ABSTRACT

Title of dissertation: DERIVATION AND REPRESENTATION OF SYNTACTIC AMALGAMS

Maximiliano Guimarães Miranda, Doctor of Philosophy, 2004

Dissertation directed by: Professor Juan Uriagereka
Department of Linguistics

This dissertation consists of an investigation of *syntactic amalgamation* (cf. Lakoff 1974): the phenomenon of combination of sentences that yields parenthetic-like constructions like (01).

(01) John invited God only knows how many people to you can imagine what kind of a party.

The theoretical framework adopted is the *Generative-Transformational Grammar* (Chomsky 1957, 1965, 1975, 1981, 1986b, 2000b), following (and elaborating on) the recent developments known as the *Minimalist Program* (Chomsky 1995, 2000a, 2001a, 2001b; Martin & Uriagereka 2000; Uriagereka 1998, 1999, 2002).

As far as the representation of syntactic amalgams is concerned, the main claim made in this dissertation is that such constructions involve a radical form of shared constituency, where two or more matrix sentences share the same subordinate sentence, in a multiply-rooted phrase marker.

As far as the derivation of syntactic amalgams is concerned, the main claims made in this dissertation are: (i) context-free shared constituency arises from overlapping numerations; and (ii) the computational system builds structure incrementally, in a *generalized tucking-in* fashion, which yields a left-to-right/top-to-bottom effect on the derivation, such that constituency is heavily dynamic (along the lines of Phillips 1996, 2003; Drury 1998a, 1998b, 1999; Richards 1999, 2003).

The conclusion is that this particular kind of paratactic-like construction is better understood as a purely syntactic phenomenon, where the resources of the computational system are pushed to the limit.

DERIVATION AND REPRESENTATION OF SYNTACTIC AMALGAMS

by

Maximiliano Guimarães Miranda

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland at College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2004

Advisory Committee:

Professor Juan Uriagereka, Chair
Professor Norbert Hornstein
Professor Michael Long
Professor Howard Lasnik
Professor Amy Weinberg

Freeze this moment a little bit longer.
Make each sensation a little bit stronger.
Make each impression a little bit stronger.
Freeze this motion a little bit longer.
Experience slips away.
The innocence slips away....
Time Stand Still!

(Neil Peart)

I dedicate this book to **Beth Rabbin**,
with Love.
For sharing the Rainbows,
in all those Magic Days.
For sharing the Music,
in all those Endless Nights.
For her Smile,
her Smell,
and her Shining Eyes.
For making me feel Happy,
like a little Boy.
For turning the worst time of my life
into my Wonder Years.

ACKNOWLEDGEMENTS

I am extremely thankful to my advisor Juan Uriagereka, for guidance and inspiration.

I am thankful to my Professors at the University of Maryland: Juan Uriagereka, Norbert Hornstein, Paul Pietroski, David Lightfoot, Howard Lasnik, Amy Weinberg, Colin Phillips, for all they taught me along the way.

I am thankful to my Professors from back home at Universidade Federal da Bahia and Universidade Estadual de Campinas: Ilza Ribeiro, Dante Lucchesi, Rosa Virgínia Matos e Silva, Charlotte Galves, Maria Bernadete Abaurre, Mary Kato, Rodolfo Ilari, and Eleonora Albano, for guiding me in my first steps.

I am extremely thankful to Cilene Rodrigues, for sharing the pain and the loneliness.

I am extremely to John Drury, for the inspiration.

I am super-duper thankful to Beth Rabbin, for her Love, for being here for me in the Magic Days.

I am thankful to the fellow graduate students Cilene Rodrigues, John Drury, Leticia Pablos, Soo-Min Hong, Hirohisa Kiguchi, Itziar San Martin, Elixabete Murguia, Ana Gouvêa, Acrisio Pires, Kleanthes Grohmann, Juan Carlos Castillo, Usama Soltan, Roberta D'Alessandro, and Andrew Nevins, for the friendship.

I am deeply thankful to Camilo Dorea and Gabriela Alvarez, for sharing, and for the friendship.

I am deeply thankful to Alexander Fortin for the friendship.

I am thankful to Marcelo Braga for inviting me to you can imagine what kind of parties.

I am deeply thankful to Francisco Simões, for the friendship and the emotional support.

I am deeply thankful to my mother Darilda, my father Murilo and my brother Carlos Frederico, for pretty much everything..

TABLE OF CONTENTS

I

Walking on the Fine Line Between Syntax and Parataxis

The content of this dissertation has both analytical and theoretical aspects to it,[1] which I introduce in **I.1** and **I.2** below, respectively.

I.1. The Structure of Syntactic Amalgams

The empirical focus of this dissertation is *syntactic amalgamation*: a very puzzling phenomenon first discovered, reported, described and analyzed by Lakoff (1974), on the basis of empirical observations made by Avery Andrews, Larry Horn, William Cantral, and Mark Liberman, as well as by George Lakoff himself.

A typical example of syntactic amalgam is given in (01).

(01) Homer drank I don't remember how many beers at the party.

Ever since Lakoff's pioneering and insightful work, the phenomenon of syntactic amalgamation has been almost completely ignored all these years. After

[1] Here I commit to the definitions of *analytical work* and *theoretical work* given by Chametzky (1996: xvii-xviii).

Lakoff's (1974) work, little, if anything, has been said about syntactic amalgams. The only exceptions to that major hiatus have been brief mentions to that seminal paper by Lakoff in the context of historical debates about the 'anti-D-structure school' of the nineteen-seventies (e.g. Huck & Goldsmith 1995: 117), in discussions about conversational implicature (e.g. Levinson 1983: 164-165). However, no new descriptive or analytical contribution has been given beyond Lakoff's findings, as far as I know. [2]

More recently, Tsubomoto & Whitman (2000) have begun reopening the debate, offering some little (but quite valuable) specific contribution to the analysis of syntactic amalgamation. Also important is the recent work by van Riemsdijk (2000, 2001) on free relatives, where syntactic amalgams are briefly mentioned for the sake of comparison. The influence of van Riemsdijk's work in this dissertation is obvious, as the notion of multiply-rooted phrase markers plays a crucial role in my analysis of amalgamation as it does in his analysis of what he takes to be similar constructions.[3]

A typical first reaction to some examples of syntactic amalgamation is to dismiss the body of facts as a 'pragmatic effect that overrides grammar', or a 'performance anomaly', or some sort of 'periphery effect', whatever that means.

[2] According to Newmeyer (1996: 141), Lakoff's (1974) paper is *"full of what a current reader would take to be a self-congratulatory gloating at having uncovered linguistic problems that no theory yet devised have succeeded in treating adequately"*.

[3] Yet another work in which syntactic amalgamation is taken seriously is Kuroda (2000), which I will not discuss in this dissertation because (i) it does not provide any new empirical generalization beyond what was already done by Lakoff (1974); and (ii) its *desiderata* pose serious incommensurability issues, as it is founded on radical connectionist-ish assumptions of total denial of all foundational concepts of Generative-Transformational Grammar.

One of my main goals in this dissertation is to scrutinize the structure of syntactic amalgams and attempt to characterize it as, essentially, a byproduct of independently motivated mechanisms of core syntax. My take on the mysterious and highly complex nature of amalgamation is that, instead of labeling these facts upfront as 'extra-grammatical' artifacts of some sort, it is wiser to capitalize on this puzzling phenomenon to explore the (potential) limits of the Theory of Grammar to see how far it may go. If we confine our descriptive, analytical and theoretical tools to the immediate horizon of apparently 'well behaved' sentences, we may be missing something quite deep about the Language Faculty.

However, the exploration of broader horizons is no guarantee that amalgamation indeed belongs in the domain of core grammar. There is simply no pre-theoretical line drawn between syntax and parataxis, grammar and parsing, competence and performance, core and periphery, phonology and phonetics, semantics and pragmatics. In the following chapters, I will be visiting some previously unexplored corners of the territory of UG and will partially (re)draw some of those boarder lines relatively to the body of facts pre-theoretically described as syntactic amalgamation.

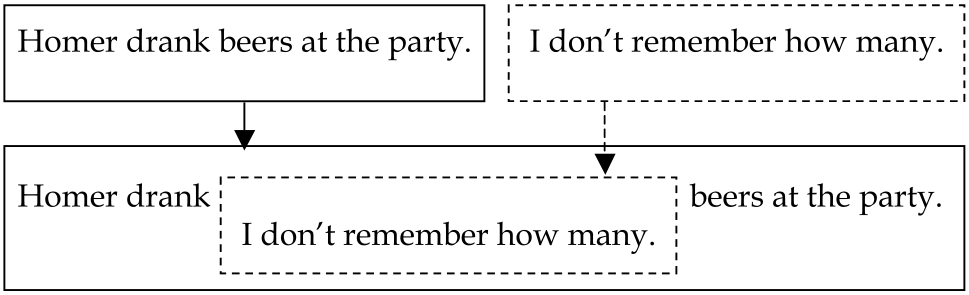
Biased by the *desiderata* of the Minimalist Program (Chomsky 1993, 1995, 2000, 2001a, 2001b; Martin & Uriagereka 2000; Uriagereka 1998, 2001, 2002) I will venture to stretch the limits of syntax as they are standardly assumed, and will ultimately claim that those boundary lines should be drawn so that most aspects of amalgamation pertain to the domain of core syntax.

As the first step towards this goal, I will show that syntactic amalgams exhibit some clear patterns. Moreover, I will claim that such patterns are grammatical in nature, as they are describable in terms of usual syntactic notions like c-command, locality, movement, pied-piping, economy of derivations, and the like.

Chapter **II** is dedicated to an extensive description of amalgamation, where I present a series of new empirical generalizations that I found, as well as the ones provided by Lakoff (1974) and Tsubomoto & Whitman (2000). Although I don't have any pretensions of doing an exhaustive and detailed comparative study, I will show some cross-linguistic observations (focusing on contrasts between English and Romance) that constitute evidence that the phenomenon of amalgamation is not restricted to the one particular language where amalgamation was first observed by Lakoff (i.e. English), and, moreover, the relevant differences correlate with independently motivated parametric choices.

Back to the example in (01), it is easy to see that, whatever the ultimate structure of this kind of construction is, we are clearly walking on the fine line between syntax and parataxis. Descriptively and Pre-theoretically speaking, what happens in constructions like (01) is that a sentence S_1 gets interrupted and 'invaded' by another sentence S_2, as shown in (02).

(02)

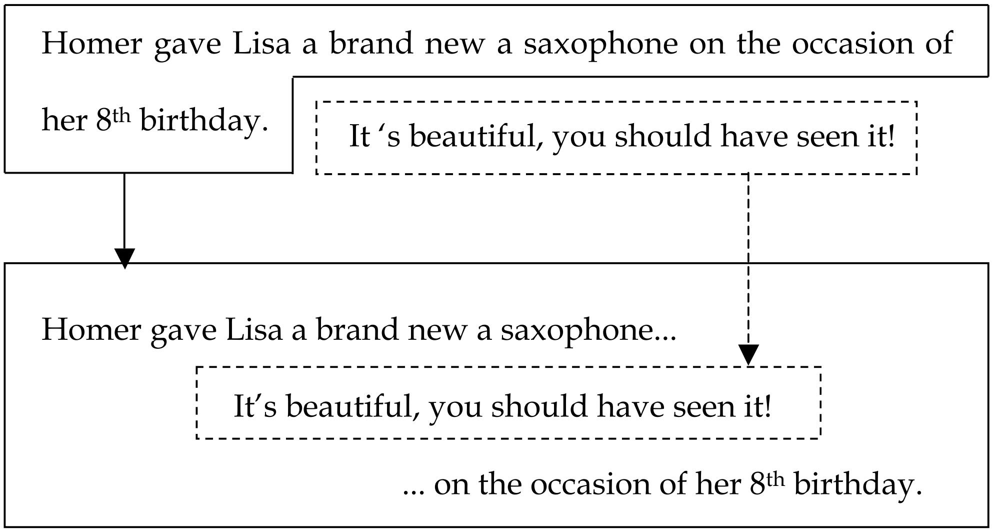


a: Homer drank beers at the party. I don’t remember how many.

b: Homer drank I don’t remember how many. beers at the party.

This suggests that syntactic amalgams are a subcase of parentheticals, where the same pattern obtains, as shown in (03).

(03)



a: Homer gave Lisa a brand new a saxophone on the occasion of her 8th birthday. It ‘s beautiful, you should have seen it!

b: Homer gave Lisa a brand new a saxophone... It’s beautiful, you should have seen it! ... on the occasion of her 8th birthday.

The picture is much more complicated than that, however. Unlike typical parentheticals, it is not immediately obvious which substrings of a syntactic amalgam count as the ‘invaded ‘and the ‘invasive’ sentences. In principle, either informal notation in (04) could be a plausible way of representing the ‘clause invasion’ going on in (01).

(04) a: Homer drank | I don’t remember how many | beers at the party.

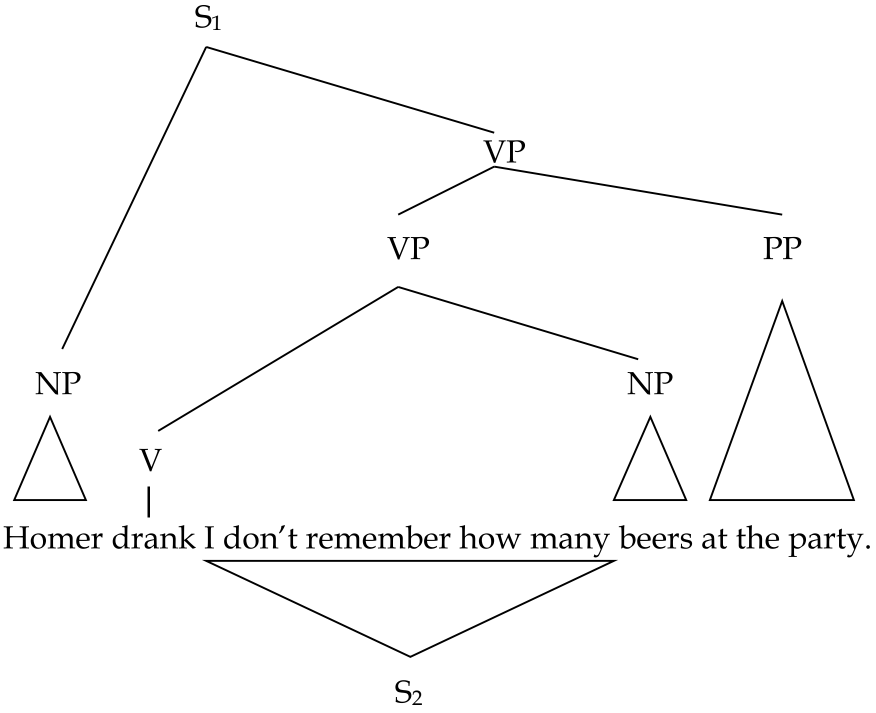
b: Homer drank | I don’t remember how many beers | at the party.

By (04a), the two input sentences would be the ones in (05), which would get paratactically combined as in (06).

(05) a: S_1 = Homer drank beers at the party.

b: S_2 = I don’t remember how many.

(06)

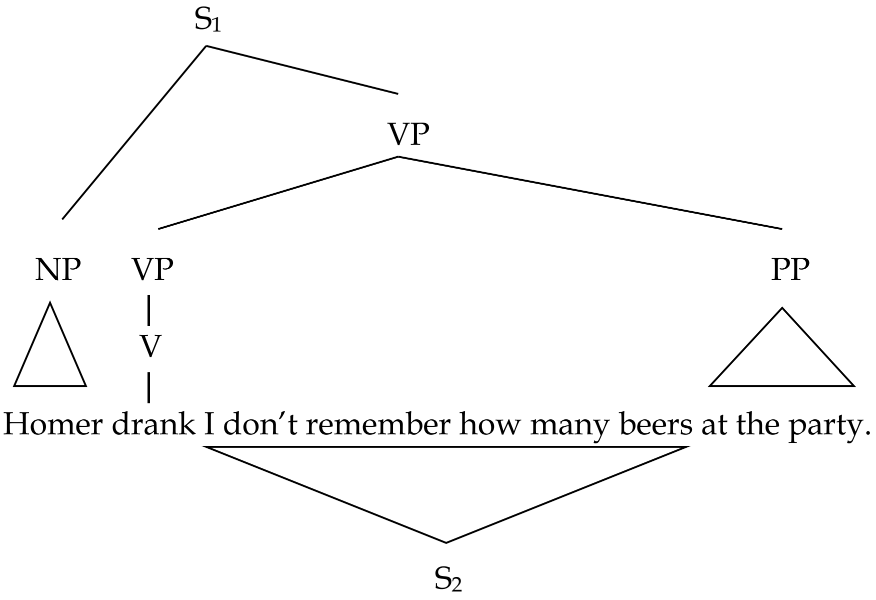


By (04b), the two input sentences would be the ones in (07), which would get paratactically combined as in (08).

(07) a: S_1 = Homer drank at the party.

b: S_2 = I don't remember how many beers.

(08)



Notice that, in (05)/(06), **drank** is treated as a regular transitive verb, taking **beers** as its complement within the domain of the 'invaded clause'; whereas the analysis sketched in (07)/(08) capitalizes on the possibility of **drank** being used intransitively, so that **beers** is a subconstituent of a more complex NP (i.e. **how many beers**) within the domain of the 'invasive clause'.

In either case, the 'invasive clause' seems somehow incomplete, as the verb **remember**, in the relevant reading, selects a full clause as its complement rather than an NP. After all, what the speaker doesn't remember is how many beers Homer drank at the party. This suggests that (01) may actually be a convoluted version of (09), as both share the very same propositional structure and truth conditions, although the their informational structures are not identical.

(09) I don’t remember [how many beers]$_1$ Homer drank t_1 at the party.

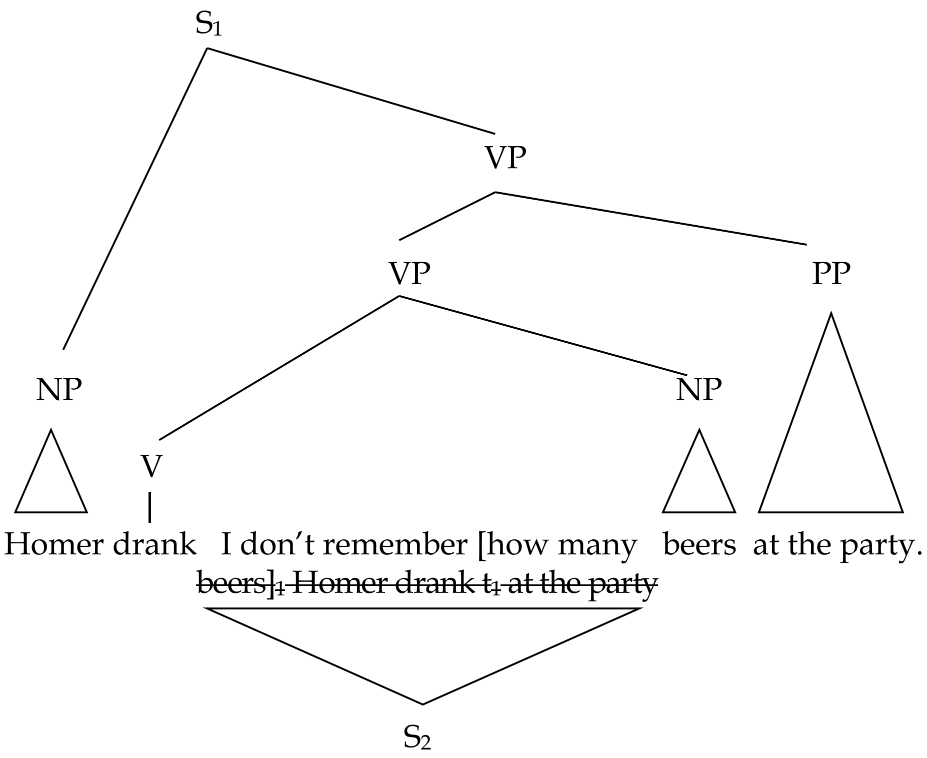
This might seem a little puzzling at first sight, as what we have been taking to be the main clause in (01) – i.e. **Homer drank (beer) at the party** – corresponds to a subordinate clause in (09). One way out of this puzzle is to assume that the syntactic representation of the invasive clause contains extra elliptical material that replicates the structure of the invaded clause. Therefore, syntactic amalgamation would reduce to a combination of sluicing (cf. Ross 1969; Merchant 2001) and parentheticalization.

From this perspective, the two alternative analysis sketched in (05)-(06) and (07)-(08) would be more accurately represented by (10)-(11) and (12)-(13), respectively.

(10) a: S_1 = Homer drank beers at the party.

b: S_2 = I don’t remember how many ~~beers Homer drank at the party~~.

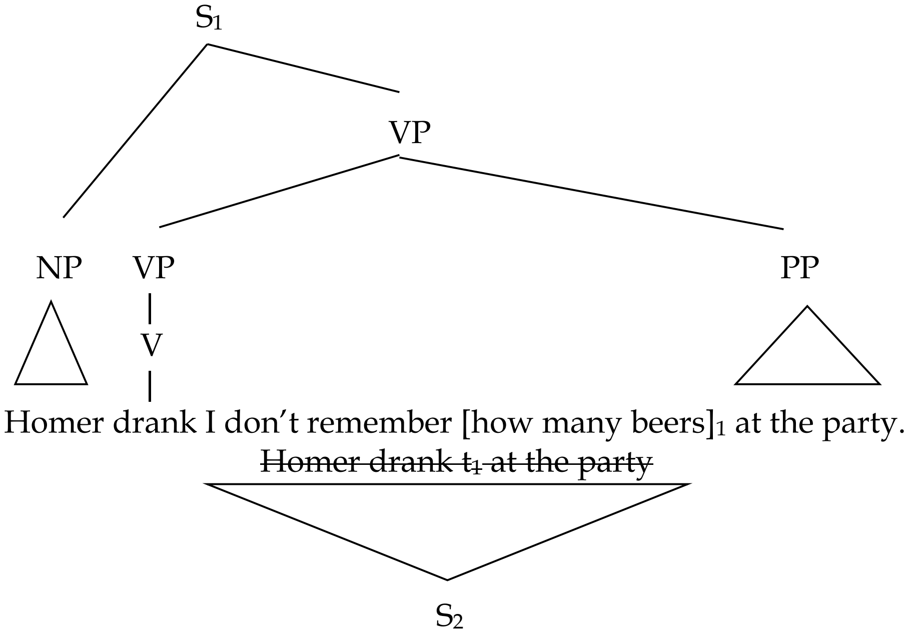
(11)


S1
VP
VP
PP
NP
NP
V
Homer drank I don’t remember [how many beers at the party.
~~beers]1 Homer drank t1 at the party~~
S2

(12) a: S_1 = Homer drank at the party.

b: S_2 = I don’t remember how many beers ~~Homer drank at the party~~.

(13)


S1
VP
NP
VP
V
PP
Homer drank I don’t remember [how many beers]1 at the party.
~~Homer drank t1 at the party~~
S2

Once invasive clauses are taken to involve internal sluicing, the null hypothesis is that they should pattern exactly like any other ordinary sluiced sentence in terms of which substrings get affected by whatever ellipsis mechanism is involved in the inner-workings of sluicing. This gives us a way of teasing apart the two alternative hypotheses in (4a) and (4b). The analysis in (11) – in which **beers** is the complement of **drank** – has an obvious advantage over the one in (13) – in which **drank** is taken to be intransitive –, since the pronounced and unpronounced substrings in (10)-(11) correspond exactly to what we obtain in the analogous case of sluicing without parentheticalization, as shown in (14a); whereas the pronounced and unpronounced substrings in (12)-(13) do not correspond to a grammatical structure in other instances of sluicing outside amalgams, as shown in (14b).

(14) a: Homer drank beers at the party, but I don't remember [how many ~~beers~~]$_1$ ~~Homer drank t_1 at the party~~.

b: * Homer drank at the party, but I don't remember [how many beers]$_1$ ~~Homer drank t_1 at the party~~.

Not surprisingly, the same reasoning extends to similar cases where the main verb of the invaded clause is transitive and does not have an intransitive analog, as in (15).

(15) Homer gave you'll never guess how much money to Lisa.

