the case of gapping, there is no overt auxiliary or verb from which to build an IP, and the antecedent needs to be accessed in order to detect the possibility of a gap. The antecedent is needed to postulate a node for the gap. This Tense effect was already noticed by Fodor (1985), and discussed by Berwick and Weinberg (1985).

In VPE and Pseudogapping sentences, the VP that is predicted is assigned a pointer to the antecedent VP and it shares the structure with the latter.[6] The antecedent structure is accessed only for interpretation. It is not accessed on-line to build the structure of the gap. For gapping, the antecedent is accessed on-line to assign structure to the gap.

The difference proposed here is supported by some findings which have been reported in the psycholinguistics literature. Frazier and Clifton (2001) report what they call "missing complexity effects" in VPE sentences. In a self-paced reading experiment, they did not find any difference in the reading times of those sentences below, even though the structure of the antecedent in (15) is more complex than in (16):

(15) Sarah left her boyfriend last May. Tina did too.
(16) Sarah got the courage to leave her boyfriend last May. Tina did too.

This contrasts with complexity effects found by Carlson (2002) for gapping sentences. This difference between VPE and gapping sentences supports the distinction that we have proposed above for VPE and gapping. In the case of VPE, it seems that the antecedent is not accessed on-line for gap detection; otherwise, if the structure of the antecedent is computed for the gap, then there should be differences in the reading times of the sentences in (15) and (16).

2.2. Low Initial Attachment of Coordinates

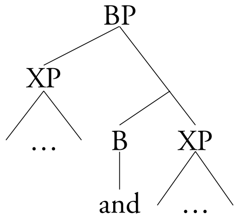
Weinberg (1999) evaluates ambiguity of attachment with respect to economy: the most economical structure is preferred, i.e. that one that involves fewest nodes or operations. We translate this economy preference into initial low attachment for coordinates.

As we will see in this section, there is ambiguity of attachment in the case of coordinates too. Coordinators are initially attached low, and this decision is revised into high attachment if later incoming material forces reanalysis —we will see how reanalysis is carried out in detail when I discuss some examples in the next section.

[6] For a detail discussion of this pointer assignment and of the sharing relation that exists between the antecedent and the gap see Murguía (2004).

We assume that coordinators head Boolean Phrases (BPs) as proposed by Munn (1987) —coordinate sentences have the structure in (17) below. As proposed in Murguía (2000), coordinates are spelled out in different CUs in order to preserve the precedence-command relationship among terminal elements:

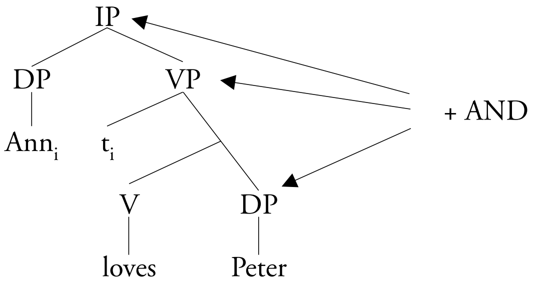
(17)



Now, let us consider why low attachment is more economical than high attachment. For a sentence like (18), at the point where the coordinator is encountered the structure computed so far is that one in (19), where the structure of what is going to be the first conjunct is already built. At this point, the next input item to be attached is the coordinator *and*. How is the coordinator attached? There are three possible attachment sites, marked with arrows—the three possible attachment sites are the (i) IP, (ii) VP, and (iii) DP nodes:

(18) Ann loves Peter, and Mary does too.

(19)

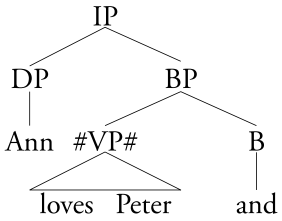


How does the parser choose among these possibilities? Recall that a minimalist grammar is assumed here, and the most important principle in minimalism is the principle of economy —derivations must be as economical as possible (fewest number of steps/operations and fewest number of nodes). This economy principle is what is going to guide the parser in choosing among the three alternatives. Let us consider these in turn.[7]

[7] Using the strategy of actually considering all the possible alternatives will be a problem for efficiency in parsing—if all the different attachment sites were considered on-line, then the algorithm will be n^2 proportional to attachment sites, which does not respect efficiency. Therefore, economy is enforced in a serial way. Low attachment is the initial choice the parser takes, because it is more economical. The goal of the discussion that follows is to illustrate why it is the case that low attachment of coordinates is more economical. This then justifies the assumption that it becomes the automatic first option without the need for global comparison.