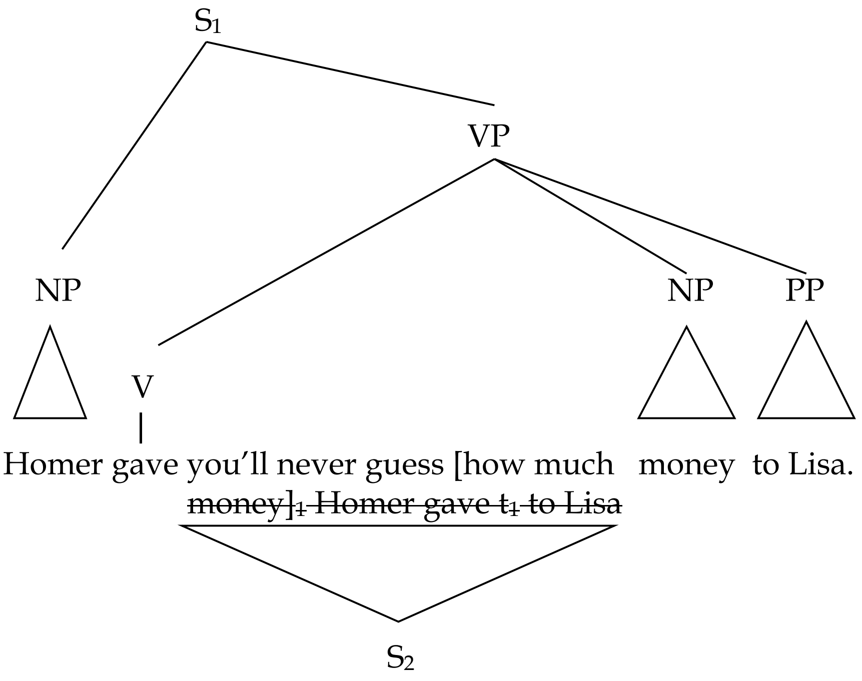
Notice that, in (16), the non-amalgamated version of (15) exhibits the same sluicing pattern predicted by the analysis in (17) — i.e. the pronounced substring ends with **how much**) —, which follows the same logic as the analysis in (11) for (01).

(16)[4] a: Homer gave money to Lisa. You'll never guess [how much ~~money]$_1$~~ ~~Homer gave t_1 to Lisa~~.

b: * Homer gave money to Lisa. You'll never guess [how much money]$_1$ ~~Homer gave t_1 to Lisa~~.

c: * Homer gave to Lisa. You'll never guess [how much money]$_1$ ~~Homer gave t_1 to Lisa~~.

(17)

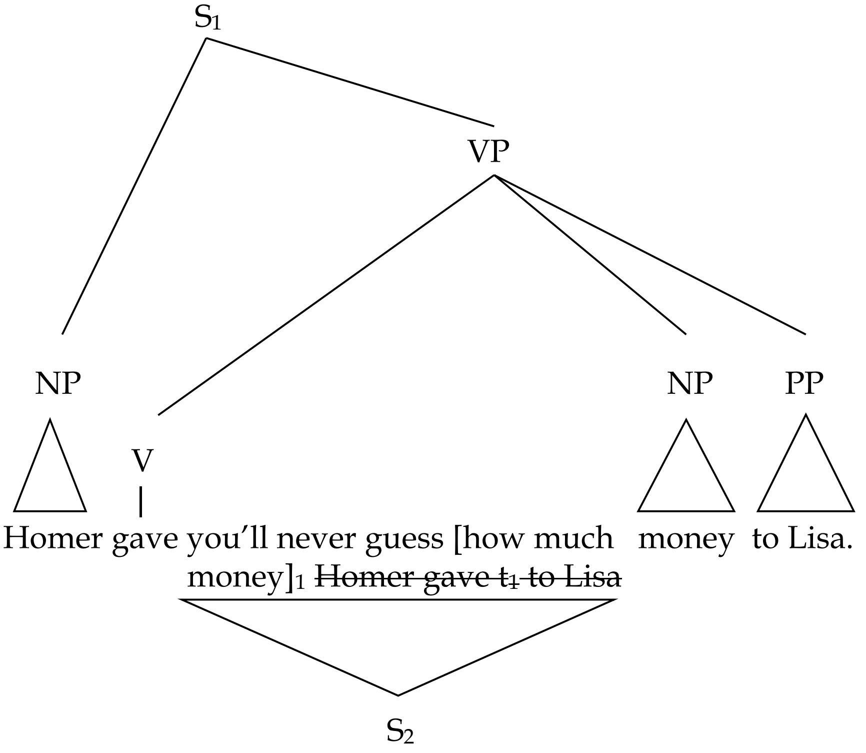

S1
VP
NP
V
NP
PP
Homer gave you'll never guess [how much
money
to Lisa.
~~money]1 Homer gave t1 to Lisa~~
S2

[4] The judgment reported in (16) relates to the default prosodic pattern. An amelioration effect arises if, in the second sentence, **how much** receives contrastive stressed, while **money** is distressed. For a discussion on destressing/deaccenting related to sluicing, see Merchant (2001).

Just like its non-amalgamated version in (16b), the amalgam in (18) – structured as in (19) – is ungrammatical by virtue of it having **money** (instead of **how much**) as the last element of the pronounced substring.

(18) * Homer gave you'll never guess how much money money to Lisa.

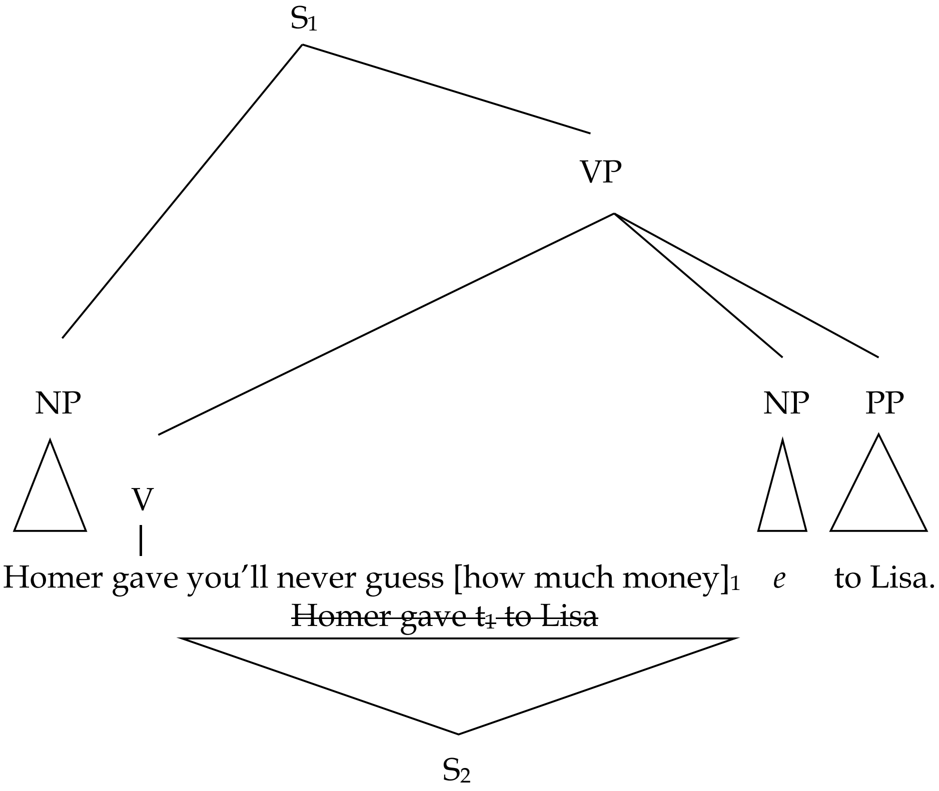
(19)


S1
VP
NP
V
NP
PP
Homer gave you'll never guess [how much money]1 ~~Homer gave t1 to Lisa~~
money
to Lisa.
S2

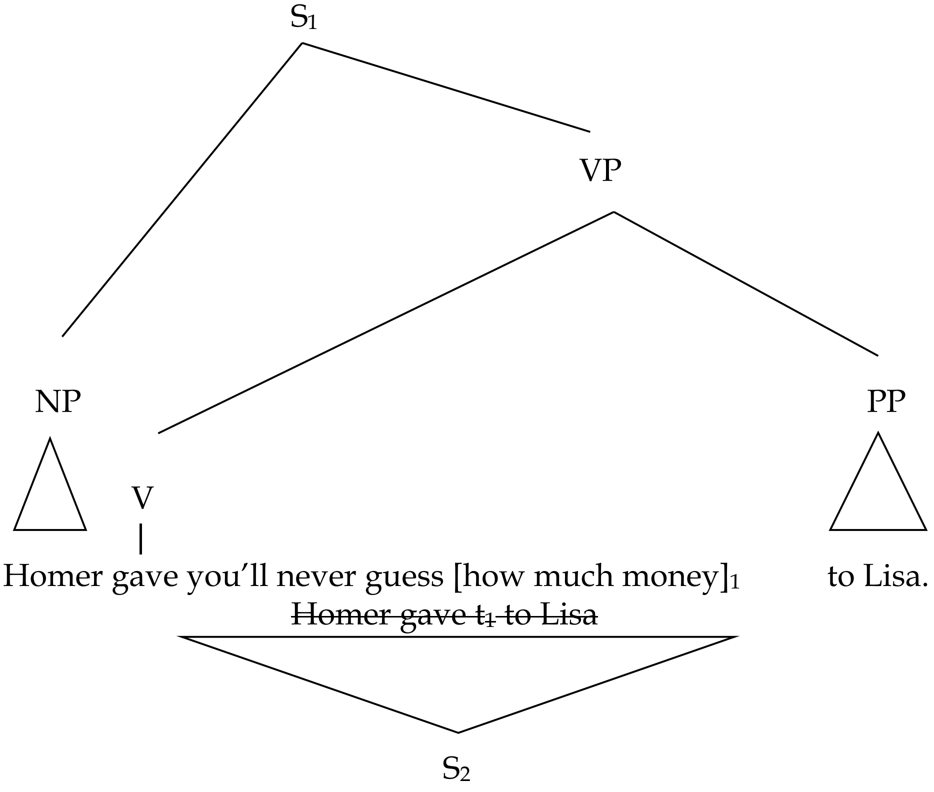
In this context, one could stipulate that the acceptable example in (20) is structured as in (21)/(22), where the parenthetical clause exhibits the same form of sluicing as in (19), and the direct object of (the pronounced token of) **gave** is missing (either because it is instantiated by an empty category in an *ad hoc* construction-specific fashion, as in (21), or because it is simply absent despite the usual theta-theoretical requirements, as in (22)).

(20) Homer gave you'll never guess how much money to Lisa.

(21)


S_1
VP
NP
NP
PP
V
Homer gave you'll never guess [how much money]$_1$ *e* to Lisa.
~~Homer gave t_1 to Lisa~~
S_2

(22)


S_1
VP
NP
PP
V
Homer gave you'll never guess [how much money]$_1$ to Lisa.
~~Homer gave t_1 to Lisa~~
S_2

However, the same (*ad hoc*) reasoning above cannot be extended to equivalent cases differing only with respect to the internal structure of the WH-phrase. That is, cases where the object is a bare WH-phrase (e.g. **<u>what</u>**, **<u>who</u>**) instead of a complex one (e.g. **<u>how much money</u>**, **<u>how many beers</u>**).

For instance, compare (15) above to (23) below.

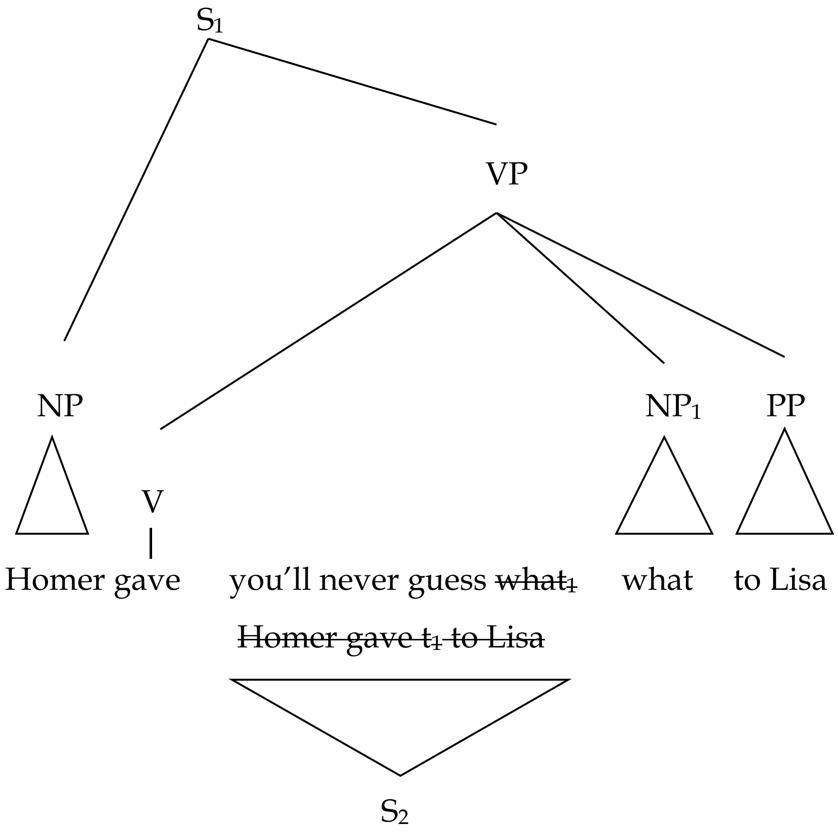
(23) Homer gave you'll never guess what to Lisa.

I have just shown how examples like (15) can successfully be analyzed as in (17), where the object of **<u>gave</u>** is simply **<u>money</u>** rather than a WH-phrase, while what occupies the spec/CP of the in the sluiced sentence is the complex WH-phrase **<u>how much money</u>**, which gets pronounced simply as **<u>how much</u>** due to sluicing. Thus, descriptively speaking, the complex WH is split across the invaded and the invasive clauses.

On the other hand, such splitting of the WH-phrase is impossible in examples like (23). This forces us to analyze those examples differently, by postulating a syntactic representation quite distinct from the one assumed in (17) for (15), and from the one assumed in (11) for (01).

One way to go about (23) would be to postulate a radical kind of sluicing internally to the invaded clause, where even the WH-phrase would be in the unpronounced substring, while the theta-theoretical requirements in the invaded clause would be satisfied by a distinct token of **<u>what</u>**, as the direct object of **<u>give</u>**. This is shown in (24).

(24)



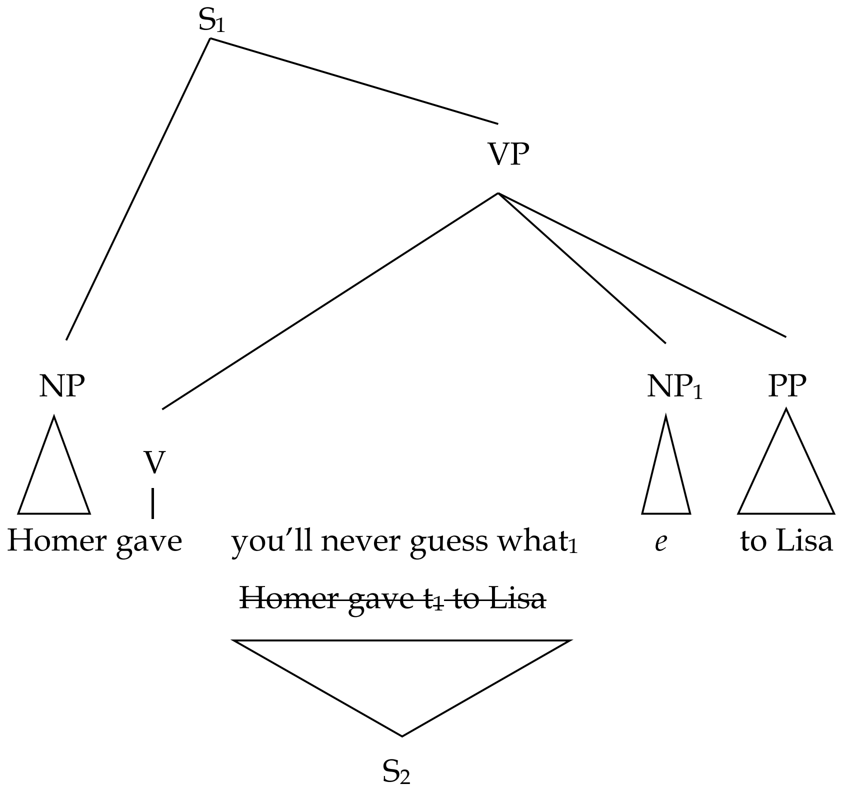
There are two immediate problems with this analysis: (i) an *ad hoc* instance of WH in situ is stipulated for the invaded clause **Homer gave ... what to Lisa**; and (ii) an *ad hoc* case of 'radical sluicing' is stipulated for the invasive clause, where even the WH phrase does not get pronounced.[5] Empirical evidence that such formal devices are *ad hoc* is shown in (25), where they are unsuccessfully applied to the non-amalgamated version of (23).

(25) * Homer gave what to Lisa. You'll never guess ~~what$_1$ Homer gave t$_1$ to Lisa~~.

[5] Alternatively, we could consider that the WH-phrase in the embedded spec/CP of the invasive clause is a pronominal empty category to begin with. That would run into similar conceptual problems, as the presence of such pronominal empty category only in invasive clauses of amalgams would be as *ad hoc* as the 'radical sluicing' in (24), and the same empirical problem posed by (25) would be faced.

Another possible analysis for (23) is the one sketched in (26).

(26)



In this structure, the theta-theoretical requirements of **<u>gave</u>** in the invaded clause are satisfied by an empty category, while the invasive clause is affected by standard sluicing.

This analysis is problematic in face of the unacceptability of (27), which shows that such hypothesized empty category is not licensed in a non-amalgamated version of (23), hence not independently motivated.

(27) * Homer gave [$^{\text{NP}}$ *e*] to Lisa. You'll never guess what$_1$ ~~Homer gave t_1 to Lisa~~.

Any solution to this problem would have to capitalize on the intuition that (23) is somehow the amalgamated version of (28), where the object of **gave** in the non-sluiced sentence is an overt indefinite pronoun co-indexed with the WH-phrase in the sluiced sentence.

(28) Homer gave something$_1$ to Lisa. You'll never guess what$_1$ ~~Homer gave t$_1$ to Lisa~~.

From that perspective, the hypothesized empty category in (26) would be an elliptical version of the indefinite pronoun in found in (28), which presumably undergoes PF-deletion under certain circumstances. The mystery, though, is how to define those circumstances without falling into *ad hoc* construction-specific mechanisms, in order to avoid the overgeneration of examples like (29).

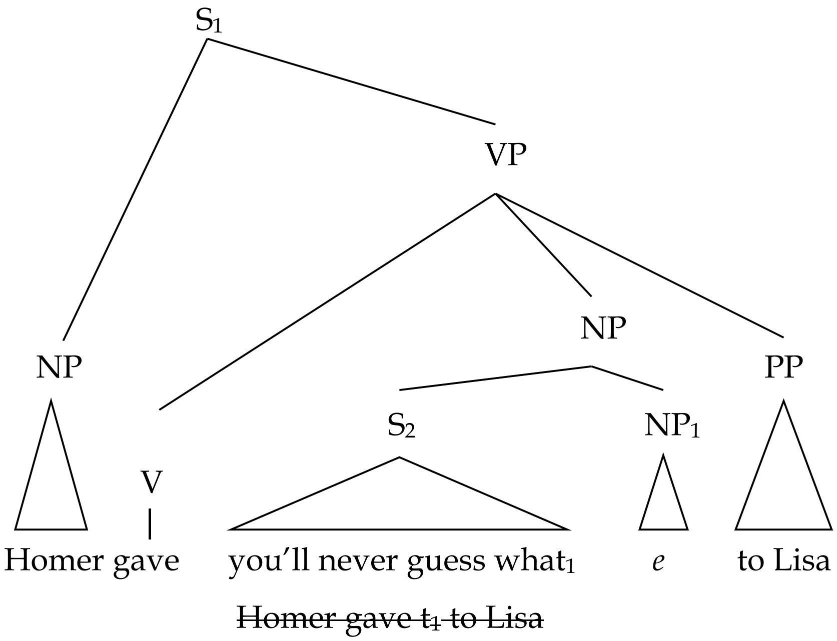
(29) * Homer gave *you'll never guess what$_1$* ~~*Homer gave t$_1$ to Lisa*~~ something$_1$ to Lisa.

Therefore, both analyses in (24) and (26) are problematic in themselves, but the most serious issue is that neither one can be reconciled with the formalism assumed in (11) and (17) for the cases in (01) and (15) respectively. In a nutshell, this 'sluicing inside the parenthetical' approach fails to give a unified account to the phenomenon of amalgamation, even if we restrict our scope to the few cases mentioned above (not to mention if we consider the full range of facts described in Chapter **II**).

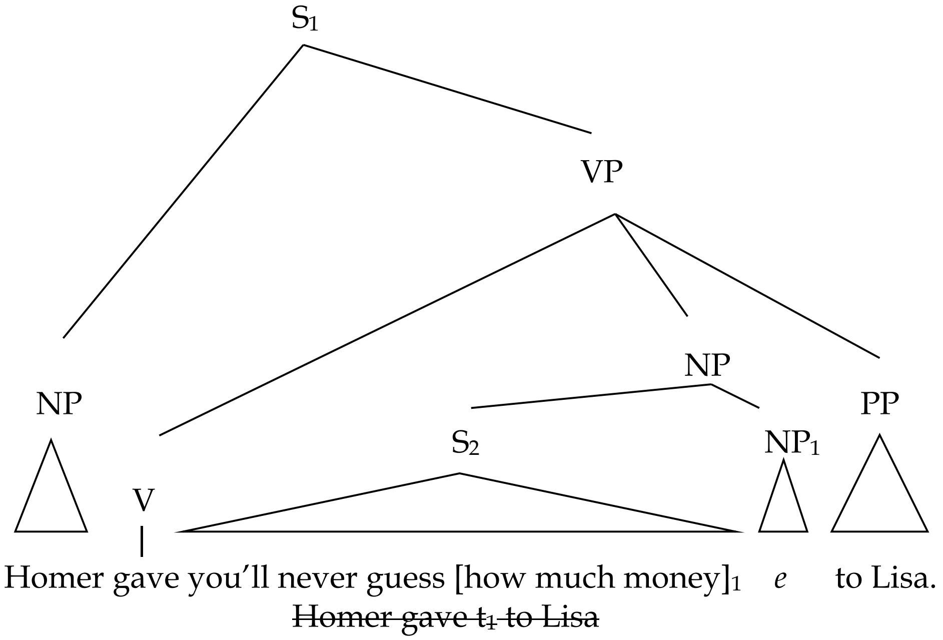
Tsubomoto & Whitman (2000) have proposed a formalism along the lines of (26) as a general theory of amalgamation, inspired by Lakoff's classical analysis, which, in its turn, was a development of an original idea by William Cantral. Thus, the invaded clause would universally contain an elliptical NP co-indexed with a WH-phrase inside the invasive clause, which undergoes internal sluicing, just as in (26). There is, however, a significant difference between the analysis sketched in (26) and Tsubomoto & Whitman's (2000) proposal. The former approach takes the invasive clause to be a parallel independent sentence at the syntactic level, which gets paratactically combined with the invaded clause through some non-trivial (re)linearization process. The latter approach takes clause invasion to be ordinary clause embedding, so that the invaded clause would be the matrix clause, while the invasive clause would be subordinated to it by being adjoined to the elliptical NP, as sketched in (30), (31) and (32).[6]

[6] In fact, Tsubomoto & Whitman (2000) do not explicitly discuss any case involving bare WH-phrases or potentially intransitive verbs, but (30) and (32) naturally follow from their formalism.