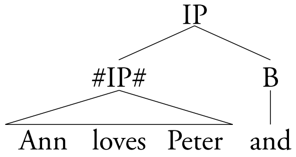
Start by considering alternative (ii): attachment to the VP. In order to attach the coordinator to the VP node and preserve the command-precedence relationship, the intervening material must be spelled out so that this position becomes available. Since Spell-Out occurs in a phrase-by-phrase manner, the DP will be spelled out first, followed by the VP. After these two Spell-Out operations, the coordinator can be merged to the structure. This is what the structure will look like after all these operations:

(20)



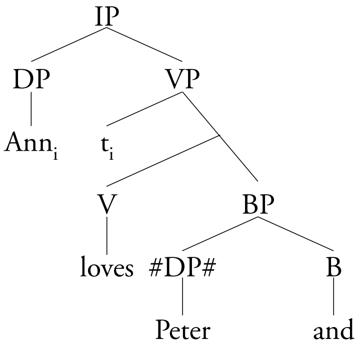
Alternative (i), attachment to the IP node, will include the same steps as attachment to the VP plus one more Spell-Out operation —Spell-Out of the IP phrase. The structure after attaching the coordinator will be that in (21):

(21)



Turning now to the third possibility: attachment to the DP. If the coordinator is attached to the DP then only one Spell-Out operation is necessary: spelling out of the DP as in (22) below. Thus, this third possibility is the most economical one (it involves fewest steps/operations), and the one that the parser chooses—decisions are taken locally, this algorithm is not a global one, and at this point in the derivation attachment to the DP is the best option in terms of economy:

(22)



Thus, according to economy, it looks like low attachment should be preferred for coordinates too. This tendency for attaching low has already been observed in the parsing literature in other contexts different from coordination: Minimal Attachment (Frazier and Rayner 1982), which has been explained in terms of a minimalist parsing algorithm that favors feature checking, and respects economy (Weinberg 1999).

2.3. Locality Effects and Verbal Ellipsis

As we have mentioned, it is a well-known fact in the literature that elliptical constructions are subject to different locality restrictions (e.g. Chao 1987, Fodor 1985, and Berwick & Weinberg 1985). VPE and Pseudogapping are not constrained by any locality restrictions, while gapping is. See examples below, which exemplify this contrast:

(23) a. Ann loves Peter, and Mary does too.
b. Ann loves Peter, and Susan thinks Mary does too.
(24) a. I gave money to Susan, and Peter did to Beth.
b. I gave money to Susan, and you heard that Peter did to Beth.
(25) a. John saw Carmen, and Tom Othello.
b. *John saw Carmen, and Bill thinks Tom Othello.

Both for sentences (23) and (24) (examples of VPE and pseudogapping, respectively) the antecedent verb phrase, and the elided constituent can be separated by intervening material —the gap can be embedded (as in the "b" examples), and still result in a grammatical sentence. However, in the case of gapping (sentence 25), if the antecedent and gap are not local (if the elided constituent is embedded as in 25) then the sentence is ungrammatical.

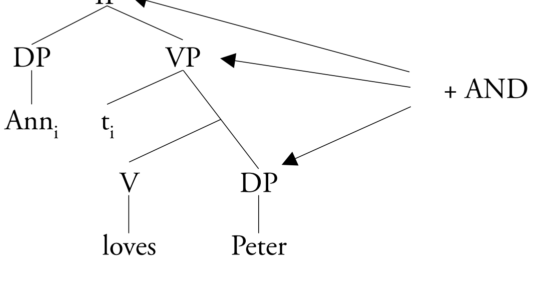
In this section, we propose an analysis for the presence/absence of locality effects in ellipsis which is based on (i) the presence/absence of the auxiliary, and (ii) the availability of left context (i.e. of the antecedent), which in turn is a result of low initial attachment of coordinates and of Spell-Out operations that render syntactic structure unavailable.