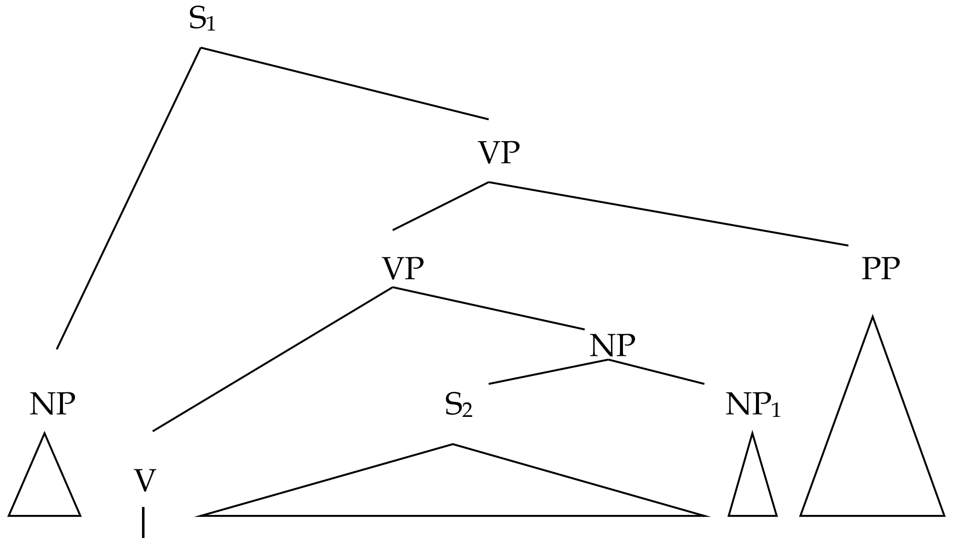
(30)


S1
VP
NP
NP
PP
S2
NP1
V
Homer gave
you'll never guess what1
e
to Lisa
~~Homer gave t1 to Lisa~~

(31)


S1
VP
NP
NP
PP
S2
NP1
V
Homer gave you'll never guess [how much money]1
e
to Lisa.
~~Homer gave t1 to Lisa~~

(32)



Homer drank I don’t remember [how many beers]$_1$ *e* at the party.
~~Homer drank t_1 at the party~~

An interesting feature of Tsubomoto & Whitman’s (2000) analysis is the intuition that the paratactic relation between the invaded and the invasive clauses is ultimately syntactic in its essence, so that the core structure and the parenthetical are somehow connected by dominance relations at some point in the structure. I share this general view, which I will be arguing for in the remainder of this dissertation. However, I advocate for a radically different view on how such syntactic connection is established derivationally and representationally.

In chapter **III**, I discuss Lakoff’s (1974) classical analysis and Tsubomoto & Whitman’s (2000) recent proposal in detail, concluding that, despite obfuscatory and misleading superficial effects, syntactic amalgamation does not involve any sluicing. Upon closer scrutiny, the sluicing-based approaches turn out to be descriptively and explanatorily inadequate, as they make wrong predictions with regards to the empirical facts presented in chapter **II**; and as they crucially rely on (i) a construction-specific sluicing mechanism for the invasive clause, and (ii) a construction-specific NP-ellipsis mechanism for the invaded clause.

The remainder of chapter **III** consists of an attempt to address the problems faced by Lakoff's (1974) and Tsubomoto & Whitman's (2000) analyses through standard theoretical tools, like clause-topicalization and remnant-movement; or clause-topicalization and scattered deletion of copies. The conclusion is that such minor tweaking is not enough, as major conceptual and empirical problems remain. Thus, the theory of UG needs to undergo more substantial revision in order to account for syntactic amalgamation.

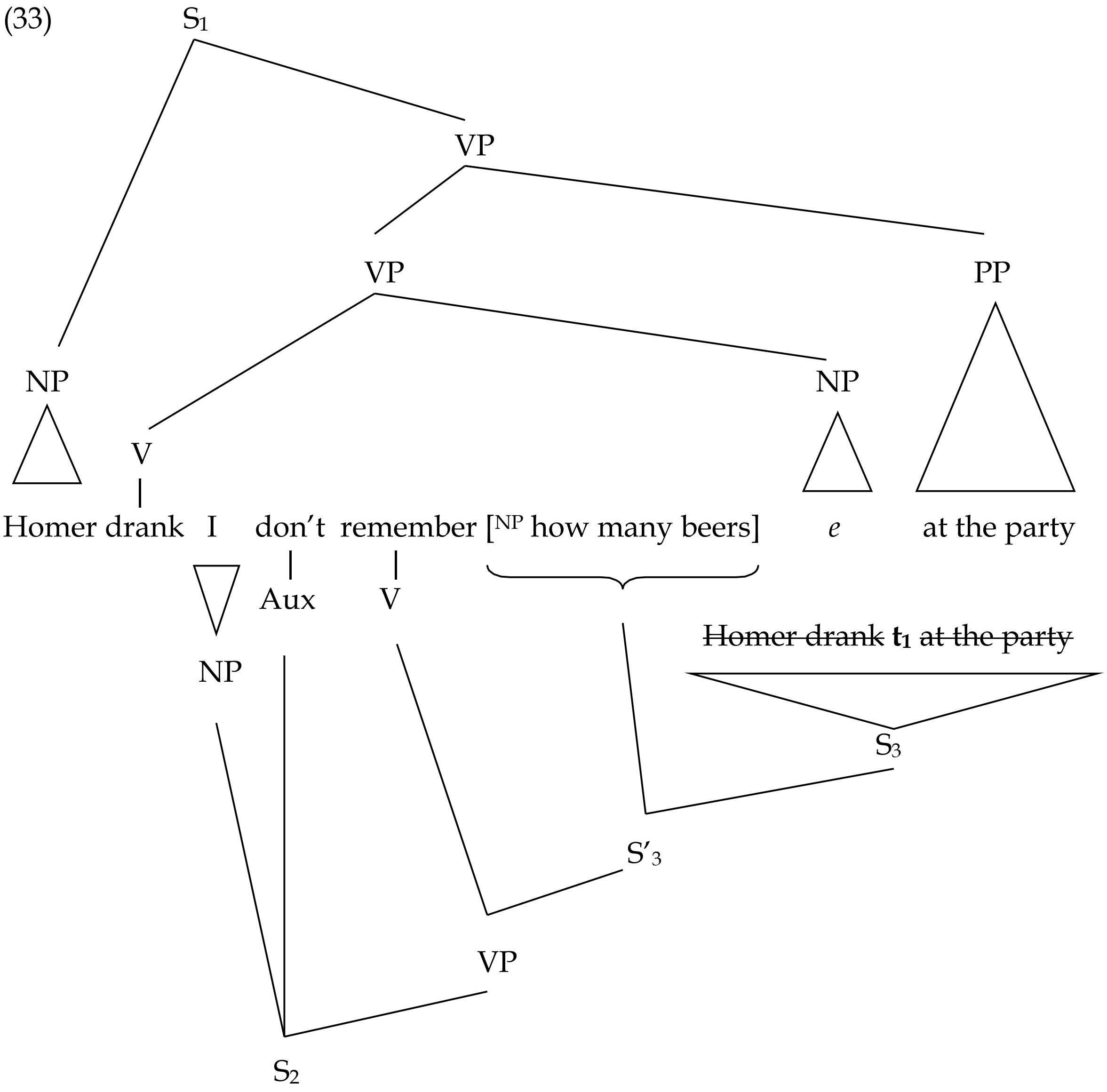
In chapter **IV**, I set the stage for presenting my sluicing-free analysis of syntactic amalgamation afterwards. I discuss the nature of the fundamental notions of syntax: (i) the integration mechanism (i.e. *merge*); (ii) the input (i.e. the *lexical array*, or *numeration*); (iii) the displacement property (i.e. *movement*); and (iv) the derivationalism versus representationalism debate. All this discussion is carried out vis-à-vis all the relevant concepts that pertain to the minimalist *desiderata* (optimal design, economy of derivations and representations, reduction of computational complexity, dynamic derivations with cyclic access to the interfaces, interface-driven requirements (bare output conditions), and the like). This discussion, then, culminates with the proposal of a specific model of syntax that incorporates some rather non-standard assumptions about the architecture of UG, which nonetheless interact in a strictly minimalist fashion, meeting requirements of optimal design. Such a framework is briefly summarized in **I.2** below.

In chapter **V**, I finally present my analysis of syntactic amalgamation, which is built from the theoretical assumptions discussed in the previous chapter. The essence of my proposal for the representation of syntactic amalgams is as follows.

Back to (01), let us hypothesize, for a moment, that its structure is as in (33). Let us take that as the starting point, and follow the reasoning below, modifying the analysis step-by-step.

In order to achieve a unified analysis for all three cases discussed so far, the representation in (33) is built without 'WH-phrase splitting' (so as to be compatible with cases exhibiting **what** instead of **how many beers**), so that the whole WH-phase is contained in the invaded clause while an empty category satisfies the theta-theoretical requirements of **drank** in the invaded clause, treated as a transitive verb (so as to be compatible with cases exhibiting *bona fide* (di)transitive verbs like **understand** or **give**, instead of 'hybrid' verbs like **drink**).

(33)

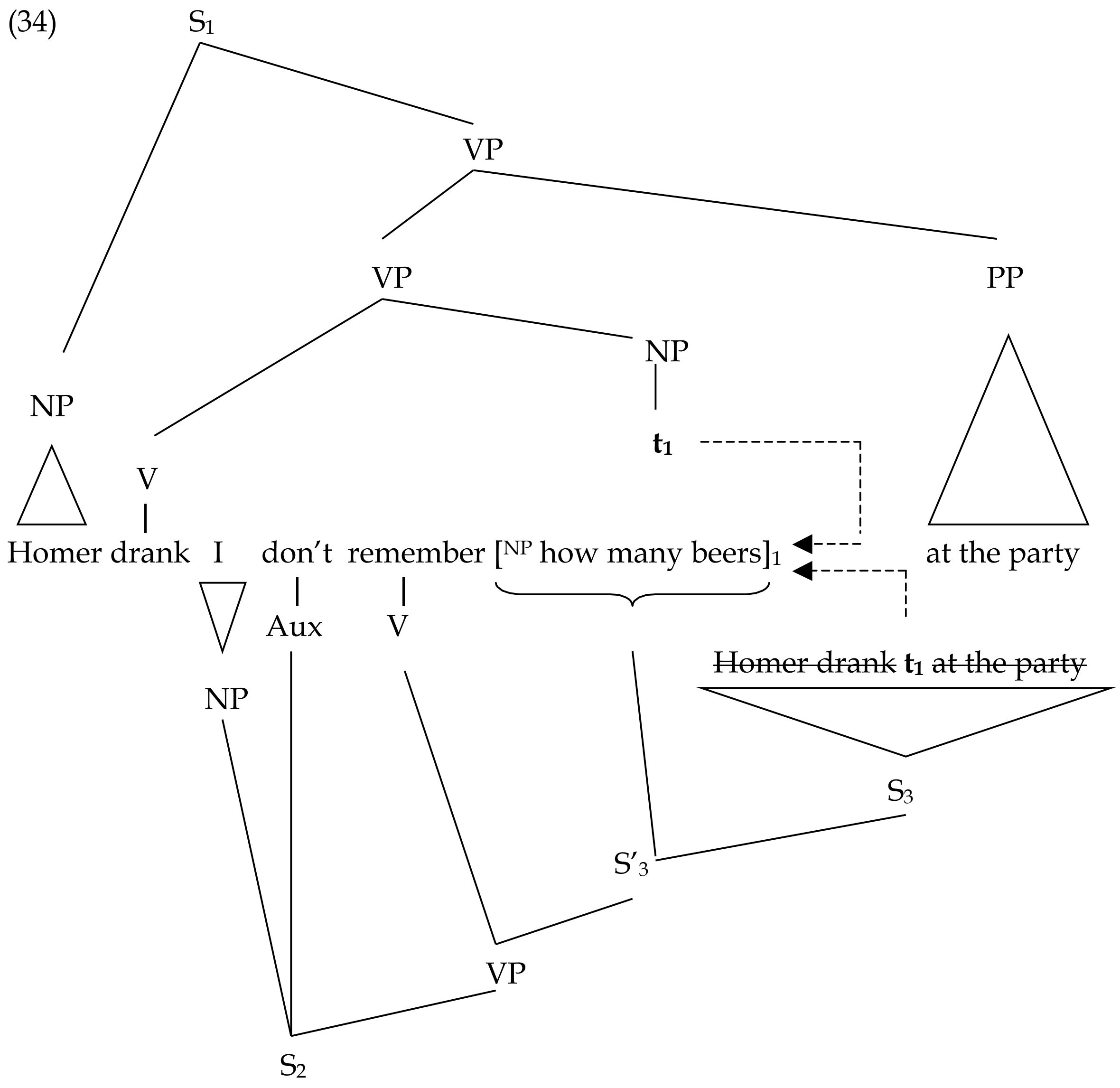

S1
VP
VP
PP
NP
NP
V
Homer drank I don't remember [NP how many beers] e at the party
Aux
V
NP
Homer drank t1 at the party
S3
S'3
VP
S2

This is essentially a version of Tsubomoto & Whitman's (2000) analysis where the invasive clause is taken to be an independent parallel sentence, rather than an embedded clause adjoined to the empty category in the invaded clause. Informally speaking, we may say that the empty category in the invaded clause is related to the rest of the structure in a way that is somewhat analogous to how a parasitic gap is licensed, with the crucial difference that this empty category of amalgams is not c-commanded by the WH-phrase that seems to act as its 'antecedent'.[7]

My position is that this intuition is on the right track, but cannot be taken literally, as it would conflict with familiar properties of parasitic gap constructions (cf. Engdahl 1981; Chomsky 1982: 36-78; Culicover & Postal 2001). Ultimately, I propose that the empty category under discussion is actually a trace of movement, rather than a parasitic gap. As a step towards what I take to be the actual structure of (01), consider, for a moment, the representation in (34), where the WH-phrase is simultaneously the head of two parallel chains, as if two distinct tokens of that WH-phrase were generated each one in a distinct theta-position (one in the invaded clause, and the other in the invasive clause) and then collapsed into a single token of that WH-phrase when both simultaneously moved to the very same COMP position of the sluiced clause, as shown in (34).

[7] The same lack of c-command is true of Tsubomoto & Whitman's (2000) analysis, where the sluiced sentence is adjoined to the empty category in the invaded clause.

(34)

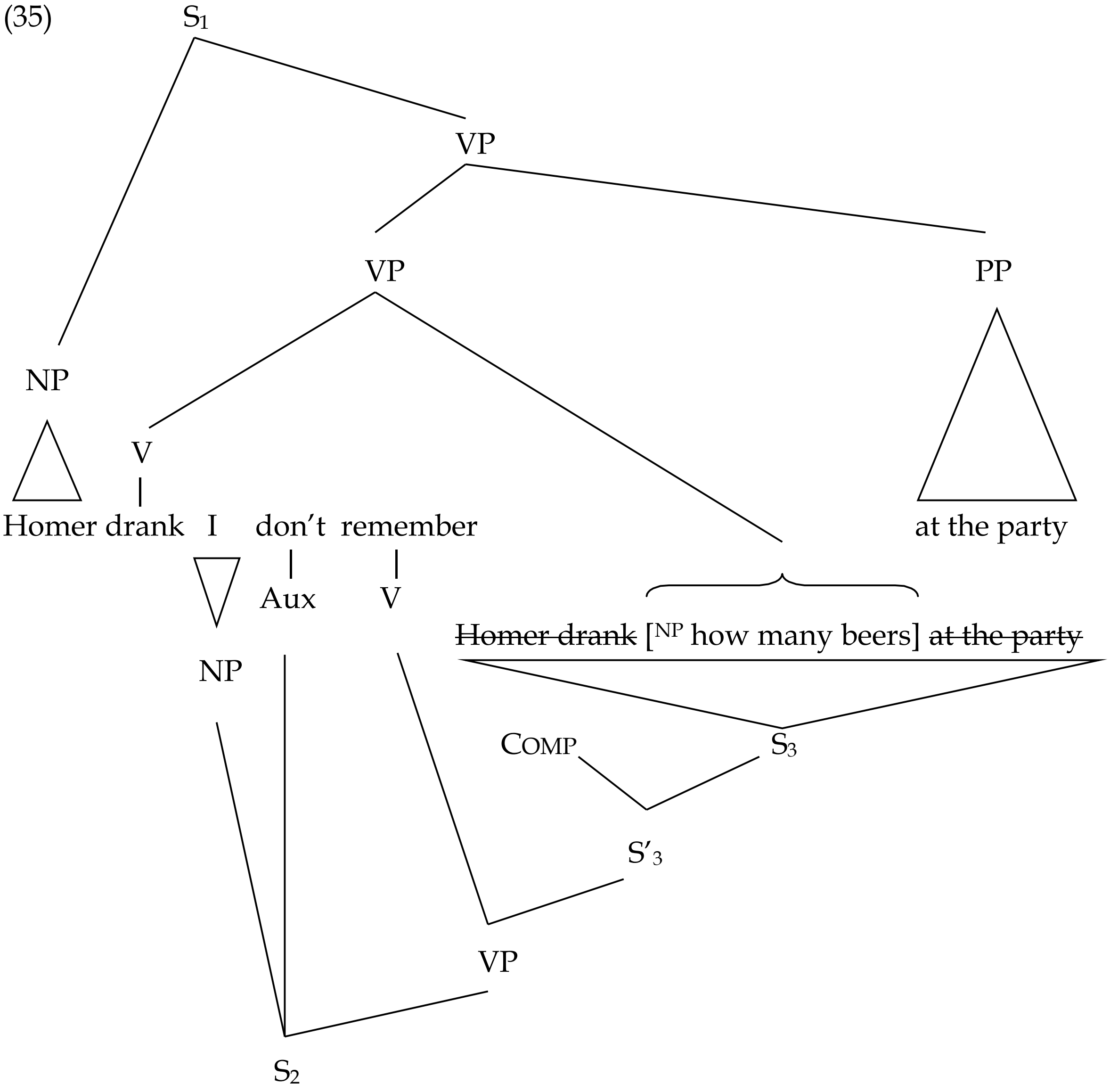


S1
VP
VP
PP
NP
NP
t1
V
Homer drank I don't remember [NP how many beers]1
at the party
Aux
V
NP
~~Homer drank~~ t1 ~~at the party~~
S3
S'3
VP
S2

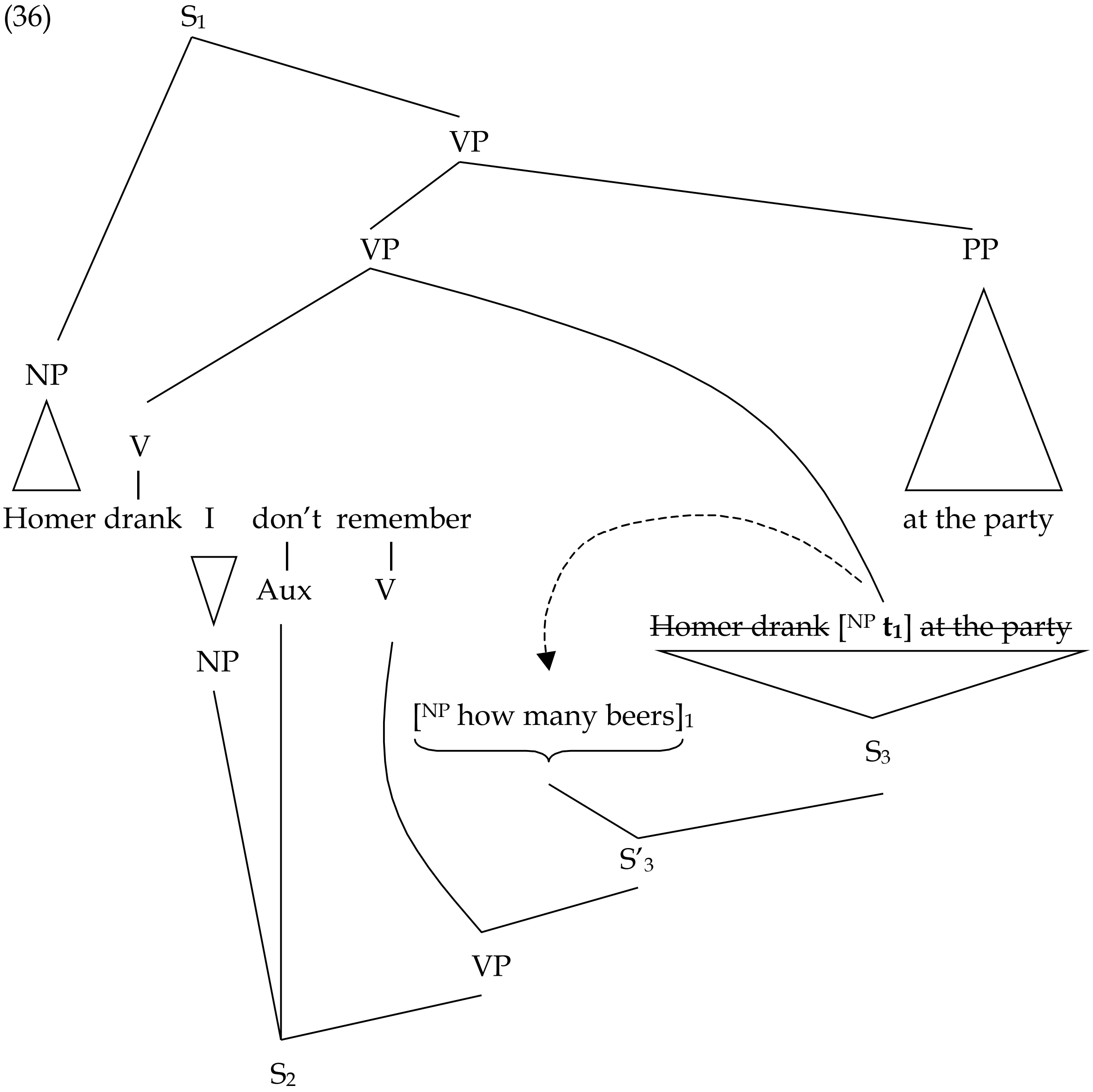
Needless to say, not only is this very chain-collapsing mechanism nontrivial in itself, but also one of the chains involved does not fit into the standard definition of chain as its trace is not c-commanded by the corresponding moved phrase. One way to go about this would be to deny those particular constraints on chains altogether. But, all else being equal, that has the unwelcome consequence of leaving the theory unable to capture many well-known and better-understood generalizations about chains (c-command, locality, and the like), in detriment of a construction-specific formalism designed to deal with syntactic amalgamation.

Nevertheless, in this particular case, there is indeed a way of having the cake and eating it too. We may capitalize on the notion of 'shared constituency' (through remerge and multi-motherhood), along the lines of what Citko (2002) proposed for Across-The-Board Extraction (ATB) and Free Relatives. Thus, abstracting away from strictly-cyclic derivations and the extension requirement, the relevant derivational step would involve a movement operation that takes (35) as the input, generating (36) as the output.

(35)

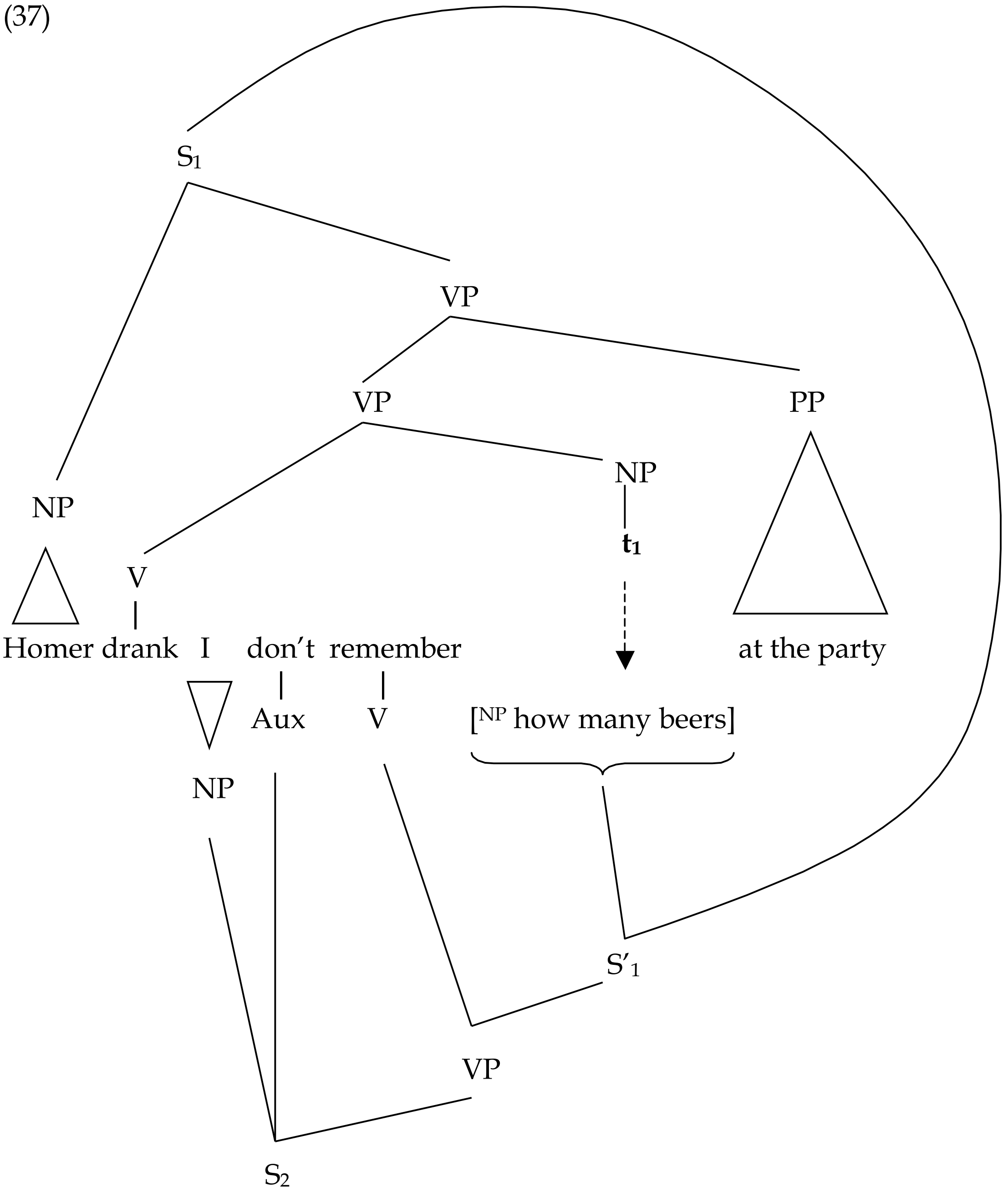

S_1
VP
VP
PP
NP
V
Homer drank I don't remember
at the party
Aux
V
~~Homer drank~~ [^{NP} how many beers] ~~at the party~~
NP
Comp
S_3
S'_3
VP
S_2

(36)


S1
VP
VP
PP
NP
V
Homer drank I don't remember
at the party
Aux
V
NP
~~Homer drank~~ [NP t1] ~~at the party~~
[NP how many beers]1
S3
S'3
VP
S2

In chapter **V**, I provide empirical evidence and conceptual arguments in favor of pushing this logic of shared constituency to the limit, so that the constituent that gets shared is not just the WH-phrase, but the whole clause containing its trace, as in (37). That way, we eliminate sluicing altogether, getting rid of all the problems related to *ad hoc* forms of ellipsis.