2.3.1. VPE

Let us start with the VPE example (26). Through subsequent Merge and Move operations, the first conjunct structure is built. The next input item to be attached is the coordinator *and* and the three possible attachment sites are those in (27):

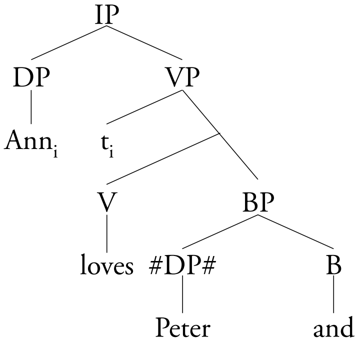
(26) Ann loves Peter, and Mary does too.

(27)



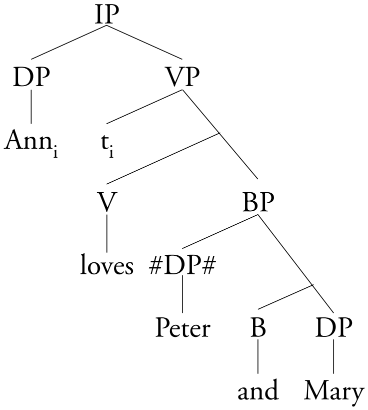
As we saw in the previous section, the parser chooses low attachment (i.e. attachment to the DP), since this is the most economical option; the one that involves fewest steps.

(28)



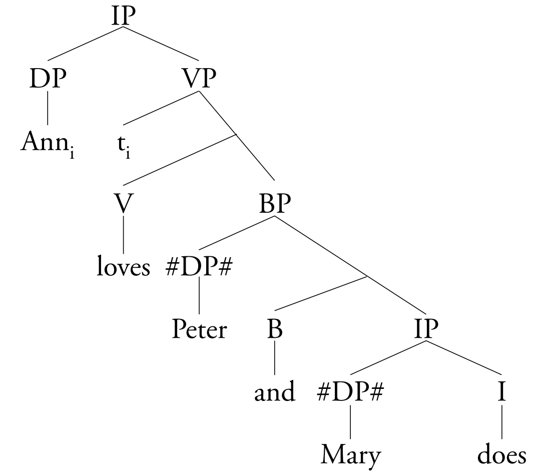
Once the coordinator has been merged to the structure, the next input item to be attached is the DP *Mary*, which will be attached as follows:

(29)



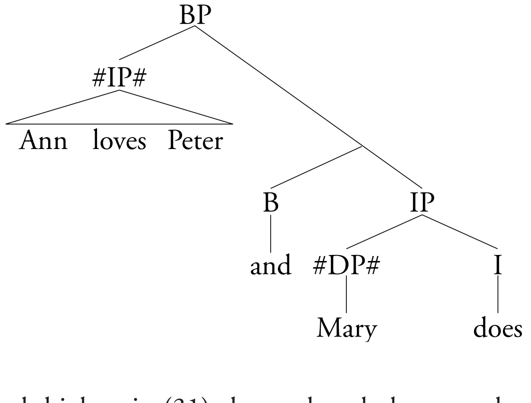
There is a condition on coordination that must be respected: the coordinator must conjoin two identical categories. (Coordination of Likes: Williams 1981). If two different categories are coordinated, then this constraint is violated and the sentence is ungrammatical. In (29) above, two DPs are coordinated, so the condition on coordination of likes is respected. The next input item to be attached is the auxiliary *does*. As before, the most economical option is to attach low for the same reason: it involves fewest steps.

(30)



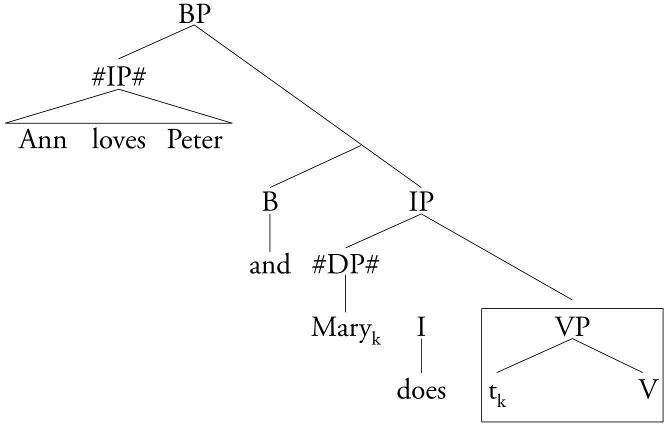
However, one more node (IP) had to be postulated (the DP *Mary* that was coordinated with the DP *Peter* has been transformed into an IP). Now *and* is coordinating a DP *Peter* and an IP *Mary does*, which violates the condition on coordination. At this point reanalysis is necessary. Low attachment is reanalyzed as high attachment, in other words, coordination of objects is reanalyzed as coordination of IPs. To respect the condition on coordination there is only one possibility now, and that is attachment to the IP, as in (31) below:

(31)



To attach high as in (31) above, the whole antecedent IP needs to be spelled out so as to preserve the precedence-command relationship among terminals. Spelling out the antecedent makes its internal syntactic structure unavailable. However, at this point where an IP has been built bottom-up for the second conjunct, we can predict a VP; since functional categories select lexical categories (a top-down prediction can be made). We do not need to look back to the antecedent to do this, so whether its syntactic structure is available or not is irrelevant for the parser to successfully detect the gap and assign a category to it. We can also relate the subject in the specifier of IP to its base position —the specifier of VP— and build all the structure in (32) below:

(32)



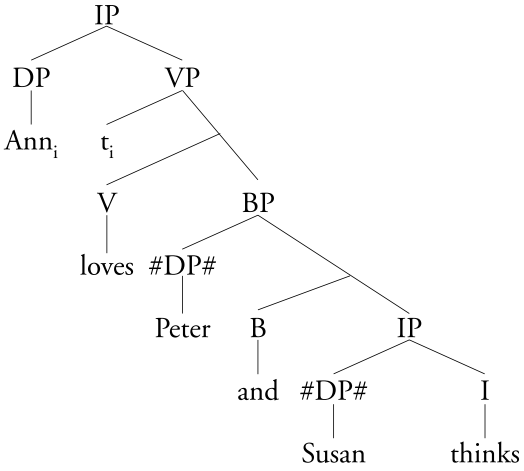
The complete internal structure of this VP, however, as well as the lexical content of the V-category and its complement are not fully specified. The rest of the VP structure and the lexical content are recovered from the antecedent, by following the pointer that the elided VP is assigned. So the antecedent is retrieved for interpretation purposes at LF, but not on-line when the gap is encountered.

Consider now the second example of VPE mentioned above —where the antecedent and the elided VP are separated by intervening material— repeated here for the reader's convenience:

(33) Ann loves Peter, and Susan thinks Mary does too.

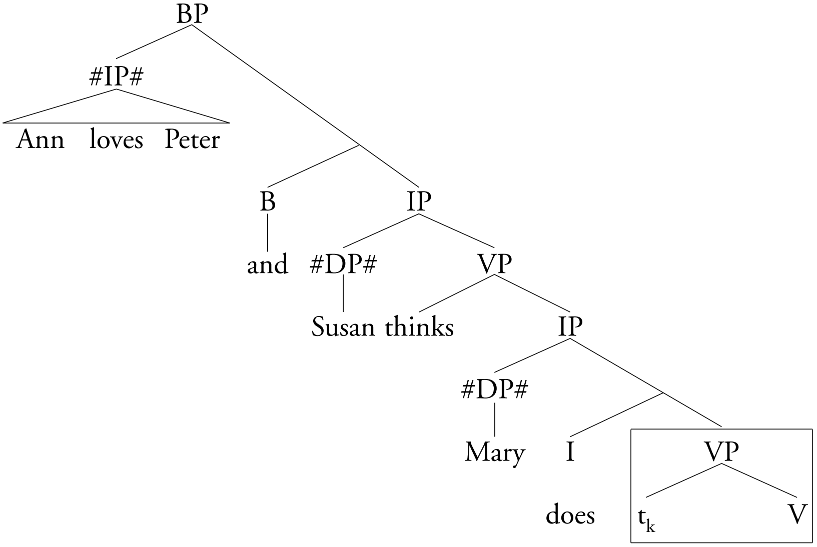
This sentence will be parsed in the same way as the previous example —the coordinator will be attached low, since this is the most economical option. Reanalysis from low into high attachment here, however, will be triggered by the attachment of the intervening clause *Susan thinks* in (34) below:

(34)



At this point, the condition on coordination is not satisfied (a DP and an IP are coordinated) and reanalysis is necessary. The coordinator is attached to the higher IP (as in (35)). In order to attach high, the antecedent clause needs to be spelled out (recall that after spelling out a category, its internal syntactic structure is no longer available); consequently, the antecedent VP will not be accessible —the antecedent cannot be retrieved to assign structure to the gap. Nevertheless, this is not a problem for VPE examples since the elided VP can always be predicted from the IP (built top-down based on the auxiliary *does*), without resorting to the antecedent's help:

(35)



2.3.2. Pseudogapping

Consider now the pseudoggapping (or subdeletion) examples mentioned above, which are repeated below:

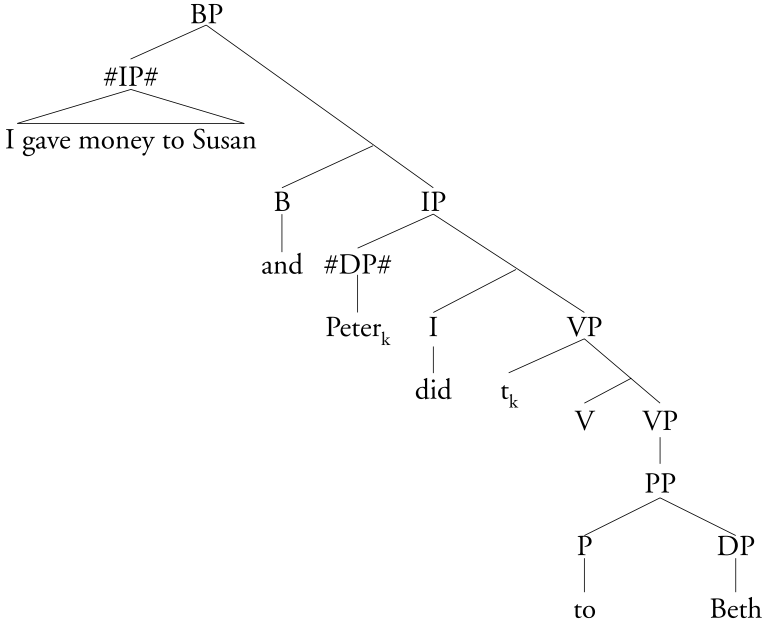
(36) a. I gave money to Susan, and Peter did to Beth.
b. I gave money to Susan, and you heard that Peter did to Beth.

Pseudogapping sentences, like VPE, are grammatical whether the antecedent and the elided clause are local or not: in (36) for example there is intervening material between both clauses, but the sentence is still grammatical. An auxiliary is always present, as in the case of VPE, so an IP is built and a VP can be predicted without the need to access the antecedent.

One difference between VPE and pseudogapping examples is that one of the verb arguments/adjuncts in the elided conjunct is overtly realized only in the latter. In example (36) above, the indirect object *to Beth* has not been elided. How is this overtly realized argument attached, when the verb phrase is elided? Since a VP is predicted top-down (as in the VPE cases) for all pseudogapping cases the argument is attached as part of that predicted VP.

Consider the parse for example (36). The argument will run in the same way as for the VPE examples above—attachment of the coordinator starts low and this is reanalyzed to IP attachment when the auxiliary *did* (in 36) and the clause *you heard* (in) are attached. Once the second conjunct is reanalyzed as an IP, a VP is predicted (a top-down prediction). Finally, the non-elided argument is merged to the structure:

(37)



The sentence in (36) is parsed in the same way, with the difference that reanalysis in this case is triggered by the intervening material (as in 33 above). But since an auxiliary is present in the elided conjunct an IP is built, and a VP predicted —to which the overtly realized argument is attached. Because of this possibility to predict a VP, the non-availability of the antecedent (it has been spelled out, so it is not available) does not pose a problem neither for the resolution of the gap nor for the attachment of the argument *to Beth* —both can be done without the need to resort to the antecedent.