(37)



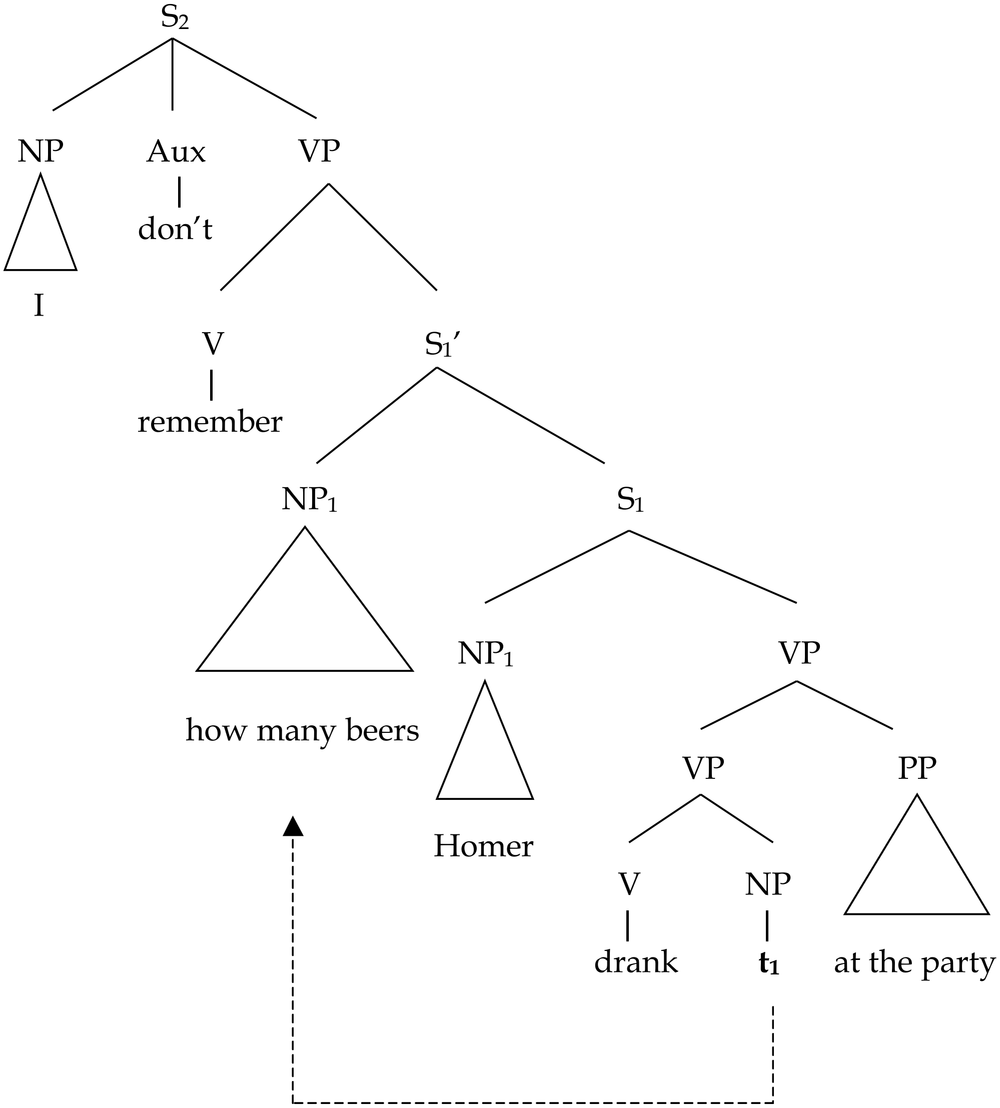
This analysis essentially treats (01) – repeated below as (38) – as a convoluted version of (39). This is intuitively appealing, as both structures share the same propositional content (despite differences in informational structure).

(38) Homer drank I don’t remember how many beers at the party.

(39) I don’t remember [how many beers]$_1$ Homer drank t$_1$ at the party.

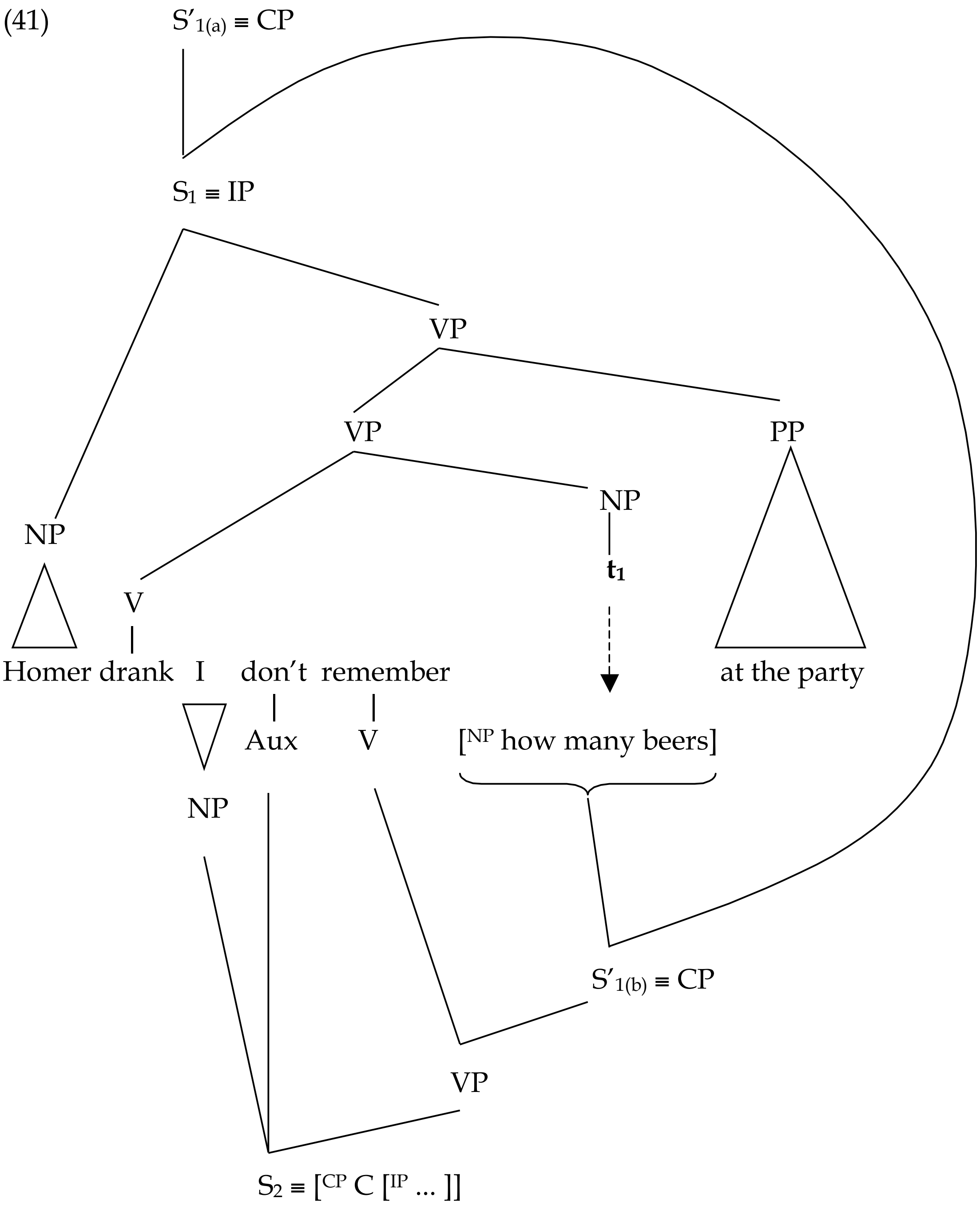
Without any further adjustments, the structure sketched in (37) is identical – abstracting away from word order – to the more familiar notation in (40), which is the standard way of analyzing (39).

(40)



In principle, the process that generates (38) from (39) can be conceived either as a complex paratactic operation that somehow ‘warps’ the phrase marker in (40) into the one in (37) – having a ‘relinearizing’ effect on the PF-string –; or as a more ordinary combination of movement transformations that apply to (40), yielding something other than (37). The latter possibility is discarded in chapter **III** on the basis of both conceptual arguments and cross-linguistic empirical evidence. In chapter **V**, the former possibility is shown to be incompatible with the range of facts presented in chapter **II**, and arguments are given in favor of an analysis that implicates a multiply-rooted phrase marker, as in (36), but involving shared-constituency (rather than sluicing), as in (37). From that perspective, the structure for the example under discussion would be as sketched in (41), where the invaded clause **Homer drank t_1 at the party** (S ≡ IP) is simultaneously embedded inside the invasive clause, and inside an extension of itself (S′ ≡ CP).

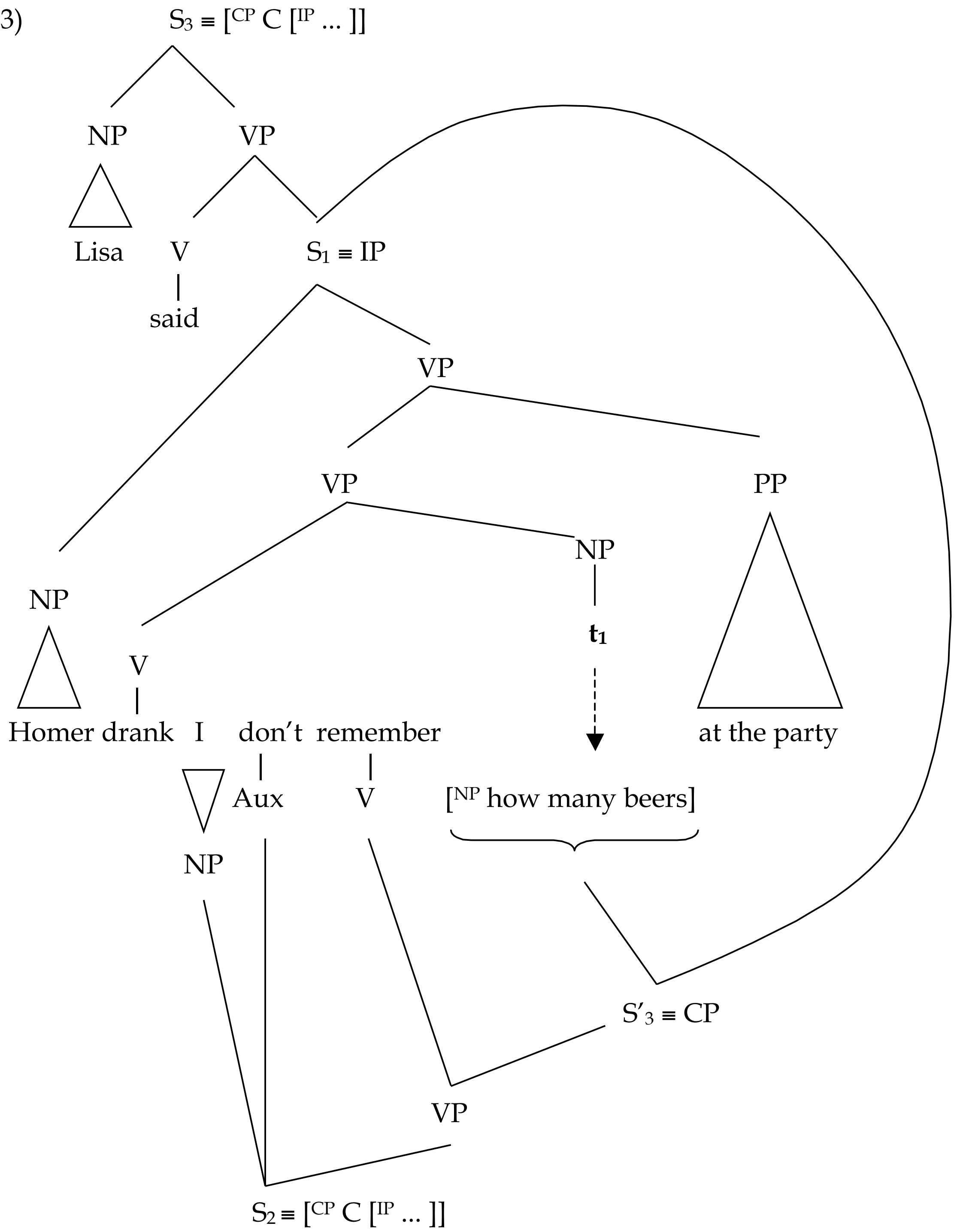
(41)



Evidence for this comes from examples such as (42), which would correspond to the structure in (43).

(42) Lisa said Homer drank I don′t remember how many beers at the party.

(43)



The crucial property of (42) is that, while the event of Homer having drank a certain quantity of beers at the party is simultaneously the theme of the saying event performed by Lisa and the not-remembering event/state experienced by the speaker, the not-remembering and the saying events are independent from one another.

In a context where (42) is true, Lisa did not say anything about the speaker not remembering how many beers Homer drank at the party. All Lisa said is that Homer drank a certain number of beers at the party. Conversely, it is not the case that the speaker does not remember Lisa having said how many beers Homer drank at the party.

All the speaker doesn't remember is the cardinality of the number ***x*** such that Homer drank ***x*** beers at the party (as opposed to the cardinality of the number ***y*** such that Lisa said that Homer drank ***y*** beers at the party).

This motivates an analysis along the lines of (43), where the sentence S_1 expressing the drinking event is simultaneously embedded within the sentence S_2 expressing the not-remembering event/state, and within the sentence S_3 expressing the saying event, with no subordination relation taking place between S_2 and S_3, which stand as parallel matrix clauses.

For consistency, I propose to extend this logic of multiply-rooted phrase markers to simpler cases like (38). Thus, I take the sentence expressing the drinking event to be simultaneously an embedded clause inside the sentence that expresses the not-remembering event/state, as well as a matrix clause, as in (41). In essence, the speaker who utters (38) is making two parallel statements: (i) the statement that Homer drank a certain number of beers at the party, and (ii) the statement that (s)he, the speaker, does not remember how many beers Homer drank at the party.

With regards to (38), the interpretive effect of having parallel statements may be way too subtle for most speakers to have sharp intuitions on, as opposed to crystal clear cases like (42). However, further observation reveals that this is not a function of the structure itself. Rather, it is an artifact of a given choice of lexical items, which may carry a certain pragmatic bias. For instance, the syntactic amalgam in (44) radically differs from its non-amalgamated version in (45) with respect to the scope of event structure.

(44) Homer drank everybody is asking me how many beers at the party.

(45) Everybody is asking me how many beers Homer drank at the party.

In (44), without the need to add one more level of embedding to the invaded clause – as we did in (42) – it is clear that the speaker is making the statement that Homer drank a certain number of beers at the party. That is, (s)he is committing to the truth that there actually happened an event of drinking beers (at the party) performed by Homer. In a parallel statement, the speaker is committing to the truth that everybody is asking him/her how many beers were drunk by Homer at the party, in that very event of drinking whose truth is being stated.

In contrast, the non-amalgamated – and arguably single-rooted – structure in (45) does not entail that the speaker is committing to the truth that an event of drinking beers at the party performed by Homer actually happen. Rather, the speaker is simply stating that everybody is asking him/her how many beers were drank by Homer at the party, in a given drinking event not presupposed to be true by him/her, the speaker. That is, it could be the case that the speaker uttering (45) believes that such event did not actually happened (in which case all the people asking him/her about Homer's personal drinking history are simply mistaken), or that (s)he simply ignores whether such event (s)he is being asked about really happened or not.

Also in chapter **V**, further evidence is provided for the hypothesis that both the invasive and the invaded clause have the status of matrix clauses, as they both exhibit properties not found in embedded clauses elsewhere. In addition, the apparent lack of superiority effects in syntactic amalgams like (46) is shown to be an epiphenomenon that follows from multiply-rooted representations, built through dynamic derivations, so that the relevant locality principle is indeed active, but gets obfuscated by the interaction of competing chains across the shared embedded clause and multiple parallel matrix clauses.

(46) a: I'll find out [how much money]$_1$ Bob gave t_1 to you can imagine [who]$_2$

b: I'll find out [who]$_2$ Bob gave you can imagine [how much money]$_1$ to t_2

That said, it is not obvious, under this approach to syntactic amalgamation, how those multiply-rooted phrase markers get mapped into a linear string of terminals at PF. Aside from the shared material, multiply-rooted phrase markers necessarily contain terminals that are dominated by only one root, and do not stand in any relation to the terminals dominated by only another of the roots. Therefore, whatever the linearization function is (e.g. Kayne's (1994) Linear Correspondence Axiom, the head parameter, etc.), it cannot establish precedence relations among all terminals in any deterministic way. Thus, in a nutshell, the multiple-root approach to syntactic amalgamation faces a linearization puzzle to the same extent that the Parallel-Intermingled-Trees approach sketched in (06) through (16) does.

As an elaboration on a general theoretical discussion from chapter **IV**, this issue of linearization of syntactic amalgams is addressed in chapter **V** from the viewpoint of the strong derivational approach. I argue that the attested word-order patterns obtain if the computational system is conceived as a derivational engine that builds phrase structure in a top-to-bottom fashion, along the lines of Phillips (1996, 2003), Drury (1998, 1999), and Richards (1999, 2002), *inter alia*.

In the Appendix to chapter **V**, I explore the consequences of this general model for the architecture of UG and its application to syntactic amalgamation (as proposed in chapters **IV** and **V**, respectively) to the PF interface. I argue that the combination of dynamic top-to-bottom derivations, multiple spell-out, and move-as-remerge makes it possible to theorematically derive some important generalizations about PF-structure, so that well-known mismatches between prosodic constituency and syntactic constituency are better understood as 'relativized isomorphism', where prosodic constituents reflect earlier stages of the syntactic derivation, pretty much like fossils of syntactic constituents that got reshaped in a later stage of the derivation – crucially after the relevant substructure being delivered to the phonological component – therefore not being reflected at LF.

Chapter **VI** concludes the dissertation, summarizing the project developed here, and pointing out issues to be addressed in future research.

I.2. Consequences for the Theory of Grammar (Architecture of UG)

Besides proposing an analysis for syntactic amalgams, this dissertation also has the more ambitious goal of contributing to broader discussions about the architecture of UG as a whole. Eventually, I end up advocating for a version of the Chomskyan Generative-Transformational framework (and, in particular, the

Minimalist Program) that considerably deviates from the mainstream versions in some significant technical aspects. Below I summarize the main theoretical claims that I make along the dissertation.

(i) The very operation of structure building (i.e. *merge*) is inherently defined as ‘tucking-in’ (Richards 1997), so that trees always exhibit endogenous growth (i.e. incoming material is never inserted at the current root node, but rather at some other node deep inside the tree). Consequently, there can be no *Extension Condition* on merge (contra Chomsky 1995: 190, 327-328; 2000: 136-137). This entails that syntactic constituency is heavily dynamic, as it is always the case that some of the sisterhood and motherhood relations among tree nodes get changed from one derivational step to the next one. Nevertheless, derivations can be considered to be fully monotonic from the point of view of the syntactic relations that grammatical principles are actually relevant to: asymmetric c-command and dominance.

(ii) Nothing in (any version of) the theory prevents a constituent from being immediately dominated by multiple other constituents distinct from one another, in a structure-sharing configuration, where multiple mother-nodes all have one daughter-node in common. This is actually a desirable consequence, as some syntactic constructions (i.e. amalgamation) require representations of this kind in order for the descriptive generalizations about it to be accounted for.

(iii) Chains are formed via multi-motherhood configurations. The kind of constituency displacement known as overt movement is achieved derivationally, with a constituent α being first merged in a position X, and then remerged in another position Y, so that X and Y stand in a c-command relation. That way, a so-called moved phrase is better understood as a pluripresent phrase, simultaneously occupying the head and the tail positions of a chain. Thus, (overt) move reduces to (re)merge[8] (cf. Abels 2001, Bobaljik 1995; Drury 1998, 1999; Epstein, Groat, Kawashima and Kitahara 1998; Guimarães 1999; 2002; 2003b/c; Gärtner 2002). Pushing this logic to the limit, I propose that the familiar c-command condition on chain-formation reduces to a c-command condition on the input to merge itself (which gets vacuously satisfied in the case of 'first merge').

(iv) Contrary to the tradition in Generative Grammar, I defend that there is no *Single Root Condition* on phrase markers.[9] Thus, in multiple-motherhood configurations, it is not necessary that there be some higher node

[8] In chapters IV and V, I argue that Covert Movement should be formalized as Agree (Chomsky 1998, 1999, 2000).

[9] This idea goes back, in some form, to Hoffman (1996) and, in a more direct way, to van Riemsdijk (2000).

dominating all of the mother-nodes that share a daughter (as in chains, cf. (iii)). In some instances, each of the multiple mothers of the shared daughter may have its own distinct dominance path above it, so that the whole phrase marker is shaped like two or more parallel trees connected to each other at some node somewhere in between the root and the leaves. From that perspective, most of what has been traditionally regarded as parataxis can be reduced to 'syntax pushed to the limit', where two or more parallel sentences get connected as what Riemsdijk (2001) called 'Siamese Trees', which allows them to be syntactically related to some extent, despite the parallelism.

(v) Following Chomsky (1995), I take inputs to syntactic derivations to be numerations, defined as sets of lexical tokens, which establish local domains where convergence and economy are evaluated (or 'derivational workspaces'). Since nothing in Set Theory prevents two or more numerations from intersecting and sharing some lexical tokens, this option is in principle available. Such intersections give rise to overlapping derivational workspaces, which allow local computations to interfere with one another to some extent, ultimately yielding Siamese Trees that exhibit paratactic effects.

(vi) With regards to the 'derivationalism *versus* representationalism' debate, I strongly endorse the view that the syntactic component of UG is a derivational system that builds structure step-by-step, so that the formal properties of phrases and sentences are taken to be effects of how syntactic structure is (economically) built, rather than effects of constraints on representations, or a combination of the two. Instead of assuming the standard view that structure is built in a bottom-up fashion, I take syntactic derivations to uniformly proceed in a left-to-right/top-to-bottom fashion, very much like in theories of parsing, as proposed by Phillips (1996, 2003), Drury (1998, 1999), and Richards (1999, 2002), *inter alia*. In fact, this top-to-bottom nature of derivations is an inevitable consequence of the 'generalized tucking-in' approach to *merge* outlined in (i) above, which enforces endogenous growth of trees across the board. Once the directionality of derivation is reversed, we start making predictions that, *ceteris paribus*, no representational approach can make (in particular, when it comes to issues of dynamic constituency, as outlined in (i) above). Moreover, these predictions seem to be by and large consistent with the facts, confirming Chomsky's (2000: 99) suspicion that the derivational approach is more than just an expository device.