2.3.3. Gapping

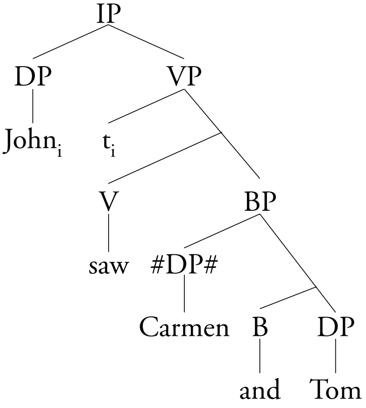
The gapping examples differ from VPE and pseudogapping by showing locality effects. When the antecedent and the elided clause are separated by intervening material (as in 38 below), then the sentence is ungrammatical:

(38) a. John saw Carmen, and Tom Othello.
b. *John saw Carmen, and Bill thinks Tom Othello.

With the minimalist parsing algorithm that we have assumed, particularly with a theory of MSO where material is rendered inaccessible for further computation after being spelled out, we can account for why these locality effects are observed in gapping.

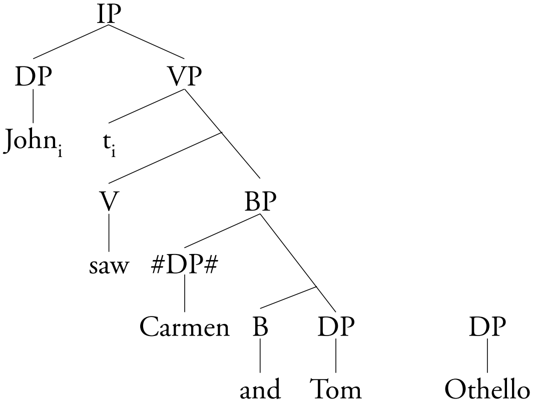
Consider sentence (38). The first conjunct is parsed and the coordinator *and* once more is attached low for economy reasons. The next item attached is the DP *Tom*, which is attached low, as a coordinated object. The structure at this point looks like:

(39)



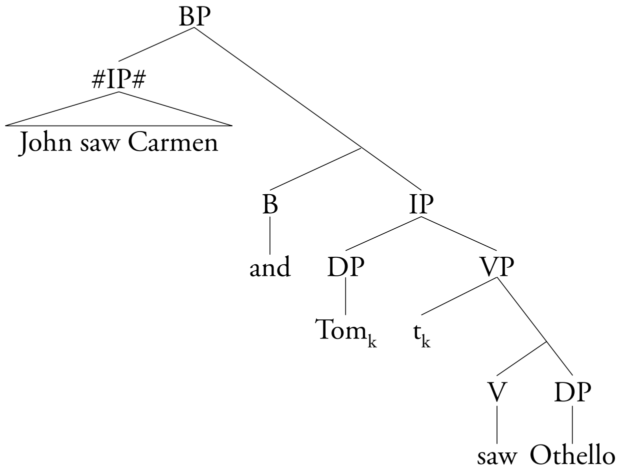
Now, the next word to be attached is the DP *Othello*, but there is no way in which it can be attached to the structure. In this case, we do not have an auxiliary or verb in the current clause, as in the VPE or Pseudogapping examples, that will help us predict a VP. The parser needs to go back to the antecedent clause and use the information about the predicate in that antecedent clause to relate the two arguments *Tom* and *Othello*:

(40)



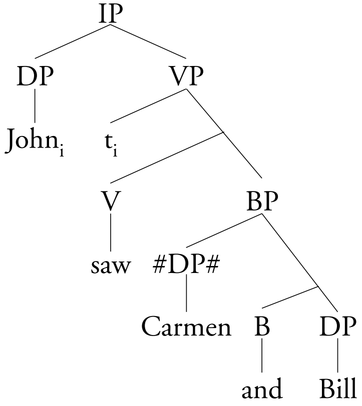
The parser reanalyzes the structure by looking for an antecedent in the c-command path. The antecedent is still available because of the initial mistake of attaching low triggered by economy. The coordination attachment is reanalyzed and the resulting structure is that in (41) below:

(41)



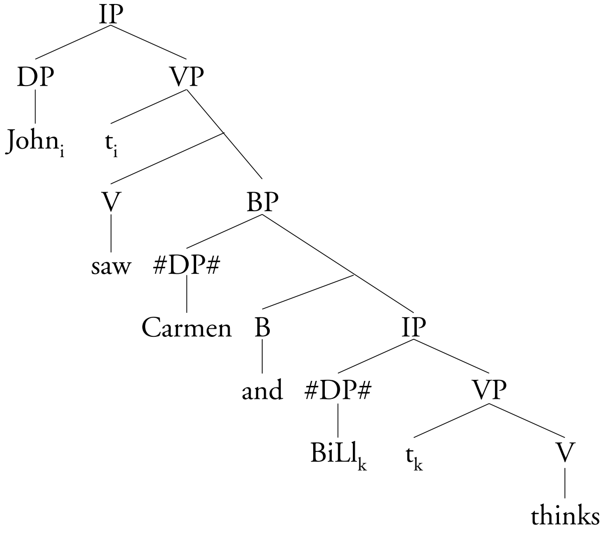
However, in the case of (38) the verb gap cannot be reconstructed, because by the time the parser gets to the gap the antecedent has already been spelled out. Let us see this in some more detail. The coordinator and the DP *Bill* are attached low (for the same reasons we have claimed for the previous examples), as in (42):

(42)



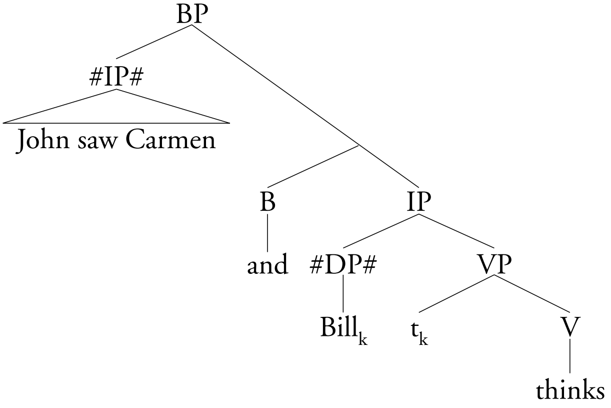
Reanalysis for sentence (38) is triggered when the verb *think* is merged to the already existing structure, since the resulting structure violates the condition on identity of categories for coordination:

(43)



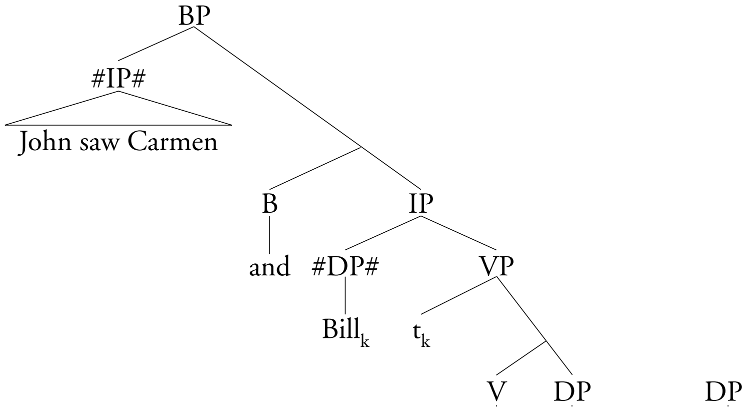
This structure is reanalyzed as follows. We have an IP so the only possibility is to attach as a coordinated IP. To do so, the first conjunct (i.e. the antecedent) is spelled out in order to preserve the command-precedence relations:

(44)



The parse will follow by attaching the DP *Tom* as in (45) below, but then the next word *Othello* cannot be attached to the existing structure in any way. An auxiliary is not present in the elided clause, as in the VPE or pseudogapping cases, so a VP cannot be predicted and the gap cannot be interpreted. Because the antecedent has already been spelled out, it cannot be accessed to license the gap, resulting in an unacceptable sentence:

(45)



Thus, we have seen how the locality effects that gapping cases display can be explained in terms of MSO, and of the minimalist algorithm that we are assuming. In gapping, the antecedent and the elided clause must be local; otherwise, the gap cannot be licensed. The gap depends on the antecedent to be reconstructed.

3. Conclusions

We have offered an account for parsing elliptical constructions which makes use of the minimalist operations: Merge, Move and Spell-Out; which takes into consideration economy issues, and which makes use of local information.