(vii) Following Uriagereka (1998, 1999, 2002), I assume that there are no levels of representations in UG, only generative and interpretive components. From that perspective, the syntactic computation of a sentence proceeds in multiple successive 'cascades'. Different parts of the structure are generated and delivered to the phonological and semantic components separately from each other. These PF and LF chunks are incrementally put together and interpreted by the phonological and semantic components respectively. Unlike in Uriagereka's original formulation, I assume that there are no 'splitting points', where the targeted syntactic (sub)structures are delivered to both PF and LF interfaces simultaneously. Rather, syntax feeds phonology and semantics independently, not necessarily at the same derivational stages. As for the PF-interface, I endorse Uriagereka's (1999) position that those (multiple) applications of spell-out are driven by the necessity to satisfy the *Linear Correspondence Axiom* (Kayne 1994) at all derivational stages, in order to guarantee that the PF-representation being built incrementally fully satisfies the linearity requirement imposed by the A-P system. As a consequence of the assumptions outlined in (i) and (vi) above, the version of Uriagereka's (1999) Multiple Spell-Out model developed here – which stems from Drury (1998, 1999) – works in a top-to-bottom fashion, yielding interesting results when combined with the move-as-remerge approach outlined in (iii). Chain-links are originally generated in their highest positions and then remerged into lower positions. When remerge takes place, the affected element has already been spelled-out. What is being lowered/remerged is a combination of formal and semantic features whose corresponding morpho-phonological counterpart had already left the derivation for good.

(viii) As already mentioned in **I.1** above, well-known mismatches between prosodic constituency and syntactic constituency can be straightforwardly accounted for in the dynamic derivational system proposed here, in terms of 'relativized isomorphism'.

(ix) Following Phillips (1996, 2003), I assume that the parser and the grammar are not two distinct subsystems of the language faculty. Rather, they are one single engine, so that 'derivational time' equals 'real time'. This has major consequences not only for PF-linearization matters (as previously explored by Drury (1998, 1999) and Guimarães (1999a, 1999b), but also for information-theoretical aspects of syntactic amalgamation, which exhibits asymmetries with respect to the parallel matrix clauses, one having the status of the '*master* matrix clause', while the other(s) are '*subservient* matrix clause(s)'. In the system proposed here, this paratactic asymmetry is an effect of another asymmetry inherent to syntactic derivations, where there must be a linear order in the very process of building the multiple hierarchically parallel structures that constitute a syntactic amalgam.

II

Towards a Descriptively Adequate Theory of Syntactic Amalgamation[*]

The goal of linguistic theory — as established in the foundational works of Generative-Transformational Grammar — is to build models of natural language grammars in a way that *explanatory adequacy* is achieved (cf. Chomsky 1965: 24-37).[10] Needless to say, that presupposes that both observational adequacy and descriptive adequacy be achieved as well.

Therefore, while the following chapters are dedicated precisely to developing a theory of syntactic amalgamation that is as explanatory as possible, this chapter is dedicated to presenting my contribution to extending the body of descriptive generalizations about syntactic amalgamation, beyond the few initial class of facts reported by Lakoff (1974) and Tsubomoto & Whitman (2000).

[*] A preliminary version of the content of this chapter was presented at the *First Joint UConn/UMass/MIT/UMD Syntax Workshop*, held at the Linguistics Department of the University of Connecticut, on February 8th, 2003. I am thankful to the audience for the comments, especially to Klaus Abels, Norbert Hornstein, Howard Lasnik, and Andrew Nevins.

[10] Recent developments in the Minimalist Program have lead Chomsky (2001b) to envision the possibility of going 'beyond explanatory adequacy'. On this matter, see also Fukui (1996), Uriagereka (1995, 1996, 1998, 2000 [DELTA paper], 2002a, 2002b), Chomsky (1994 [MIND paper], 2001c), Martin & Uriagereka (2000), Freidin & Vergnaud (2001), and Epstein & Seely (2002).

I will postpone any substantial analytical and theoretical discussion to the following chapters, and simply focus on presenting 'raw facts'[11] and drawing new empirical generalizations concerning syntactic amalgamation as pre-theoretically as possible.

II.1. On the Ontology and Productivity of Syntactic Amalgamation

Let us begin with Lakoff's (1974) classical example of syntactic amalgam in (01).

(01) John invited you'll never guess how many people to his party.

Lakoff reports that examples of this kind were originally discovered by Avery Andrews. This particular construction was initially referred to as 'indirect question amalgam' by Lakoff (1974), and later called 'WH-amalgam' by Tsubomoto & Whitman (2000).

Another construction that Lakoff also took to be an instance of syntactic amalgamation is the one exemplified in (02).

[11] As far as the 'raw data' are concerned, I am extremely thankful to the following people for judgements and discussion: Juan Carlos Castillo, Stephen Crain, John Drury, Scott Fults, Norbert Hornstein, Howard Laskink, Paula Kempchinsky, Anthony Kroch, Ruth Lopes, Elixabete Murguia, Andrew Nevins, Leticia Pablos, Colin Phillips, Paul Pietroski, Beth Rabbin, Phillip Resnik, Cilene Rodrigues, Francisco Simões, Rosalind Thornton, Juan Uriagereka, and Jacek Witkos.

(02) John is going to I think it's Chicago on Saturday.

Lakoff referred to examples like (02) as 'embedded cleft sentences' (which Tsubomoto & Whitman (2000) later called 'cleft-amalgam'), and credited Larry Horn for the discovery.

These two constructions are indeed very similar. They both exhibit a sentence in the first plain conveying the main message, which gets interrupted and 'invaded,' so to speak, by another sentence introducing a secondary message in a *quasi*-parenthetical fashion, such that the 'invasive clause' contains a focalized phrase somewhere in its embedded CP-domain.

In this dissertation, I will focus on *WH-amalgamation*. Most of what I have to say — both descriptively and analytically — will naturally carry over to cleft-amalgams. There will be, however, a few contrasts and incomensurabilities, which I will point out along the way. At any rate, although the ultimate goal is to build a general theory of amalgamation, the scope of this dissertation is to be understood primarily as being restricted to WH-amalgamation.[12]

At first blush, there seems to be something unusual or out of ordinary about examples like (01) and (02). They feel somehow 'marked'. *Prima facie*, there

[12] In fact, Lakoff (1974) takes into consideration six different constructions, and claims that they are all instances of syntactic amalgam: (i) Andrews' indirect questions, (ii) Horn's embedded cleft sentences, (iii) Forman's parentheticals, (iv) Davison's performative predicate modifiers, (v) Liberman's because-cases, (vi) Liberman's or-cases, and (vii) tag questions. The sentences in (01) and (02) are examples of (i) and (ii), respectively. In my view, although Lakoff's (1974) analyses of all these cases share important similarities, it is not accurate to say that he gave a unified account of all six cases. Crucially, in his analysis, each construction has its own rule, which is similar in format to the other rules, but explicitly refers to syntactic and pragmatic properties specific to that given construction. An introductory presentation of all amalgamation rules proposed by Lakoff is found in the *Appendix* to chapter **III**.

are at least four ways of approaching this 'marked' character of syntactic amalgamation.

At first sight, amalgamation may seem to belong somehow in the periphery, or even outside of the grammar, as some sort of 'paralinguistic' discourse strategy. There seems to be something 'idiomatic' about amalgamation, as if those 'invasive *quasi*-parenthetical chunks' were all bits of previously 'lexicalized' sentential material, which then behave as determiners or modifiers to NPs. In fact, there is a relative small class of 'invasive chunks' (i.e. *God only knows*, *you can imagine*, *you'll never guess*, etc) that appear — in some variant or other — over and over in the typical examples of amalgamation, as shown in (03).

(03) a: John invited 300 people to *you can imagine what kind of party*.

b: John has been writing his autobiography for *God only knows how many years*.

c: Ever since he ran away, John has been hiding *nobody has a clue where*.

d: John gave all his money to *I wonder who*.

e: John was nominated for *I forgot which music award*.

f: John was kissed in public by *we all still remember which celebrity*.

Although those recurrent substrings might indeed be instances of 'readymade bits of discourse', and despite the fact that their meanings are somewhat 'pragmatically equivalent', there is also enough evidence that such constructions are way less formulaic and way more productive than they seem to be at first blush, as the examples in (04) reveal.

(04) a: John made a big deal out of having met *I couldn't care less which celebrity* at the party.

b: Noam Chomsky wrote *maybe Zellig Harris kept track of how many drafts of 'Transformational Analysis'* until the final version of the dissertation was presented to the committee.

c: The Beatles had *not even George Martin remembers how many recording sessions* at the Abbey Road studios.

d: Batman can be reached at *nobody in Gotham City but commissioner Gordon knows which phone number*.

e: Achilles held Hector's corpse for *Priamus certainly never forgot how long* during the Trojan War.

f: Developing a new antibiotic takes many years of research and *you can figure* how much money.

A second way of approaching syntactic amalgams is to treat them as structures that are anomalous at some relevant level of representation of the grammar, but which are somehow able to 'fool the parser', similarly to what happens with examples like (05).[13]

(05) * More people have been to Russia than I have.

There is, however, a clear difference between the 'markedness' of (01) and (02) on the one hand, and the 'oddity' of (05) on the other hand. Upon closer scrutiny, the latter exhibits an anomalous representation, despite the fact that it may 'sound good' at first blush. Speakers who initially judge it as acceptable inevitably recognize its oddity and change their minds immediately after being asked what the meaning is supposed to be. As a matter of fact, examples like (05) can be found only in works by linguists, and do not show up in spontaneous speech. The former examples, on the other hand, do have crystal clear meanings that can be paraphrased (cf. (09) below). Syntactic amalgams are uniformly way more acceptable than structures like (05), typically triggering quite robust judgments. Also, examples like (01) and (02) can be found in spontaneous speech, and are quite productive, as shown in (03) and (04).

Another possibility is that the 'markedness' of (01) and (02) is just the opposite of 'grammatical anomaly that manages to fool the parser'. Instead, such data would be fully well-formed, but would still sound a bit 'unnatural' due to the high complexity of their structures, therefore 'giving the parser a hard time',

[13] cf. Fults & Phillips (2004).

not to the point of leading to unacceptability, but to the point of making the structure have the status of 'marked'. From that perspective, syntactic amalgams would be comparable to structures like (06) below.

(06) The rat that the cat that the dog bit chased ran away.

Sentences like (06) are standard examples of structures which look like 'word-salad' at first blush. However, upon closer inspection, it turns out that they do not exhibit any formal property that could remotely be pointed out as a potential source of ungrammaticality. Moreover, those same examples are eventually judged as acceptable if one gives the speaker paper-and-pen, and enough time to decompose it, going through the reasoning summarized in (07)

(07) a: The rat [that the cat [that the dog bit] chased] ran away.
b: The rat [that the cat chased] ran away.
c: The rat ran away.

Notice, however, that examples of the sort of (01) and (02) are not nearly as confusing as (06). Despite sounding 'marked', syntactic amalgams are quite easily understandable and acceptable right from the outset, as opposed to structures with multiple center-embedding like (06). Ironically, although it doesn't take paper-and-pen and extra-time for any speaker to figure out what a syntactic amalgam means, any attempt to decompose its structure in its parts is a huge challenge to any syntactician, not to mention a naïve speaker.

Finally, one may take this marked status of syntactic amalgams as a consequence of some grammatical process(es) of a more familiar kind. Whatever amalgamation ultimately is, it would trigger 'markedness' no more and no less than whatever grammatical processes are responsible for the generation of sentences with topic-comment or focus-pressuposition structures, for instance, which are arguably also 'marked', to the extent that their use is way more restricted by contextual variables.

From that perspective, it seems reasonable, as a starting point, to hypothesize that the syntactic amalgams in (01) and (02) — repeated below as (08a) and (08b) —are 'convoluted' versions of (09a) and (09b) respectively, generated through some complex combination of familiar context-sensitive operations (e.g. movement, ellipsis, binding, control, etc).

(08) a: John invited you'll never guess how many people to his party.

b: John is going to I think it's Chicago on Saturday.

(09) a: You'll never guess **[how many people]$_1$** John invited **t$_1$** to his party.

b: I think it's **[Chicago]$_1$** (that) John is going to **t$_1$** on Saturday.

If this reasoning is on the right track, then the 'marked feel' of amalgams like (08a) and (08b) would have a status similar to the one of the parasitic gap constructions in (10), whose acceptability and grammaticality are nowadays a consensus, but once were controversial (cf. Engdahl 1981; Chomsky 1982: 36-78; Culicover & Postal 2001).[14]

(10) a: **[Which articles]**$_1$ did John file $\mathbf{t_1}$ without reading $\boldsymbol{e_1}$ first ?

b: Here is **[the influential professor]**$_1$ that John sent his book to $\mathbf{t_1}$ in order to impress $\boldsymbol{e_1}$.

c: **[Who]**$_1$ did you give a picture of $\mathbf{t_1}$ to $\boldsymbol{e_1}$?

d: **[Who]**$_1$ did John's talking to $\boldsymbol{e_1}$ bother $\mathbf{t_1}$ most?

e: It was Ernest **[who]**$_1$ pictures of $\boldsymbol{e_1}$ tended to depress $\mathbf{t_1}$.

Therefore, despite initial impressions, there is enough reason to pursue the hypothesis that syntactic amalgams are actually built through rather ordinary

[14] Making explicit reference to the sentence here reproduced in (08c), Chomsky (1982: 37) said: *"In Chomsky (1981), I assumed that (08c)* [my numbering, MG] *was ungrammatical, but that was not really correct; rather, it is more or less acceptable under the interpretation given, while other examples of a similar kind are quite acceptable, as we shall see directly"*. Ahead, Chomsky (1982: 39) asked the following concrete questions about parasitic gaps:

a: Why does the phenomenon [of parasitic gaps] exist at all?
b: What are the basic properties of parasitic gaps?
c: What principles and mechanisms determine these properties?

I see no reason to consider that the 'markedness' of syntactic amalgamation is *a priori* any different from the 'markedness' of parasitic gaps. The natural questions to ask, at this point, are:

a': Why does the phenomenon of syntactic amalgamation exist at all?
b': What are the basic properties of syntactic amalgamation?
c': What principles and mechanisms determine these properties?

This chapter addresses question (b'); while questions (a') and (c') are dealt with in the following chapters.

syntactic tools, as just plain syntax pushed to the limit. This is the position that I will ultimately defend throughout this dissertation.

The issue, then, is to identify what exactly those 'ordinary tools' of 'syntax in-the-limit' are. The first steps toward achieving that goal must necessarily involve describing the facts as accurately and exhaustively as possible.

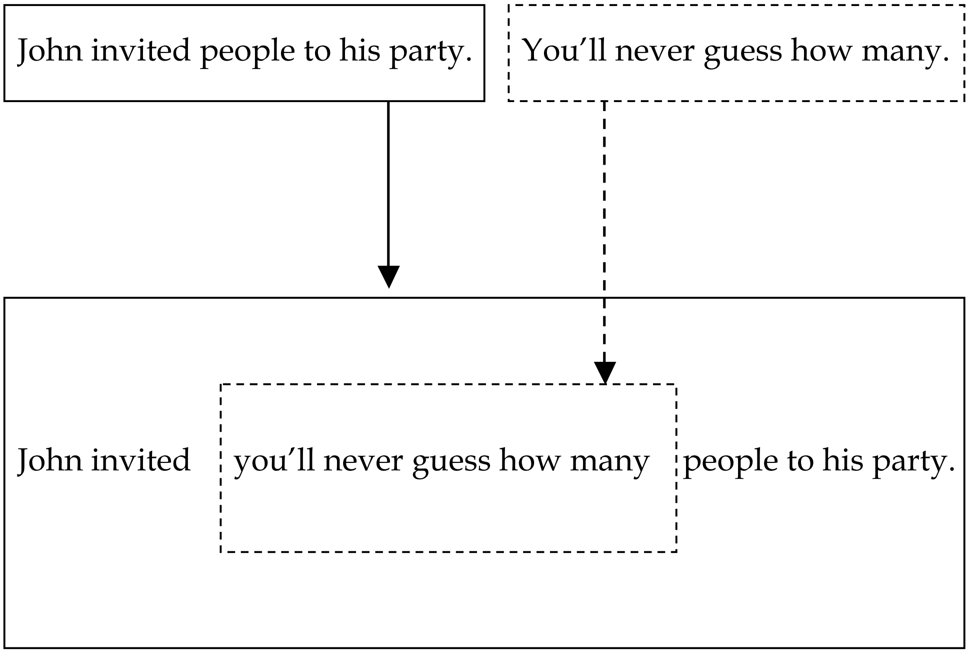
As a first approximation, one may adopt the view that the amalgams in (08) are actually 'convoluted versions' of (09), as sketched above. This seems to be intuitively on the right track, but there is something else about the examples in (08) that indicates that syntactic amalgamation has a paratactic nature, rather than a fully hypotactic nature. That would call for a trans-derivational approach. For instance, although (08a) is semantically equivalent to (09a) as far as their propositional contents are concerned, their informational structures differ in such a way that the mini-text in (11) is a more accurate paraphrase of (08a).

(11) John invited people to his party. You'll never guess how many.

This is so because the 'main message' in both (08a) and (11) is about the event of John inviting people to his party, whereas, in (09a), the 'main message' is about the guessing event.

Thus, descriptively speaking, it seems as if **<u>you'll never guess how many</u>** somehow 'invades' **<u>John invited people to his party</u>** in a *quasi*-parenthetical fashion, as suggested in chapter **I**, and repeated below in (12).

(12)


John invited people to his party.
You'll never guess how many.
John invited
you'll never guess how many
people to his party.

This is actually the intuition behind the classical analysis proposed by Lakoff (1974), who defined syntactic amalgam as follows.

> By 'syntactic amalgam' I mean a sentence which has within it chunks of lexical material that do not correspond to anything in the logical structure of the sentence; rather they must be copied in from other derivations under specifiable semantic and pragmatic conditions.
>
> Lakoff (1974: 321)

Although, in this passage, Lakoff was being somewhat vague about what exactly such 'clause invasion' would be, he actually proposed a specific technical formalism where relatively precise definitions of *"not corresponding to anything in the logical structure of the sentence"* and *"copied in from other derivations"* were spelled out. Those details will be presented and criticized in chapter **III**, and a concrete alternative proposal will be provided in chapter **V**. For now, let us focus on the structural contexts under which amalgams are licensed. Notice that Lakoff talks about *"specifiable semantic and pragmatic conditions"* which would license the relevant trans-derivational operations. Crucially, he did not talk about any 'syntactic condition' in his formalization of the amalgamation rules (cf. chapter **III**).

In what follows, I show that, aside from semantic and pragmatic conditions, amalgamation is sensitive to the syntactic properties of both the 'invaded' and the 'invasive' sentences.

II.2. On the 'Appropriate Modification' Requirement

Consider (13a) and its amalgamated version in (13b).

(13) a: Everybody knows what Sarah gave to Joe.

b: Sarah gave *everybody knows what* to Joe.

Both examples are acceptable to the same extent. This contrasts with the slightly different pair of examples in (14).

(14) a: Tom knows what Sarah gave to Joe.

b: ? Sarah gave *Tom knows wha*t to Joe.

If uttered in an out-of-the-blue situation, (14b) tends not to be as acceptable as its non-amalgamated counterpart in (14a), or as the similar amalgam in (13b).

Interestingly, acceptability is significantly improved if the subject of the relevant verb[15] in the invasive clause is appropriately modified, as shown in (15).

(15) a: ? Sarah gave *Tom knows what* to Joe.

b: Sarah gave *only Tom knows what* to Joe.

c: Sarah gave *perhaps Tom knows what* to Joe.

d: Sarah gave *not even Tom knows what* to Joe.

e: Sarah gave *certainly Tom knows what* to Joe.

The same 'acceptability-boost' effect obtains if the 'invasive clause' is made more complex in the appropriate way, as in (16).

(16) Sarah gave *I bet Tom knows what* to Joe.

Likewise, the same relatively unacceptable string of words in (14b) becomes fully acceptable if the subject of the relevant verb in the invasive clause has the status of a contrastively focalized phrase, marked with the appropriate prosody, as indicated in (17).

[15] The 'relevant verb' is the one which, in the non-amalgam counterpart, selects the material corresponding to the invaded clause as an indirect-question.

(17) Sarah gave **TOM** knows what to Joe.

Finally, no explicit ‘appropriate modification’ to the structure is necessary if the same syntactic amalgam is uttered in a favorable context. For instance, suppose that the following background information is shared by both the utterer and the addressee.

(18) *Context*

Joe is a prisoner at a maximum-security penitentiary. Once every week, his daughter Sarah visits him, sometimes bringing him candies, magazines, and other gifts. There is one soldier named Tom, whose job is to watch out every meeting between Joe and any visitor in its entirety, inspecting and keeping track of every object exchanged, making sure that no weapon, cell phone or any other unauthorized object gets in Joe’s hands. Once every week, Tom is supposed to report to the security supervisor whatever happened in Joe’s meetings with any visitors. By default, nobody else performs Tom’s task.

In the scenario described above, the syntactic amalgam in (14b) – repeated below as (19) – is indeed fully acceptable, entirely dispensing with any additional ‘appropriate modifications’.[16]

[16] In certain cases, like (i) below, the ‘appropriate context’ that licenses not-so-acceptable syntactic amalgam is so salient and predictable, that no further explanation (like (18)) is necessary.

(19) Sarah gave *Tom knows wha*t to Joe.

The common feature to all those strategies is that the 'invasive clause' introduces a parallel secondary message that contrasts with the main message introduced by the 'invaded clause' in terms of the knowledge (or memory) of a given propositional content (i.e. the propositional content of the core structure being 'invaded'), or lack thereof, on the part of all people in the relevant universe of discourse. What all those 'appropriate modifications' do is precisely to introduce the contrast of knowledge just mentioned.

Mutatis mutandis, the acceptability-tied-to-appropriate-modification pattern just described resembles very much what is observed in middle constructions, as shown below (cf. Roberts 1986).

(20) *Active Structures*

a: I biased the vacuum-tubes of Max's amplifier.

b: I biased the vacuum-tubes of Max's amplifier quite quickly.

(21) *Middle Structures*

a: * The vacuum-tubes of Max's amplifier biased.

b: The vacuum-tubes of Max's amplifier biased quite quickly.

(i) During the recording sessions of *St. Pepper's Lonely Hearts Club Band*, Ringo Starr recorded the vocal parts of *With a Little Help from My Friends* Paul McCartney couldn't believe how many times until the drummer was finally satisfied with his own performance as a front singer.

(22) *Middle Structures in a Cleft Environment*

a: It's Max's amplifier whose vacuum-tubes were able to get biased.

b: It's The vacuum-tubes of Max's amplifier which were able to get biased.

(23) *Middle Structures in a Contrastive-Focus Environment*

a: The vacuum-tubes of MAX'S AMPLIFIER biased.

b: THE VACUUM-TUBES OF MAX'S AMPLIFIER biased.

(23) *Plain Middle Structure in an Appropriate Context*

I couldn't get most of the amplifiers to work properly. Somehow, only the vacuum-tubes of Max's amplifier biased.

II.3. On the Unboundness of Amalgamation

Consider the sentence in (24a) and its corresponding amalgamated version in (24b).

(24) a: Only his wife knows exactly [how much money John has donated to charity ever since he became rich].

b: John has donated *only his wife knows exactly how much money* to charity ever since he became rich.

As shown below, the parenthetical-like invasive chunk may be complex, exhibiting embedded sentences in it. The data in (25) and (26) indicate that, no matter how deeply embedded a subordinate (indirect question or cleft) sentence is, it is possible to build a corresponding syntactic amalgam where that embedded sentence figures as the 'invaded clause', with the 'invasive clause' being a complex structure of recursively embedded sentences. The phenomenon seems to be unbounded at the competence level, limited only by parsing limitations.[17]

(25) a: Sarah once told me [that only his wife knows exactly [how much money John has donated to charity ever since he became rich]].

b: John has donated *Sarah once told me that only his wife knows exactly how much money* to charity ever since he became rich.

[17] As shown below, the same pattern obtains with regards to cleft-amalgams:

(i) a: I think it was sixty-five thousand Euros that John has donated to charity ever since he became rich.

b: John has donated *I think it was sixty-five thousand Euros* to charity ever since he became rich.

(ii) a: Sarah once told me that she believes it was sixty-five thousand Euros that John has donated to charity ever since he became rich.

b: John has donated *Sarah once told me that she believes it was sixty-five thousand Euros* to charity ever since he became rich.

(iii) a: I remember that Sarah once told me that she believes it was sixty-five thousand Euros that John has donated to charity ever since he became rich.

b: John has donated *I remember that Sarah once told me that she believes it was sixty-five thousand Euros* to charity ever since he became rich.

(26) a: I remember [that Sarah once told me [that only his wife knows exactly [how much money John has donated to charity ever since he became rich]]].

b: John has donated *I remember that Sarah once told me that only his wife knows exactly how much money* to charity ever since he became rich.

This lack of upperbound on the complexity of the 'invasive clause' further suggests that those quasi-parenthetical chunks are not formulaic idioms (cf. **§I.1** above). However, that does not yet argue for the unboundness of amalgamation itself. As a matter of fact, it is also the case that there can be multiple amalgamation. That is, a sentence may be 'invaded' at many points by many 'invaded clauses'. For instance, consider the sentence in (27).

(27) John invited *you will never guess how many people* to *you can imagine what kind of a party*.

As it will be shown in **§II.4** below, whenever there is multiple amalgamation, the many 'invasive clauses' are hypotactically unrelated (although paratactic related). In fact, the only possible way to build a non-amalgamated version of (27) is through a paratactic arrangement of independent sentences, as in (28).

(28) John invited people to some event. You'll never guess how many... and you can imagine what kind of party.

Crucially, the multiple amalgam in (27) has no hypotactically structured correlate, unlike what happens to simple amalgams, as shown in (24), (25) and (26) above.

In fact, this is evidence against taking amagamation to be the result of a combination of transformations on a single-rooted phrase marker of the sort of (24a), (25a) and (26a). This was actually the main point that Lakoff (1974) was making when he first brought up the example in (27) above.

Lakoff (1974) also pointed out that, setting aside parsing limitations, the iterative nature of a syntactic amalgam is unbounded, as shown in (29).

(29) John invited *you will never guess how many people* to *you can imagine what kind of a party* at *it should be obvious where* with *God only knows what purpose in mind*, despite *you can guess what pressures*.

Two minor observations to be added to Lakoff's original findings are illustrated below.

First, not only can there be many 'invasions' to a sentence, but the 'invasion points' need not to be all associated to positions of arguments and adjuncts of the same predicate. The 'invasive clauses' may be distributed across different predicated along the sentence, even across predicates that don't scope over one another, as shown in (30).

(30) The fact that Tom invited *he couldn't even count how many people* to his graduation party was the reason why his father ended up spending *God knows how much money* on beers, snacks, napkins, and so on.

Moreover, multiple amalgamation may be an arbitrary combination of WH-amalgamation and Cleft-amalgamation, as shown in (31).

(31) John invited *I think it was three-hundred people* to *you can imagine what kind of a party*.

II.4. Multiple Parallel Messages Presented in Two Layers of Information

Consider again the amalgam in (01), repeated below as (32).

(32) John invited you'll never guess how many people to his party.

As a first approximation, the meaning of this syntactic amalgam can be informally described through the paraphrase in (33).

(33) You'll never guess how many people John invited to his party.

Whether or not the syntactic structures of (32) and (33) are transformationally related, these two examples seem to be, by and large, semantically equivalent, sharing the same propositional content.

There is, however, one asymmetry in the meaning of (32) which is not captured by the paraphrase in (33). Despite the propositional contents of (32) and (33) being the same, these two examples differ in informational structure. Speakers report that they feel the invitation event being somehow more 'discoursively salient' in (32) than it is in (33). Descriptively speaking, the contrast under discussion is the fact that, in (32), **John invited people to his party** has a 'matrix-clause feel' to it, with **you'll never guess how many** acting as a parenthetical; whereas, in (33), it is **[how many people]$_1$ John invited t$_1$ to his party** that is perceived as a subordinate clause embedded under a VP headed by **guess**, around which the matrix clause is built.

Therefore, a more accurate paraphrase of (32) would be the one in (34).

(34) John invited people to his party. You'll never guess how many.

The mini-text in (34) is a much better paraphrase than (33) for expressing the informational content of the amalgam in (32), to the extent that it shows, in a transparent way, two parallel messages. One is structured around an inviting event, and the other one around a guessing event, such that the first is somehow at the front 'informational layer' — as the main message — and the second is in another informational layer behind it, as a secondary chunk of information.

This generalization is further supported by multiple amalgams. Consider the example in (27), repeated below as (35).

(35) John invited you'll never guess how many people to you can imagine what kind of party.

Any utterance of the construction in (35) will be mainly about John's invitation of people to a certain party. As secondary thoughts, there are two parallel messages being conveyed: one concerns a guessing event, and the other one concerns an imagining event (state). A reasonably faithful paraphrase is the one in (36).

(36) John invited people to some event. You'll never guess how many... and you can imagine what kind of party.

In (35) — and, to some extent, in (36) —, there is no obvious hierarchical organization between the two secondary messages. They both seem to be equally 'behind' the main message; and fully parallel to each other, so that none of them is more 'discoursively salient' than the other. [18]

As opposed to simple amalgams like (32), multiple amalgams like (35) are not paraphrasable in a hypotactic fashion, along the lines of (33), even if imprecisely. Any attempt to do so fails, as shown in (37).

(37) a: * You'll never guess [how many people]$_1$ you can imagine [what kind of party]$_2$ John invited t_1 to t_2.

b: * You can imagine [what kind of party]$_2$ you'll never guess [how many people]$_1$ John invited t_1 to t_2.

Not only are these two hypotactic structures both unacceptable to begin with (being arguably agrammatical due to violations of the relevant locality constraints on (WH) movement);[19] but their LF structures exhibit properties that couldn't even remotely correspond to that.

Abstracting away from syntactic locality matters, standard assumptions about semantic compositionality would predict that, in (37a), the clause corresponding to the invitation event is the complement of the verb **imagine**; and

[18] This bi-layered informational structure seems to be constant across amalgams, no matter how many invasions there are. When faced with multiple amalgams like (i), speakers report to have a 'gut feeling' that there are only two informational layers: one displaying the inviting event; and another one displaying all other events grouped in a flat, parallel fashion in terms of discourse salience.

(i) John invited *you will never guess how many people* to *you can imagine what kind of a party* at *it should be obvious where* with *God only knows what purpose in mind*, despite *you can guess what pressures*.

[19] cf. **§II.6** for a more detailed description, and **§V.5** for discussion and analysis.

the clause corresponding to the imagining event is the complement of the verb **guess**. The resulting meaning is such that what is being guessed is something about an event of imagining that concerns an invitation event. But this is not what the multiply amalgamated structure in (35) means. Instead, the meaning of (27) is such that what is being guessed is something about the invitation event itself, which is also what is being imagined. The events of imagining and guessing are independent. The same logic applies to (37b), which exhibits the opposite scope between **guess** and **imagine**. Given standard assumptions about semantic compositionality, the clause corresponding to the imagining event is the complement of the verb **guess**; and the clause corresponding to the guessing event is the complement of the verb **imagine**. The resulting meaning is such that the event of imagining that concerns a guessing event, which — in its turn — is a guessing of some property of the invitation event. But this is not what the multiply amalgamated structure in (27) means either.

In a nutshell, an accurate paraphrase of (27) must necessarily exhibit a structure neither **guess** scopes over **imagine**, nor **imagine** scopes over **guess**. This is precisely the case with the paratactic arrangement in (36).

The data in (38) show further evidence that syntactic amalgams exhibit informational structures quite distinct from the ones of their corresponding hypotactic paraphrases.

(36) a: John invited his fiancé keeps asking me how many women to his bachelor party.

b: His fiancé keeps asking me how many women John invited to his bachelor party.

c: John invited women to his bachelor party. His fiancé keeps asking me how many.

In (36a), the speaker is making the statement that John invited a certain number of women to his bachelor party. Thus, the speaker is committing to the truth that there actually happened an event of inviting people (to one's own bachelor party) whose agent was John. In a parallel statement, the speaker is committing to the truth that John's fiancé keeps asking him/her (i.e. the speaker) how many women were invited by John to such a party, and that such an invitation corresponds to that same inviting event whose truth is being stated.

On the other hand, the hypotactic paraphrase in (36b) does not entail that the speaker is committing to the truth that such event of inviting people to a bachelor party performed by John actually happened. Rather, the speaker is merely stating that John's fiancé keeps asking him/her how many women were invited by John to a bachelor party, in a given inviting event that the event does not presupposes to be true. It could well be the case that the speaker uttering (36b) strongly believes that such invitation never happened, and/or that such a party was never planned, and will never take place (in which case John's fiancé is simply mistaken about the whole story). It could also be that the speaker simply ignores whether or not such invitation event (s)he is being asked about really happened. Crucially, the paratactic paraphrase in (36c) faithfully captures the relevant pressuposition present in the informational structure of the amalgam (36a).

Finally, the example in (37) shows further evidence that the meaning of a syntactic amalgam is structured in two layers.

(37) Tom said that John invited I forgot how many people to his party.

In (37), the event of John inviting a certain number of people to his party is simultaneously the theme of the saying event performed by Tom and the theme of the forgetting event experienced by the speaker. The key property of (37) is that the forgetting and the saying events are completely independent from one another.

In a context where (37) is true, Tom did not say anything about the speaker's lack of memory regarding how many people John invited to his party. All Tom said is that John invited a certain number of people (for instance, fifty-seven) to his party. Conversely, it is not the case that the speaker forgot Tom having said how many people John invited to his party. All the speaker forgot is the cardinality of the number x such that John invited x people to his party (as opposed to the cardinality of the number ***y*** such that Tom said that John invited ***y*** people to his party).

Therefore, the informational structure of (37) must contain two layers. Presented 'at the front', there is a message about Tom having said that John invited a given number of people to his party. As a secondary message behind that one, there is the event/state of forgetting, experienced by the speaker, so that what is being forgotten is the exact number of people that John invited to his party.

The paratactic structure in (38) is a paraphrase of (36) which — as opposed to (39) — reflects the informational structure of (37).

(38) Tom said that John invited (a bunch of) people to his party. I forgot how many

(39) I forgot **[how many people]$_1$** Tom said that John invited **t$_1$** to his party.

II.5. Insensitivity to Islands

As pointed out by Tsubomoto & Whitman (2000: 80), invasive clause(s) may appear inside certain domains that are well-known islands for extraction (cf. Ross 1967, 1986), and which actually block movement of the relevant phrase in the corresponding hypotactic non-amalgamated versions of the same examples, as shown in (40) through (44) below.

(40) *Coordinate-Structure Island*

a: John invited one-hundred men and *you can imagine how many women* to his party.

b: John invited one-hundred men and two-hundred women to his party.

c: * You can imagine **[how many women]$_1$** John invited **{** [one-hundred men] and **t$_1$ }** to his party.

(41) *Relative Clause Island*[20]

a: John invited a woman he met *you'll never guess where* to his party.

b: John invited a woman he met at the church to his party.

c: * You'll never guess **[where]$_1$** John invited [a woman$_2$ **{** he met *e*$_2$ **t }**] to his party.

[20] The unacceptability/ungrammaticality of the example in (41c) is tied to the reading in which the WH phrase **where** is interpreted as the place in which the male person denoted by **he** (most likely John) met the female person denotated by **woman**. It is irrelevant for this discussion that the same string of words is acceptable under the reading in which **where** is interpreted as the place in which John when he invited a woman he met (somewhere) to the party. By standard the syntactic and semantic assumptions, only the first reading is associated with a syntactic structure involving extraction (of a WH) out of an island.

(42) *Adjunct Clause Island*

a: John invited all his friends to a big party immediately after I hired *it's obvious who* for the job.

b: John invited all his friends to a big party immediately after I hired his daughter for the job.

c: * It's obvious **[who]$_1$** [John invited all his friends to a big party **{** immediately after I hired **t$_1$** for the job **}**].

(43) *Subject Island*

a: Chatting with *you can imagine who* on the phone makes Max happy.

b: Chatting with his brother on the phone makes Max happy.

c: * You can imagine **[who]$_1$** **{** chatting with **t$_1$** on the phone **}** makes Max happy?

(44) *Complex NP/DP Island*

a: Susan dismissed the claim that her husband dated *I can't remember who* before they got married.

b: Susan dismissed the claim that her husband dated Sarah before they got married.

c: * I can't remember **[who]$_1$** Susan dismissed **{** the claim that her husband dated **t$_1$** before they got married **}**.

The same pattern obtains in cleft-amalgams, as shown below.

(45) *Coordinate-Structure Island*

a: John invited one-hundred men and *I think it was two-hundred women* to his party.

b: John invited one-hundred men and two-hundred women to his party.

c: * I think it was **[two-hundred women]$_1$** that John invited **{** [one-hundred men] and **t$_1$** **}** to his party.

(46) *Relative Clause Island*[21]

a: John invited a woman he met at *I think it was the church* to his party.

b: John invited a woman he met at the church to his party.

c: * I think it was **[at the church]$_1$** that John invited [a woman$_2$ **{** he met *e*$_2$ **t$_1$** **}**] to his party.

(47) *Adjunct Clause Island*

a: John invited all his friends to a big party immediately after Mr. Goldstein hired *I'm pretty sure it was his daughter* for the job.

b: John invited all his friends to a big party immediately after Mr. Goldstein hired his daughter for the job.

[21] cf. previous footnote.

c: * I'm pretty sure it was **[his daughter]**$_1$ that [John invited all his friends to a big party **{** immediately after Mr. Goldstein hired **t**$_1$ for the job **}**].

(48) *Subject Island*

a: Chatting with *I guess it's his brother* on the phone makes Max happy.

b: Chatting with his brother on the phone makes Max happy.

c: * I guess it's **[his brother]**$_1$ that **{** chatting with **t**$_1$ on the phone **}** makes Max happy.

(49) *Complex NP/DP Island*

a: Susan dismissed the claim that her husband dated *I guess it was Sarah* before they got married.

b: Susan dismissed the claim that her husband dated Sarah before they got married.

c: * I guess it was **[Sarah]**$_1$ that Susan dismissed **{** the claim that her husband dated **t**$_1$ before they got married **}**.

II.6. Apparent Lack of Superiority Effects

Ever since Chomsky (1973), contrasts like the one in (50) have been treated as effects of a *Superiority Condition* on transformations (or whatever deeper principle such condition ultimately reduces to), which is a locality requirement on displacement.[22]

(50) a: I'll find out [how much money]$_1$ Bob gave t$_1$ to [whom]$_2$
b: * I'll find out [who]$_2$ Bob gave [how much money]$_1$ to t$_2$

I will postpone any technical discussion on the nature of superiority to chapter **V**. For now, from a general descriptive perspective, it suffices to say that whatever the deeper principle that ultimately accounts for superiority is, it has the effect of preventing a phrase α of a given type (in this case, a WH) from moving to the next available target position up (in this case, the embedded spec/CP) if there is another β of the same type as α that is closer to the target than α is (where *closeness* is yet to be precisely defined).

Before any movement, **how much money** is closer to the target than **who** is. Thus, by superiority, the movement of **who** across **how much money** is

[22] Standard examples of Superiority typically involve competition between a subject and an object (as in (i) below) rather than two objects, which may raise further issues regarding equidistance.
(i) a: who$_1$ t$_1$ bought what?
b: * what$_1$ did who buy t$_1$?
All my examples involve double objects because here I am focusing on how superiority works in syntactic amalgams; and syntactic amalgamation cannot affect *bona fide* subjects to begin with, as shown below (cf. Guimarães 2003a/b/c):
(i) * I wonder what$_1$ you can imagine who bought t$_1$.

forbidden. On the other hand, provided that movement of some WH phrase to spec/CP is required, **how much money**, being the closest WH to the target, should move, as it does in (50a).

Such contrast does not exist in (51), however.

(51) a. I'll find out [how much money]$_1$ Bob gave t_1 to [someone]$_2$

b. I'll find out [who]$_2$ Bob gave [some money]$_1$ to t_2

Given the way superiority is defined, this is not surprising. The grammaticality of (51a) is predicted straightforwardly. As for (51a), no violation of superiority exists even though, at the derivational stage before movement, **who** is as far alway from the target as it is in (50b). It is irrelevant that **some money** is closer to the target than **who** is, since only phrases of the relevant type (in this case, WH) can count as interveners, and block the movement of distant phrases.[23]

The superiority effect illustrated in (50) is not observed in cases where one of the WHs is part of a syntactic amalgam (cf. Guimarães 2002; 2003a/b/c/d).[24]

[23] Strictly speaking, the movement of **how much money** in both (50a) and (51a) indeed crosses an intervening phrase, namely: the subject **Bob**. This is being ignored for expository reasons, and it does not affect the reasoning, since **Bob**, not being a WH-phrase, cannot block the movement of a WH phrase, just like the direct object **some money** in (51b).

[24] Some English speakers judge (52b) as somewhat degraded in comparison to (52a). As suggested by Howard Lasnik (personal communication), this may be due to parsing difficulties associated with a highly complex material intervening between the WH and the stranded preposition that selects it; as independently attested in cases like (i) which most speakers report to be significantly less acceptable than (ii).

(i) ?* **Who**$_1$ did you give that Beatles record autographed by George Harrison that you got in London last year to t_1 ?

(ii) **Who**$_1$ did you give that Beatles record to t_1 ?

Crucially, even for those speakers, (52b) is much more acceptable than (50b), which is just plain impossible. Interestingly, such degrading effect does not exist at all in Romance (exemplified

(52) a: I'll find out [how much money]$_1$ Bob gave t_1 to you can imagine who

b: I'll find out [who]$_2$ Bob gave you can imagine how much money to t_2

Notice that (52) patterns like (51) rather than like (50), even though both sentences in (52) have two WH-phrases apparently competing with each other for occupying the spec/CP position right under the VP headed by **find out**, just as in (50).

Apparently, in both (52a) and (52b), **how much money** is the closest WH to the target before any movement takes place. Thus, at first blush, it is surprising that there is not a significant acceptability contrast between (52a) and (52b) the same way that there is between (50a) and (50b).[25]

Thus, a WH that participates in a syntactic amalgam, showing up at the edge of an invasive clause, does not count as a WH for all intents and purposes. Further evidence for this is shown below.

below with Portuguese), where the analogues of (52a) and (52b) are both equally acceptable, which is consistent with the reasoning just sketched, since WH-movement must involve pied-piping in Romance (cf. **§II.8**).

(iii) Eu vou descobrir quanto dinheiro Bob deu você pode imaginar pra quem.

I will discover how-much money Bob gave you can imagine to who.

(iv) Eu vou descobrir pra quem Bob deu você pode imaginar quanto dinheiro.

I will discover to who Bob gave you can imagine how-much money.

[25] One could deny that this is a real problem under the assumption that **how much money** in (52b) is deeply embedded inside a complex constituent also containing the parenthetic-like string, as the brackets in (ib) indicate.

(i) a: I'll find out **[how much money]$_1$** Bob gave **t_1** to *[you can imagine* ***who]***

b: I'll find out **who$_2$** Bob gave *[you can imagine* ***how much money]*** to **t_2**

That way, despite what the linear- precedence dimension may suggest, **how much money** would arguably not count as the closest WH-phrase to the target in the relevant technical sense (since it would not scope over **who**), hence not counting as an intervener. This hypothesis will be addressed in chapter **V**, where I will present arguments that those two WH phrases are indeed in competition, and subject to superiority, but other structural variables (basically, shared constituency) plays a key role in these constructions, ultimately obfuscating superiority effects.

The paradigm in (53) involves an ordinary instance of indirect question, which requires a WH-phrase occupying a position (arguably, spec CP) in the left-periphery of the embedded clause. If there is a normal (non-amalgamated) WH-phrase fronted to the left-periphery of the embedded clause, as in (53a), the structure is acceptable. If the WH-phrase phrase fronted to the left-periphery of the embedded clause is affected by amalgamation, as in (53b), the structure is unacceptable to the same extent that it is unacceptable to have a non-WH phrase at the left-periphery of the embedded clause, as in (53c).

(53) a: Amy wonders **[how much money]**$_1$ Bob gave **t**$_1$ to Tom.
b: * Amy wonders *God knows* ***[how much money]***$_1$ Bob gave **t**$_1$ to Tom.
c: * Amy wonders **[some money]**$_1$ Bob gave **t**$_1$ to Tom.

In (54), we observe the opposite pattern. The embedded sentence is not an indirect question. Thus, there is no relevant element in the left periphery of the embedded sentence, which could license a WH-phrase. If one of the arguments of the verb of the embedded sentence is a WH-phrase *in situ*, as in (54a), the structure is unacceptable (unless it is associated with an echo-question interpretation and intonation). If, however, that very same WH-phrase not fronted to the left-periphery of the embedded clause participates in a syntactic amalgam, as in (54b), then the structure is acceptable,[26] to the same extent that it is acceptable to have a bona fide non-WH phrase in that position, as in (54c).

(54) a: * Amy believes Bob gave **[how much money]** to Tom.

b: Amy believes Bob gave **God knows *[how much money]*** to Tom.

c: Amy believes Bob gave **[some money]** to Tom.

II.7. On Possible and Impossible Target Positions for Clause Invasion

At first blush, it seems that syntactic amalgams exhibit no effects of a constraint on which point of the string a sentence may be invaded by another one sentence. Apparently, (the initial boundary of) any argument (or adjunct) of the invaded sentence can be targeted as the 'invasion point'.

The data below support this preliminary generalization.

(55) *Invasion at the Direct Object Position*

Tom believes that Amy has been dating *I forget **who*** since last October.

(56) *Invasion at the Indirect Object Position*

Tom believes that Amy gave all her money to *I forget **who*** yesterday.

[26] Notice that, contrary to (54a), an echo-question interpretation and intonation is not possible in (54b).

(57) *Invasion at the Adjunct (to VP) Position (with a governing preposition)*

Tom said that Amy has been dating Bob since *I forget* ***when.***

(58) *Invasion at the Adjunct (to VP) Position (without a governing preposition)*

Tom said that Amy met Bob *I forgot* ***when.***

(59) *Invasion at a Nominal Complement Position*

Tom believes that the general will demand the destruction of *I forget* ***which city*** by tomorrow morning.

(60) *Invasion at a Nominal Adjunct Position*

Tom said that the president will hire a person from *you'll never guess* ***which country*** for the job of secretary of international affairs.

However, upon closer inspection, we notice that there is in fact a limit to this freedom, as shown in (61) and (62), which are both unacceptable.[27]

[27] The examples in (61) and (62) are totally unacceptable under the relevant interpretations, which correspond to the paratactic paraphrases in (i) and (ii), respectively:
(i) Tom said that a given person is dating Amy. I forgot who that person is.
(ii) Tom said that a given person kissed Amy at the party. I forgot who that person is.
However, the same strings of words in (61) and (62) are acceptable under distinct interpretations, which correspond to the hypotactic paraphrases in (iii) and (iv), respectively:
(iii) Tom said that I forgot the identity of the person who is dating Amy.
(iv) Tom said that I forgot the identity of the person who kissed Amy at the party.
The acceptability/grammaticality status of (61) and (61) under the readings in (iii) and (iv) is irrelevant for the present discussion, since, in that case, the corresponding syntactic representations would arguably be something along the lines of (v) and (vi), respectively, which definitely are instances of ordinary subordination rather than syntactic amalgamation.
(v) [^{CP} [^{IP} Tom [^{VP} said [^{CP} that [^{IP} I [^{VP} forgot [^{CP} **who$_1$** [^{IP} **t$_1$** is dating Amy]]]]]]]]

(61) *Invasion at the Subject Position (active structure)*

* Tom said that *I forgot* ***who*** is dating Amy**.**

(62) *Invasion at the Subject Position (passive structure)*

* Tom said that *I forgot* ***who*** was kissed by Amy at the party**.**

What distinguishes all the acceptable examples in (55-60) from the unacceptable ones in (61-62) is the fact that, in the former group, the constituent that defines the target of the 'clause invasion' is a complement of either a verb or a preposition,[28] as opposed to the latter group. Therefore, there is something special about the subject position that makes it an invalid target for 'clause invasion'.

Notice, however, that it is possible for clause invasion to target a subject position that is associated with E(xeptional) C(ase) Marking, as shown in (64).

(63) *Invasion at the Subject Position (ECM structure)*

The conductor of the orchestra wants *you'll never guess* ***which musician*** to be in charge of the rehearsal while he will be out of town.

(vi) [^{CP} [^{IP} Tom [^{VP} said [^{CP} that [^{IP} I [^{VP} forgot [^{CP} **who**$_{\mathbf{1}}$ [^{IP} **t**$_{\mathbf{1}}$ was kissed **t**$_{\mathbf{1}}$ by Amy at the party]]]]]]]]

[28] Actually, the example in (58) is an exception exception to this generalization, as it shows clause invasion targeting a 'bare adverb', not governed by a preposition. For further discussion, cf. chapter **V**.