We have accounted for the presence/absence of locality restrictions in ellipsis as a result of the interaction of the following factors: overt tense presence/absence, and of the availability of left-context (i.e. the antecedent), which in turn is a consequence of (i) low initial attachment of coordinates, and (ii) Spell-Out operations which render syntactic structure unavailable. We have seen that in the case of gapping (an ellipsis construction where the relation between the antecedent and the gap must be local) the antecedent needs to be accessed to assign structure to the gap; therefore locality restrictions between the antecedent and the gap apply. However, in the case of VPE and Pseudogapping, a VP may be predicted top-down without resorting to the antecedent, which is only accessed for interpretation purposes. Therefore, these two are not subject to locality restrictions.

Thus, it has been showed that locality restrictions in ellipsis, which have not been properly accounted for from the competence side, do find a natural and satisfactory answer in the processing domain. Locality in ellipsis is reformulated here in terms of c-command: an antecedent is available for gap resolution if it remains in the same c-command path of the gap, or in other words, if they belong to the same CU.

Acknowledgements

I would like to acknowledge the contributions of different people to this paper. First, I have to thank Amy Weinberg who has directed this research. Thanks to Juan Uriagereka and Colin Philips who have also discussed the contents of this paper. I am also indebted to the audiences of the Euresco Conference on Theoretical and Computational Linguistics, the 16th Annual CUNY Conference on Sentence Processing, the CUNY Syntax-Psycholinguistics Supper Club Meeting, the CNL Lab Lunch Meeting, and the Euskal Mintegia. Thanks to Peter Lavery for proof reading and comments. This research was supported by a grant from the Basque Government Department of Education, Universities and Research.

Selected references

Berwick, B., and A. Weinberg, 1985, "Deterministic Parsing and Linguistic Explanation", *Language and Cognitive Processes* 2, 109-34.

Carlson, K., 2001, *Parallelism and Prosody in the Processing of Ellipsis Sentences.* Doctoral Dissertation. U. of Massachusetts, Amherst.

Chao, W., 1987, *On Ellipsis.* Doctoral Dissertation, U. of Massachusetts, Amherst.

Chomsky, N., 1993, "A Minimalist program for Linguistic Theory". In *The View from Building 20: Essays in Linguitics in Honor of Sylvain Bromberger*, eds. K. Hale and S. J. Keyser. Cambridge, Mass.: MIT Press.

—, 1995, "Categories and transformations". In *The Minimalist Program.* Cambridge, Mass.: MIT Press.

Epstein, S., 1999, "Un-Principled Syntax". In Epstein and Horstein (eds.).

— & N. Hornstein (eds.), 1999, *Working Minimalism,* Cambridge, Mass: MIT Press.

Fodor, J. D., 1985, "Deterministic Parsing and Subjacency", *Language and Cognitive Processes* 1, 3-42.

Frazier, L. and Rayner, K., 1982, "Making and Correcting Errors during Sentence Comprehension. Eye Movements in the Analysis of Structurally Ambiguous Sentences", *Cognitive Psychology* 14, 178-210.

Kayne, R., 1994, *The Antisymmetry of Syntax.* Cambridge, Mass.: MIT Press.

Munn, A., 1987, "Coordinate Structures and X-bar Theory". *McGill Working Papers in Linguistics* 4-1, 121-40.

Murguia, E., 2000, "VP Ellipsis as Null Lexicalization: Strict and Sloppy Readings Emerge at Different Stages in the Derivation". In *Maryland Working Papers in Linguistics (UMWPiL)* 9, 172-202. M. Guimaraes, L. Meroni, C. Rodriguez, and I. San Martin (eds.).

—, 2004, *Syntactic Identity and Locality Restrictions on Verbal Ellipsis.* Doctoral Dissertation. University of Maryland at College Park, USA.

Neijt, A., 1980, *Gapping: A Contribution to Sentence Grammar.* Doctoral Dissertation. University of Utrecht, Dordrecht, Holland.

Pritchett, B., 1992, *Grammatical Competence and Parsing Reference.* Chicago: U.P.

Uriagereka, J., 1999, "Multiple Spell-Out". In Epstein and Horstein (eds.).

Weinberg, A., 1999, "A Minimalist Theory of Human Sentence Processing." In Epstein and Horstein (eds).