The same pattern shown in (55) through (64) above obtains in cleft amalgams, as shown below.

(64) *Invasion at the Direct Object Position*

Tom believes that Amy has been dating *I think it's* ***Bob*** since last October.

(65) *Invasion at the Indirect Object Position*

Tom believes that Amy gave all her money to *I think it's* ***Bob*** yesterday.

(66) *Invasion at the Adjunct (to VP) Position (with a governing preposition)*

Tom said that Amy has been dating Bob since *I think it's* ***last October.***

(67) *Invasion at the Adjunct (to VP) Position (without a governing preposition)*

Tom said that Amy met Bob *I think it was* ***last October.***

(68) *Invasion at a Nominal Complement Position*

Tom believes that the general will demand the destruction of *I think it's* ***Tehran*** by tomorrow morning.

(69) *Invasion at a Nominal Adjunct Position*

Tom said that the president will hire a person from *I think it's* ***Holland*** for the job of secretary of international affairs.

(70) *Invasion at the Subject Position (active structure)*

* Tom said (that) *I think it's **Bob*** is dating Amy**.**

(71) *Invasion at the Subject Position (passive structure)*

* Tom said (that) *I think it's **Bob*** was kissed by Amy at the party**.**

(72) *Invasion at the Subject Position (ECM structure)*

The conductor of the orchestra wants *I think it's **Mr. Petrovic*** to be in charge of the rehearsal while he will be out of town.

II.8. Cross-Linguistic Word Order Variation

As shown in **§II.7**, the position of object of a preposition is a possible target for clause invasion, as exemplified in (73) and (74).

(73) John invited 300 people <u>to</u> *you can imagine <u>what kind of party</u>*.

(74) John has been planning his 40th birthday party <u>since</u> *you can imagine <u>when</u>*.

Examples of this kind have been previously described and analyzed by Lakoff (1974), without being given any special status. However, something that has gone unnoticed is the fact that this subclass of syntactic amalgam displays effects of parametric variation.

The same construction exists in other languages, but there is variation with respect to the relative order between the relevant preposition and the invasive clause.

For comparison, let us first exhaust the description of the English paradigm. As shown in (75) and (76), in English, the invasive clause must appear in between the relevant preposition and its complement, as in (75a) and (76a), so that the PP becomes a discontinuous constituent at PF. If the invasive clause appears before the preposition, as in (75b) and (76b), the structure unacceptable.

(75) a: John invited 300 people to *you can imagine what kind of party*.

b: * John invited 300 people *you can imagine* to *what kind of party*.

(76) a: John has been planning his 40th birthday party since *you can imagine when*.

b: * John has been planning his 40th birthday party *you can imagine* since *when*.

In other languages, the opposite pattern obtains, as illustrated below with examples from Romance (cf. (77) and (78)).[29] In such languages, it is not possible for the invasive clause to appear in between the relevant preposition and its complement, as in (77a) and (78a). Rather, the invasive clause must appear immediately before the preposition, as in (77b) and (78b).

(77) *Romance (Portuguese)*

a: * João convidou 300 pessoas pra você pode imaginar que tipo de festa
John invited 300 persons to you can imagine what kind of party

b: João convidou 300 pessoas você pode imaginar pra que tipo de festa
John invited 300 persons you can imagine to what kind of party

(78) *Romance (Portuguese)*

a: * João vem planejando a festa de 40˚ aniversário dele desde você
John has planned the party of 40th birthday of+him since you
pode imaginar quando
can imagine when

b: João vem lanejando a festa de 40˚ aniversário dele você pode
John has planned the party of 40th birthday of+him you can
imaginar desde quando
imagine since when

[29] I have chosen to illustrate the point with examples from Portuguese, but the same pattern obtains in Spanish, Galician, French, as well as in other languages pied-piping languages outside the Romance family, like Polish and Russian.

A generalization that can be drawn from the data of the languages I observed (and which may be further supported or refuted by future comparative studies) is that there is a strong correlation between the word-order patterns above and whether the language allows preposition stranding in WH-movement (like English) or not (like Romance and Slavic in general), as the data below indicate. This correlation will be analyzed in chapters **III** and **V**.

(79) *English*

a: What$_1$ are you talking about **t$_1$** ?

b: * [About what]$_1$ are you talking **t$_1$** ?

(80) *Romance (Portuguese)*

a: * [O quê]$_1$ você está falando sobre **t$_1$** ?

What$_1$ you are talking about **t$_1$** ?

b: [Sobre o quê]$_1$ você está falando **t$_1$** ?

about what$_1$ you are talking **t$_1$** ?

The generalization just presented deserve further comment, as far the English facts are concerned. At first blush, there seems to be no one-to-one correspondence in English between the pied-piping/preposition-stranding distinction and the order of the preposition with respect to the invasive clause in syntactic amalgams. It is a well-known fact that, although pied-piping the whole

PP is worse than stranding the preposition in cases like (81); there are other cases where the contrast is not nearly as strong. For instance, in (82), although the preposition-standing strategy is, by far, more acceptable, the pied-piping strategy is by no means as unacceptable as it is in (81b). In fact, (82b) is relatively acceptable despite its heavily marked status.

(81) a: What$_1$ are you talking about t$_1$?

b: * [About what]$_1$ are you talking t$_1$?

(82) a: Who$_1$ are you talking to t$_1$?

b: ? [To whom]$_1$ are you talking t$_1$?

This can be easily accommodated if the generalization is stated in terms of 'availability of preposition stranding (or lack thereof)' rather than 'availability of pied piping (or lack thereof)'. However, as it will become clear when I discuss this at the analytical level (cf. chapters **III** and **V**), stating the generalization in terms of 'availability of preposition stranding (or lack thereof)' is nothing but a mere rhetorical move that has the negative effect of biasing the analysis towards a model that lacks explanatory power. *Prima facie*, if some pied-piping is possible in English, and if the suggestion I just made that the order of the preposition relatively to the invasive clause in syntactic amalgams correlates with the pied-piping/preposition-stranding distinction, then we would expect to find some

Romance-style syntactic amalgams in English to the same extent that we find pied-piping in analogous constructions. But this is not the case. For instance, while (83a) merely has the status of marginal or 'too formal' in comparison to (83b), its amalgamated version in (84a) — which inherits its pied-piping configuration — is considerably degraded in comparison with both (83a) and (84b) to most speakers.

(83) a: ? I forgot to whom Mr. Smith was speaking after the meeting.
b: I forgot who Mr. Smith was talking to after the meeting.

(84) a: * Mr. Smith was speaking I forgot to whom after the meeting.
b: Mr. Smith was talking to I forgot who after the meeting.

In this dissertation, I endorse Murphy's (1995) position that English is essentially a full preposition-stranding language, with all instances of pied-piping being an artifact of E-language.[30] That is, there would be code-switching between (at least) two distinct grammars with distinct parametric settings. One of them is 'actual English', and is spoken in most situations. The other one is 'formal English', and its usage is *"reserved for literary diction"* (cf. Visser 1968:

[30] I am thankful to Anthony Kroch, Colin Phillips and Paula Kempchinsky for discussion on this idea of English pied piping being an E-language artifact.

406), and requires conscious application of a prescriptive rule learned at school, in a self-monitoring fashion.[31]

Interestingly, even when one is speaking this literary dialect, the preference for pied-piping does not carry over to all cases. Some instances of pied-piping are just plain unacceptable despite one's urge to abide by the rules of 'proper grammar', as shown in (85) and (86).[32]

(85) a: What$_1$ did you see a picture of t$_1$?
b: * [Of what]$_1$ did you see a picture t$_1$?
c: * [A picture of what]$_1$ did you see t$_1$?

[31] It is true, however, that, in a few cases, pied-piping seems to be quite present even in colloquial speech. Nevertheless, as Murphy (1995: 74) points out, there is reason to believe that those cases fall outside the core grammar, rather being artifacts of peripheral glitches. On this matter, Murphy says:

> *"There are other cases where pied piping seems to have been more widespread, as in certain relatives like he is a man in whom I trust. Interestingly, Visser notes that in Old English there was a 'condensed' relative construction hwan (= him…whom) that was preceded by the preposition (1968: 400). Quite possibly the 'pied piping' is not movement at all, but rather was the result of the loss of the object of the preposition, and a reanalysis whereby the preposition is associated with the verb of the relative clause instead of that of the matrix clause. At any rate, there are signs that the morphological marking of who (whom) was already fading during the Old English period."*

[32] As Murphy (1995: 72) observes, some instances of heavy pied piping that are unacceptable in matrix questions become significantly more acceptable (in 'literary diction' contexts) if they appear in embedded domains, as shown in (i) below.

(i) a: I met the woman [the picture of whom]$_1$ John saw t$_1$.
b: I met the woman [proud of whom]$_1$ John is not. t$_1$.

Notice, however, that this is only possible in relative clauses. Importantly, the same flexibility does not exist in indirect questions, as shown in (ii). Not surprisingly, the corresponding syntactic amalgams are equally unacceptable, as shown in (iii).

(ii) a: * I wonder [the picture of whom]$_1$ John saw t$_1$
b: * I wonder [proud of whom]$_1$ John is t$_1$.

(iii) a: * John saw I wonder the picture of whom
b: * John is I wonder proud of whom.

(86) a: Who$_1$ is John proud of t$_1$?

b: * [Of whom]$_1$ is John proud t$_1$?

c: * [Proud of whom]$_1$ is John t$_1$?

Therefore, it is reasonable to assume that the possibility of not stranding the preposition in English is truly a peripheral E-language artifact. Those speakers who are more into 'literary diction' can train themselves to master the pied-piping strategy to the same extent that one can become relatively fluent in a foreign language.

Some speakers can produce and comprehend sentences like (82b) and (83a) quite naturally, as those do not exhibit much structural complexity. On the other hand, syntactic amalgamation is quite demanding for the parser, due to its arguably high structural complexity that goes beyond the hypotactic level. Presumably, that is just too much for the ordinary non-native speaker of the 'literary dialect' to handle, and, at this point, his/her intuitions will reflect his/her native dialect, hence the unacceptability of Romance-style amalgams by even 'highly educated' English speakers.

Not surprisingly, there are a few speakers who are much more fluent in their 'second language', up to the point of judging Romance-style amalgams like (75b) and (76b) — repeated below as (87a) and (87b), respectively — as merely marginal (even if slightly so), instead of completely unacceptable.[33]

[33] A distinguishing aspect that I found about of all of my informants who fall into this category is that they are all extremely fluent (*quasi*-bilingual) speakers of at least one dialect of Romance.

(87) a: * John invited 300 people *you can imagine* to *what kind of party*.

b: * John has been planning his 40th birthday party *you can imagine* for *how many years*.

The comparison between this subclass of amalgams and *bona fide* parentheticals with regards to this word-order is quite revealing. Although the English cases of parentheticals do not exhibit any contrast with amalgams, as shown in (88), the Romance data show a word-order pattern distinct from the one found in amalgams, as shown in (89).[34]

(88) a: John invited 300 people to – as we already suspected – a wild party.

b: * John invited 300 people – as we already suspected – to a wild party.

(89) a: João convidou 300 pessoas pra, como a gente já suspeitava, uma
John invited 300 people to, as we already suspected, a
festa do cabide.
party of+the hanger.
'John invited 300 people to – as we already suspected – a wild party'

Thus, presumably, their judgments are very likely to be the result of 'overlapping intuitions', where one grammar 'contaminates' the other. Further research is necessary in order to figure out whether this is a systematic pattern, or just a coincidential idyossincrasy of my data sample.

[34] The examples in (88b) and (89b) are not acceptable under the relevant interpretation, in which the content of the parenthetical is a comment on the kind of party. The same examples become acceptable under an alternative (non-relevant) interpretation, in which the content of the parenthetical is a comment on the number of guests.

b: * João convidou 300 pessoas, como a gente já suspeitava, pra uma
John invited 300 people, as we already suspected, to a
festa do cabide.
party of+the hanger.

This contrast in Romance further supports the view that invasive clauses of amalgams are not genuine parentheticals.[35]

There is yet one third word-order pattern that deserves to be mentioned and looked at carefully, so that we can tease apart the relevant and the irrelevant data. In Romance, it is possible for the invasive clause to appear both before and after the preposition. That is, the preposition may be pronounced twice, one token before and the other one after the invasive clause, as shown in (90) and (91).[36]

(90) *Romance (Portuguese)*
João convidou 300 pessoas pra você pode imaginar pra que tipo de festa
John invited 300 persons to you can imagine to what kind of party

[35] Another kind of evidence for that lies in their rather distinct prosodic structures, which I will not discuss here.
[36] I am thankful to Leticia Pablos for discussion on this matter.

(91) *Romance (Portuguese)*

João vem planejando a festa de 40˚ aniversário dele desde você pode
John has planning the party of 40th birthday of+him since you can
imaginar desde quando
imagine since when

These constructions are possible in Romance only if the content of the invasive clause is presented as an afterthought, with a major intonational break (typical of hesitation, suspense or memory lapse) after the first occurrence of the preposition, followed by an increase in speed. The degree of acceptability of these examples is directly proportional to how fast the string of words in the invasive clause is pronounced, and how salient the intonational break right after first occurrence of the preposition is (measurable mostly by the degree of lengthening of the segmental material of the preposition, and the by the identification of the proper intonational curve).

The heavily marked status of these examples seems to be related to performance factors triggering their usage. Typically, this construction emerges in situations where the speaker does not initially intend to produce a syntactic amalgam, but, at the point he/she reaches the preposition, (s)he changes his/her mind and decides to make a comment about the entity denoted by the object of that preposition. In doing so, the main sentence is not just interrupted, but also abandoned, and followed by that afterthought, whose structure is typical of a sluiced sentence. Thus, the evidence points to the direction that cases of preposition-doubling in (90) and (91) are parentheticals, rather than amalgams.

Not surprisingly, all that I said above about WH-amalgams applies to cleft amalgams, as shown below.

(92) *English*

a: John will travel to I think it's Chicago tomorrow.

b: * John will travel I think it's to Chicago tomorrow.

(93) *Romance (Portuguese)*

a: * João vai viajar pra eu acho que é Curitiba amanhã.

John will travel to I think that is Curitiba tomorrow.

b: João vai viajar eu acho que é pra Curitiba amanhã.

John will travel I think that is to Curitiba tomorrow.

Finally, let us take a quick look at another fact about syntactic amalgams where the 'invasion' targets the object of a preposition.[37]

A quite idiosyncratic fact about English is that, in sluiced sentences where the WH phrase is the object of a preposition, that preposition may be pronounced at the end of the word string, right after the WH-phrase, as if the PP,

[37] I am thankful to Satoshi Tomioka and Andrew Nevins for discussion on this matter.

exclusively, were somehow linearized as in head-final languages. This is shown in (94).

(94) John danced at the party. But I don't remember who with.

Descriptively speaking, this pattern seems to involve a special kind of ellipsis of the sluiced material, so that the preposition is left pronounced for some reason, as indicated in (95).

(95) John danced at the party. But I don't remember **who$_1$** ~~John danced~~ with **t$_1$** ~~at the party~~.

As already discussed in chapter **I**, and as will be further discussed in chapter **III**, syntactic amalgams may be potentially analyzed in terms of sluicing. From that perspective, it is not obvious, at first blush, why the possibility of the word-order pattern illustrated in (94) does not carry over to syntactic amalgams, as shown in (96).

(96) * John danced *I don't remember who with* at the party.

II.9. Co-reference Possibilities Within Syntactic Amalgams

Another property of syntactic amalgamation concerns co-reference possibilities among pronouns and R-expressions that are distributed one in the 'invasive clause' and the other in the 'invaded clause'. In all cases, the co-reference possibilities for any given syntactic amalgam mimics exactly the readings available in the corresponding paratactic paraphrase, rather than the readings available in the corresponding hypotactic paraphrase, as shown below.

First, consider the case of potential co-reference between a pronoun in the invasive clause and an R-expression in the invaded clause, as in (97).

(97) a: **[Homer]**$_1$ drank **[he]**$_{1/2}$ doesn't even remember how many beers at the party.

b: **[He]**$_{*1/2}$ doesn't even remember how many beers **[Homer]**$_1$ drank at the party.

c: **[Homer]**$_1$ drank beers at the party. **[He]**$_{1/2}$ doesn't even remember how many.

Now, take the case of potential co-reference between an R-expression in the invasive clause and a pronoun in the invaded clause, as in (98).

(98) a: **[He]**$_{*1/2}$ drank **[Homer]**$_1$ doesn't even remember how many beers at the party.

b: **[Homer]**$_{1}$ doesn't even remember how many beers **[he]**$_{1/2}$ drank at the party.

c: **[He]**$_{*1/2}$ drank beers at the party. **[Homer]**$_{1}$ doesn't even remember how many.

The paradigm in (99) is similar to the one in (97), except that the pronoun in the invasive clause is embedded inside a more complex NP/DP. In this case, all potential co-reference possibilities are attested, making both paraphrases accurate.

(99) a: **[Homer]**$_{1}$ drank I bet **[[his]**$_{1/2}$ **wife]** remembers how many beers at the party.

b: I bet **[[his]**$_{1/2}$ **wife]** remembers how many beers **[Homer]**$_{1}$ drank at the party.

c: **[Homer]**$_{1}$ drank beers at the party. I bet **[[his]**$_{1/2}$ **wife]** remembers how many.

The paradigm in (100), in its turn, is similar to the one in (98), except that the pronoun in the invaded clause is embedded inside a more complex NP/DP. Again, all potential co-reference possibilities are attested, making both paraphrases accurate in this case too.

(100) a: **[[His]$_{1/2}$ wife]** drank I bet **[Homer]$_{1}$** remembers how many beers at the party.

b: I bet **[Homer]$_{1}$** remembers how many beers **[[His]$_{1/2}$ wife]** drank at the party.

c: **[[His]$_{1/2}$ wife]** drank beers at the party. I bet **[Homer]$_{1}$** remembers how many.

The paradigm in (101) contains two R-expressions: one in the invaded clause, and the other one in the invaded clause. Similarly to what happens in (97a) and (98a), the co-reference possibilities in the amalgam in (101) match the ones in the corresponding paratactic paraphrase, rather than the ones in the hypotactic paraphrase.

(101) a: **[Homer]$_{1}$** drank **[the idiot]$_{1/2}$** doesn't even remember how many beers at the party.

b: **[The idiot]$_{*1/2}$** doesn't even remember how many beers **[Homer]$_{1}$** drank at the party.

c: **[Homer]$_{1}$** drank beers at the party. **[The idiot]$_{1/2}$** doesn't even remember how many.

Finally, consider the paradigm in (102). Structurally, it is identical to the one in (101), except that the two R-expressions switch positions. Again, the co-reference possibilities in the amalgam match the ones in the corresponding paratactic paraphrase, rather than the ones in the hypotactic paraphrase.

(102) a: **[The idiot]**$_{*1/2}$ drank **[Homer]**$_{1}$ doesn't even remember how many beers at the party.

b: **[Homer]**$_{1}$ doesn't even remember how many beers **[the idiot]**$*_{1/2}$ drank at the party.

c: **[He]**$_{*1/2}$ drank beers at the party. **[Homer]**$_{1}$ doesn't even remember how many.

II.10. The Matrix-clause Behavior of Invaded and Invasive Clauses

Yet another indication of the paratactic nature of syntactic amalgams comes from the fact that both the invaded clause and the invaded clause(s) behave as matrix clauses, as I show below.

In (103) and (104), we see that the quasi-parenthetic 'invasive' clause may exhibit syntactic patterns found only in matrix clauses, like auxiliary-inversion for questions, or imperative mood.

(103) [Bob told me that Amy danced with [*do you know who?*] at the party]

(104) [Bob told me that Amy danced with [*guess who!*] at the party]

Another piece of evidence that invasive clauses are not embedded clauses comes from Brazilian Portuguese, where – unlike in most Romance languages – gaps in the position of a (3rd person) subject are licensed only in certain specific kinds of embedded clauses, as in (105), but never in matrix clauses, as in (106)[38].

(105) *Brazilian Portuguese*

a: Maria$_1$ não se lembra quantos homens ela$_{1/2}$ beijou na festa.

Mary$_1$ not REFL remember how+many men she$_{1/2}$ kissed at+the party.

'Mary$_1$ doesn't remember how many men she$_{1/2}$ kissed at the party'

b: Maria$_1$ não se lembra quantos homens *e*$_{1/*2}$ beijou na festa.

Mary$_1$ not REFL remember how+many men ∅$_{1/*2}$ kissed at+the party.

'Mary$_1$ doesn't remember how many men she$_{1/*2}$ kissed at the party'

[38] For a complete analysis of the licensing and distribution of gaps in subject position in Brazilian Portuguese, as well as their morpho-syntactic nature, and the constraints of their reference, see Rodrigues (2002, 2004). For the present purposes, the descriptive generalization above suffices. I am extremely thankful to Juan Uriagereka, and, especially, Cilene Rodrigues, for discussion on the data in this section.

(106) *Brazilian Portuguese*

a: Maria$_1$ beijou muitos homens na festa. Ela$_1$ nem se lembra quantos.
Mary kissed many men at+the party. She not+even REFL remember how+many
'Mary kissed many men at the party. She doesn't even remember how many'

b: * Maria$_1$ beijou muitos homens na festa. *e*$_1$ nem se lembra quantos.
Mary kissed many men at+the party. ∅ not+even REFL remember how+many
'Mary kissed many men at the party. She doesn't even remember how many'

In this regard, the 'invasive' clauses of syntactic amalgams behave exactly as matrix clauses, as no gap in subject position is possible there, as shown in (107).

(107) *Brazilian Portuguese*

a: Maria$_1$ beijou ela$_1$ nem se lembra quantos homens na festa.
Mary kissed she not+even REFL remember how+many men at+the party
'Mary kissed she doesn't even remember how many men at the party'

b: * Maria$_1$ beijou e_1 nem se lembra quantos homens na festa.

Mary kissed $\varnothing$ *not+even REFL remember how+many men at+the party*

'Mary kissed she doesn't even remember how many men at the party'

Notice the contrast between Brazilian Portuguese – cf. (105), (106) and (107) above– and *bona fide* pro-drop Romance languages like Galician – cf. (108), (109) and (110) below – and Spanish – cf. (111), (112) and (113) below.

(108) *Galician*

a: * María$_1$ non se lembra cántos homes ela$_{1/2}$ bicou na festa.

Mary$_1$ *not REFL remember how+many men she*$_{1/2}$ *kissed at+the party.*

'Mary$_1$ doesn't remember how many men she$_{1/2}$ kissed at the party'

b: María$_1$ non se lembra cántos homes $e_{1/2}$ bicou na festa.

Mary$_1$ *not REFL remember how+many men* $\varnothing_{1/*2}$ *kissed at+the party.*

'Mary$_1$ doesn't remember how many men she$_{1/*2}$ kissed at the party'

(109) *Galician*

a: ? María$_1$ bicou moitos homes na festa... Ela$_{1/2}$ nin se lembra cántos.

Mary kissed many men at+the party. She not+even REFL remember how+many

'Mary kissed many men at the party. She doesn't even remember how many'

b: María$_1$ bicou moitos homes na festa... *e*$_1$ nin se lembra cántos.

Mary kissed many men at+the party. ∅ not+even REFL remember how+many

'Mary kissed many men at the party. She doesn't even remember how many'

(110) *Galician*

a: *? María$_1$ bicou ela$_1$ nin se lembra cántos homes na festa.

Mary kissed she not+even REFL remember how+many men at+the party

'Mary kissed she doesn't even remember how many men at the party'

b: María$_1$ bicou *e*$_1$ nin se lembra cántos homes na festa.

Mary kissed ∅ not+even REFL remember how+many men at+the party

'Mary kissed she doesn't even remember how many men at the party'

(111) *Spanish*

a: * María$_1$ no se acuerda cuántos hombres ella$_{1/2}$ besó en la fiesta.

Mary$_1$ not REFL remember how+many men she$_{1/2}$ kissed at the party.

'Mary$_1$ doesn't remember how many men she$_{1/2}$ kissed at the party'

b: María$_1$ no se acuerda cuántos hombres *e*$_{1/2}$ besó en la fiesta.

Mary$_1$ not REFL remember how+many men $\varnothing_{1/*2}$ *kissed at+the party.*

'Mary$_1$ doesn't remember how many men she$_{1/*2}$ kissed at the party'

(112) *Spanish*

a: María$_1$ besó muchos hombres en la fiesta... Ella$_{1/2}$ ni se acuerda cuántos.

Mary kissed many men at+the party. She not+even REFL remember how+many

'Mary kissed many men at the party. She doesn't even remember how many'

b: María$_1$ besó muchos hombres en la fiesta... *e*$_1$ ni se acuerda cuántos.

Mary kissed many men at+the party. $\varnothing$ *not+even REFL remember how+many*

'Mary kissed many men at the party. She doesn't even remember how many'

(113) *Spanish*

a: ? María$_1$ besó ella$_1$ ni se acuerda cuántos hombres en la fiesta.

Mary kissed she not+even REFL remember how+many men at the party

'Mary kissed she doesn't even remember how many men at the party'

b: María$_1$ besó *e*$_1$ ni se acuerda cuántos hombres en la fiesta.

Mary kissed ∅ not+even REFL remember how+many men at the party

'Mary kissed she doesn't even remember how many men at the party'

Finally, it is worth emphasizing that what we have been describing as the 'invaded clause' also shares with 'invasive clauses' the property of licensing certain syntactic patterns found only in matrix clauses, like auxiliary-inversion for questions, or imperative mood, as shown in (114).

(114) a: [Go tell Bob that Amy gave all her money to [Do you still remember who?]!]

b: [Go tell Bob that Amy gave [Do you still remember how much money?] to Tom!]

III

(Neo)Conservative Approaches to Syntactic Amalgamation

This chapter is dedicated to a detailed presentation and discussion of Lakoff's (1974) seminal work on syntactic amalgams. I will first introduce the mechanics of his analysis vis-à-vis the original framework it was proposed, and its historical moment. Then I will elaborate on the main consequences of that kind of formalism in the context of recent developments of the Theory of Grammar, which includes discussion Tsubomoto &Whitman's (2000) work. Finally, I will evaluate that traditional approach for descriptive adequacy, on the basis of the facts presented in chapter **II**, and for explanatory adequacy, on the basis of minimalist criteria. After attempting to translate this traditional approach into an analysis that is commensurable with the contemporary Principles & Parameters metalanguage, I will eventually conclude that the general approach to syntactic amalgamation proposed by Lakoff and further worked out by Tsubomoto &Whitman ultimately fails to meet both descriptive and explanatory adequacy, and needs to undergo radical revision, which I will leave for the subsequent chapters.

III.1. Avoiding a Constituency Paradox by Postulating Extra Hidden Structure: a brief overview of the traditional analysis of amalgamation

One puzzling aspect of syntactic amalgams is the fact that, at first blush, they seem to involve a paradoxical constituency, in which the container would somehow be inside the content. In other words, although it is clear that the whole construction involves two (or more) sentences standing in a subordination relation, it is not obvious, without any further systematic investigation, which clause is the matrix and which is the embedded one.

For instance, let us take a closer look at the example (01), which is originally due to Avery Andrews.

(01) John invited you'll never guess who to his party.

From a naïve perspective, this construction seems to be built around a matrix clause structured as sketched in (02a), conveying the main message that John invited X to the party, where X stands for a person whose identity the speaker takes to be impossible for the listener to figure out. The syntactic material of X would be as in (02b).

(02) a: [^{IP} John [^{VP} invited **X** [^{PP} to his party]]]

b: **X** = [you'll never guess who]

By this reasoning, the substring **you'll never guess who** is a constituent. If so, what kind of constituent is it? In order for the selectional requirements of *invited* to be satisfied, X must be an NP, in which case **who** would be the head of the structure, whereas **you'll never guess** would be a complex modifier of some sort, as in (03).

(03) [^{IP} John [^{VP} invited [^{NP} [^{X} you'll never guess] [$^{N'}$ who]] [^{PP} to his party]]]

But such a structure is problematic to the extent that the selectional requirements of **guess** are not being satisfied. Therefore, the idea of taking the string **you'll never guess who** to be a constituent is impractical, at least under a generative-transformational approach.[39]

An alternative would be to postulate a structure like (04), in which some of the material is duplicated, and **guess** takes the whole clause **John invited who**

[39] If we assume a Categorial Grammar approach (Ajdukiewicz 1935; Bar-Hillel 1953; Steedman 1996, 2000; *inter alia*), with radical type-shifts, there is room for an analysis along the lines suggested in (03). However, that doesn't immediately solve the basic problem in any trivial way. In principle, one could come up with a combination of type-shift mechanisms that could make possible to combine **you'll never guess** with **who**, yielding **you'll never guess who**, which would eventually act as an argument of **invited**. However, as far as the semantic interpretation is concerned, **who** alone cannot be the argument of **guess**. We need the whole sentence **John invited who to his party** to be taken as an argument of **guess**. Evidence for this comes from the fact that examples like (i) are ungrammatical.

(i) * How many people will you never guess?

Syntactically, the discontinuous string **John invited ... to his party** cannot be inserted into **you will never guess who**. So, this integration has to be an effect of semantic interpretation, which, as far as I can see, cannot be trivially achieved without further assumptions.

to his party as its clausal complement, within which **who** undergoes local WH-movement and the IP undergoes internal sluicing.[40]

(04) [$^{\text{IP}}$ John [$^{\text{VP}}$ invited [$^{\text{IP}}$ you'll [$^{\text{VP}}$ never guess [$^{\text{CP}}$ [$^{\text{NP}}$ who] [$^{\text{IP}}$ ~~John invited who to his party~~]]]] [$^{\text{PP}}$ to his party]]]

This is consistent with the internal semantic structure of the 'constituent X'. What the listener will never guess is not just the identity of a person x, but rather the identity of the person x such that John invited x to his party. However, this alternative analysis solves one problem by creating another one of the same kind. If **who** is deeply embedded inside the complement of **guess**, as in (04), then the 'constituent X' is not an NP. Thus, the selectional requirements of **invited** are not being satisfied.

A way to satisfy the selectional requirements of both **invited** and **guess** is to adopt a more elaborated version of (04), as did Tsubomoto &Whitman (2000), piggybacking on Lakoff's original insight.[41] From that perspective, the core

[40] It may seem, at first blush, that the complement of **guess** is **who**, instead of a more complex structure with **who** in it. But the impossibility of structures like the ones in (i) indicates otherwise.

(i) a: * How many people will you never guess?
b: * You will never guess 300 people.

This is a general property of the class of verbs that appear in those 'parenthetic-like strings' of amalgams (e.g. **guess**, **wonder**, **imagine**, **ask**). Under the relevant readings, they select only CPs as their complements, rather than pure DPs. For instance the construction in (ii) is possible but the ones in (iii) are not..

(ii) Homer drank I wonder how many beers at the party.
(iii) a: * I wonder 75 beers.
b: * How many beers do you wonder?

[41] Lakoff's insight is summarized the following passage: *"By 'syntactic amalgam' I mean a sentence which has within it chunks of lexical material that do not correspond to anything in the logical structure of*

structure of (01) would be (5a), where the direct object is an elliptical indefinite NP (perhaps a PF-deleted version of **someone**). A subsidiary structure (05b) is built in parallel and it further undergoes sluicing and adjoins to the elliptical NP inside (05a), in a generalized-transformational fashion, finally yielding (05c).

(05) a. [^{IP} John invited **[^{NP} *e*]** to his party]

b. [^{IP} you'll never guess [^{CP} [who]$_1$ [^{IP} ~~John invited t_1 to his party~~]]]

c. [^{IP} John invited **[^{NP} [^{NP} *e*]** [^{IP} you'll never guess [^{CP} [who]$_1$ [^{IP} ~~John invited t_1 to his party~~]]]**]** to his party]

Looking at syntactic amalgams from another angle, another possibility is that the structure behind (01) is actually something like (06). That way, all selectional requirements of all predicates are satisfied straightforwardly.

(06) [^{IP} you will never guess [^{CP} [^{DP} how many people]$_1$ John invited t_1 to his party]]

This cannot be the whole story, however. The precedence relations among the words in (06) radically differs from the ones in (01). The null hypothesis, then, is that the word-order pattern in (01) is associated with another phrase marker, which is derived from (06) through a combination of movements.

the sentence; rather they must be copied in from other derivations under specifiable semantic and pragmatic conditions". Lakoff (1974: 321)

This kind of approach – which was explicitly rejected by Lakoff (1974) – will be given a try in **§III.3.2 and §III.3.4**, by means of four different alternative technical implementations based on remnant movement (Müller 1998). Eventually, I will conclude that, although some of its features are on the right track, this analysis needs to undergo major change in order to account for the full range of empirical facts described in chapter **II**.

III.2. The Mechanics of Lakoff's (1974) 'Classical Analysis'

III.2.1. Amalgamation Rules

According to Lakoff (1974), the generation of syntactic amalgams involves rules like the one in (07).[42]

[42] As I mentioned in chapter **II** (footnote 4), Lakoff (1974) recognizes six different kinds of syntactic amalgam. For each one, he postulates a rule along the lines of (06), except for tag questions, which he leaves without a technical implementation. Each amalgamation rule has its own idiosyncrasies, and is explicitly stated as being sensitive to construction-specific structural properties. However, all rules share the following features. They require the existence of three sentences S_0, S_1 and S_2, and an NP_1, such that S_0 is embedded within S_1, with S_2 being a separate sentence, and NP_1 being a constituent of S to be replaced with a reduced version of S_1 without S_0. Also, all amalgamation rules require that S_1 entails S_2 for them to apply. The list of all amalgamation rules proposed by Lakoff (1974) is given in the appendix.

(07) For all contexts C, if (a) & (b) & (c) & (d), then (e):

a: S_1 is an indirect question with S_0 as its complement S;

b: S_2 is the i^{th} phrase marker in a derivation D whose logical structure is conversationally entailed[43] by the logical structure of S_1 in context C;

c: NP_1 is an NP in S_2, such that S_2 minus NP_1 is identical to S_0;

d: S_1 has the force of an exclamation;

e: relative to context C, S_1 minus S_0 may occur in place of NP_1 in the $i+1^{th}$ phrase marker of derivation D.

To see how the rule above works, consider the example in (08).

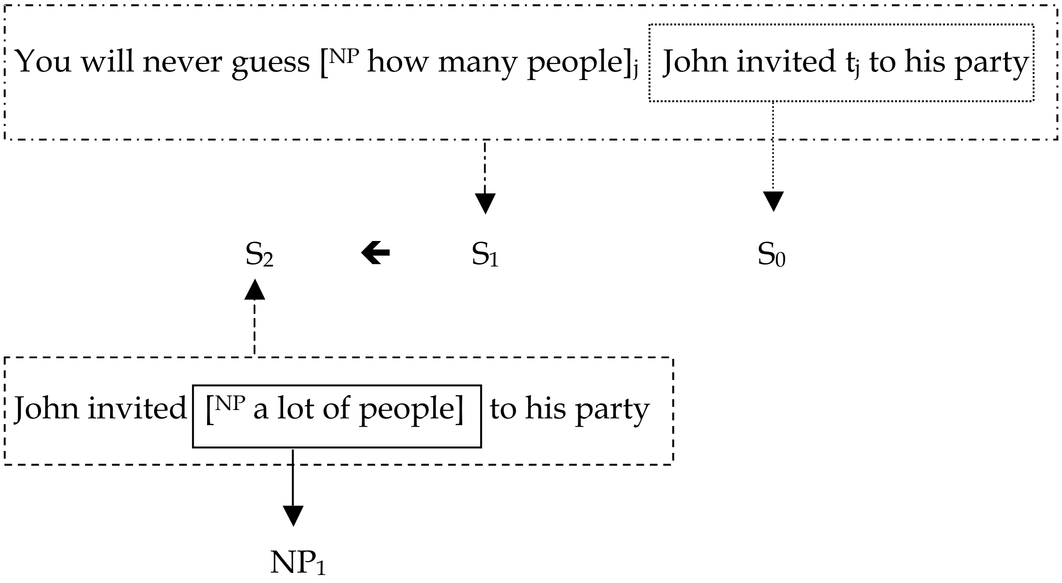
(08) John invited you will never guess how many people to his party.

In this case, the particular syntactic structures corresponding to S_0, S_1, S_2 and NP_1 are the ones shown in (09).[44]

[43] This entailment is indicated in (08) by the symbol "🡐" (meaning that what comes to the left of the arrow is entailed by what comes to the right of the arrow).

[44] For expository reasons, I decided to use a trace in the notation in (09) — as well as in (12) — to indicate the movement of **how many people** within S_0. Keep in mind, though, that Lakoff's (1974) analysis does not involve traces at all, given the framework in which it was formulated.

(09)


You will never guess [NP how many people]j John invited tj to his party
S2
S1
S0
John invited [NP a lot of people] to his party
NP1

Since S_0 = **[John invited t_j to his party]** is embedded within S_1 = **[^{S} you will never guess [^{NP} how many people]$_j$ John invited t_j to his party]** as a clausal complement, the condition (07a) is met.

Since, in a given context C, *"you will never guess how many people John invited to his party"* (=S_1) entails that *"John invited a lot of people to his party"* (=S_2), the condition (07b) is met too.[45]

The terminal string that corresponds to the surface structure of S_0 is the one in (10). Also, the terminal string that corresponds to the surface structure of S_2 without the substring corresponding to NP_1 is the one in (11). Since (10) is identical to (11), the condition (07c) is met.[46]

[45] Depending on the context, we may have other NPs than **a lot of people** acting as the direct object of **invited**, such as **very few people**, **an amazing number of people**, **a huge number of guests**, etc.

[46] We should keep in mind that the notion of Surface Structure used here (which goes back to the old days of generative grammar) is not equal to the notion of S-Structure of the Principle-&-Parameters approach, which recognizes different kinds of empty categories with different

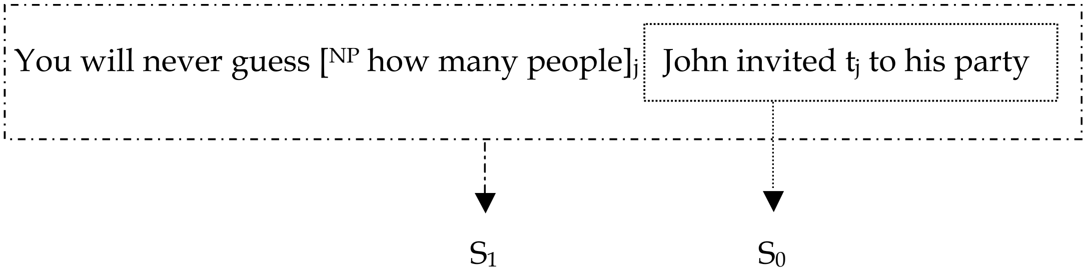
(10) John invited ∅ to his party.

(11) John invited ∅ to his party.

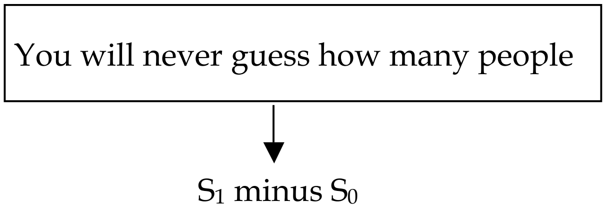
In addition, in the same context C in which the condition (07a) is met, *"you will never guess how many people John invited to his party"* (=S_1) has the force of an exclamation, which means that the condition (07d) is also met.

The terminal string that corresponds to S_1 without the substring corresponding to S_0 is the one in (13).

(12)


You will never guess [NP how many people]j John invited tj to his party
S1
S0

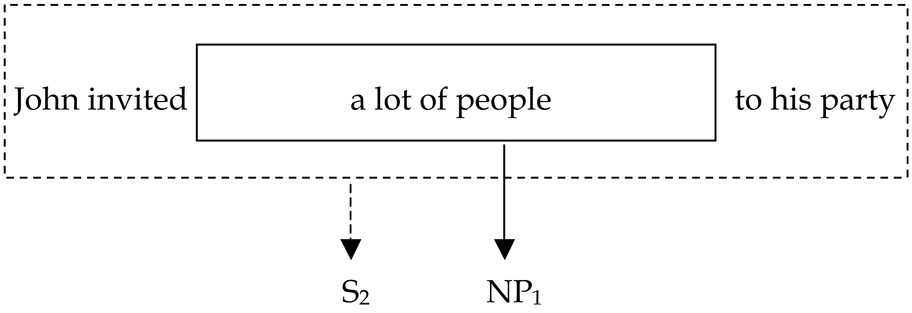
(13)



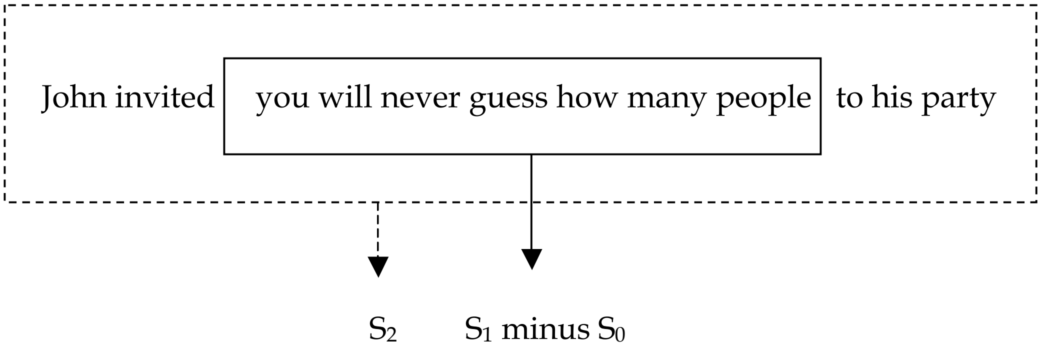
syntactic and semantic behaviors. The grammatical mechanism discussed here operates on phrase-markers, defined as sets of strings of symbols, and recognizes that the non-terminal string John⁀invited⁀NP⁀to⁀his⁀party is a member of both the phrase-marker of S_0 and the phrase marker of S_2. It also recognizes that the terminal string John⁀invited⁀to⁀his⁀party is a member of both the phrase-marker of S_0 and the phrase marker of S_2. Somehow, this allows the NP symbol in the string John⁀invited⁀NP⁀to⁀his⁀party of S_2 to be replaced with a reduced version of S1 which does not contain S_0 as a substring. The gaps indicated by ∅ in the notation in (10) and (11) are there for expository reasons only. They have no theoretical status, and basically encode the fact that there is a string (namely: John⁀invited⁀NP⁀to⁀his⁀party) in which an NP occupies the slot here marked as ∅.

Given that all four conditions are met, then, by (07e), the whole chunk in (13) (=S_1 minus S_0) may be copied from another derivation into derivation D, replacing NP_1 inside S_2 in the context C. This generalized transformation gets us from (14) to (15), yielding the syntactic amalgam in (01), repeated below as (16).

(14) **i^th^ phrase marker of derivation D**



(15) **i+1^th^ phrase marker of derivation D**



(16) John invited you'll never guess how many people to his party.

III.2.2. The Inner-Workings of Amalgamation: Sluicing, Cross-Derivational Adjunction & NP Ellipsis

So far, we have been talking about the recognition of certain 'incomplete' strings that are put together via generalized transformation. It is clear that they are not kernel sentences in the sense of Chomsky (1975). So, how do we get those chunks of sentence? Also, how come a string that 'is a' sentence can replace another string that 'is a' noun phrase? Certainly, this kind of rule is conceivable under the classic transformational approach underlying Lakoff's (1974) analysis. However, we should be careful and skeptical about it. This kind of formalism has been abandoned long ago precisely because it makes the system too unrestricted.[47]

This does not mean that we should drop Lakoff's (1974) proposal from serious consideration. Lakoff (1974) seems to have been aware of these issues already.[48] Along the paper, he supports a particular technical implementation of syntactic amalgamation given by William Cantrall, in which the overwriting of strings (i.e. replacement of NP_1 with "S_1 minus S_0") is factored out into three independent syntactic processes: (i) sluicing, (ii) adjunction, and (iii) NP ellipsis.

[47] Moreover, the postulation of amalgamation rules like (07) faces serious problems with regards to learnability. That is, how do children acquire such rules without negative evidence? Is the input robust enough, with plenty of examples of syntactic amalgam? It doesn't seem so. The way out of the problem would be to assume that all amalgamation rules are fully innate. Besides, syntactic amalgams don't have exactly the same structure in all languages (cf. **§II.8**). That could be treated in terms of parametrization, of course. But that would require postulating **construction-specific** (parametrized) constraints.

[48] However, Lakoff (1974) does not say anything about learnability.

Since this account is much closer to any given Principle & Parameters account, we should examine it before we move on to any other technical solution.

> Bill Cantrall has suggested what may be a more plausible derivation for the Andrews sentences. He suggests that (1′) may be an intermediate stage in the derivation of (1).
>
> (1) John invited you will never guess how many people to his party.
>
> (1′) John invited a surprising number of people – you will never guess how many (people) – to his party.
>
> First the sentence remnant "you will never guess how many people" is inserted under pretty much the same conditions as those given in (06)[49], with perhaps the additional proviso that the constituent in S$_2$ that corresponds to the questioned constituent in S$_1$ is modified by the adjective "surprising" or "unexpected" or the equivalent. (1) would then be derived from the structure underlying (1′) by the deletion of "a surprising number of people". Cantrall's suggestion amounts to breaking up the substitution rule of (06) into two rules – an insertion rule and a deletion rule. this has the advantage of being able to account for constructions like (1′).
>
> Lakoff (1974: 323-324)

So, according to this line of reasoning, the generation of syntactic amalgams involves the following steps.

In the first stage, we have two separate sentences like (17) and (18).

(17) [^{S} John invited [^{NP} a surprising number of people] to his party]

(18) [^{S} you will never guess [^{NP} how many people]$_1$ John invited t$_1$ to his party]

[49] In Lakoff's (1974) paper, the rule (07) is numbered (12). In this quotation, (07) refers to the rule given in the previous subsection of this paper.

Then, (18) undergoes sluicing (whatever that process ultimately is)[50], yielding (19).

(19) [^{S} you will never guess [^{NP} how many people]$_1$ ~~John invited t$_1$ to his party~~]

Then, we insert (19) inside (17), as an adjunct to **[^{NP} a surprising number of people]**, via generalized transformation, yielding (20).

(20) [^{S} John invited [^{NP} [^{NP} a surprising number of people] [^{S} you will never guess [^{NP} how many people]$_1$ ~~John invited t$_1$ to his party~~]] to his party]

Finally, the NP that hosts the adjunct gets deleted, yielding (21).[51]/[52]

(21) [^{S} John invited [^{NP} [^{NP} ~~a surprising number of people~~] [^{S} you will never guess [^{NP} how many people]$_1$ ~~John invited t$_1$ to his party~~]] to his party]

[50] See Ross (1969) and Merchant (2001: chapter 2) on the matter.

[51] Tsubomoto & Whitman (2000) also postulate a further LF-movement internal to the clause that is adjoined to the elliptical DP, as in (i).

(i) [^{DP} [^{DP} *e*]$_1$ [^{CP} [^{CP} [^{DP}how many people]$_1$ ~~John invited~~ t$_1$ ~~to his party~~]]$_2$ [^{IP} you'll never guess t$_2$]]

[52] In this exposition, sluicing precedes adjunction, which precedes NP-deletion. However, it is not obvious from this example whether this is the actual order of application of the rules, or even whether there is any (intrinsic or derived) order of application to those rules. In principle, it could be that the order is arbitrary, or even that all those operations are parallel (which is perhaps the null hypothesis in a generalized-transformational approach).

III.2.3. Problems

Although it captures the essence of the paradoxical constituency effect, Lakoff's (1974) account of syntactic amalgams leaves many questions without answers.

First of all, notice that, in Lakoff's (1974) analysis, all amalgamation rules are sensitive to specific syntactic constructions in a direct and explicit way (cf. 07); therefore this approach assigns a theoretical status to descriptive notions such as 'indirect question', 'cleft sentence', 'relative clause', 'reason clause', etc (cf. the *Appendix* at the end of this chapter). Although it is possible, in principle, to 'lexicalize' all of this, it is better if we could derive the amalgamation effects from the interaction of other parameters that we already need to assume on independent grounds.

Also, when we submit the analysis of WH-amalgams above to close scrutiny, we detect that, aside from the adjunction operation, there are two deletion rules involved: NP ellipsis and sluicing; and both seem to be obligatory, as shown in the paradigm in (22).

(22) a: ✓ **[+ NP Ellipsis, + Sluicing]**

John invited [^{NP} [^{NP} ~~a surprising number of people~~] [^{S} you'll never guess how many people ~~John invited to his party~~]] to his party.

b: * **[– NP Ellipsis, + Sluicing]**[53]

John invited [^{NP} [^{NP} a surprising number of people] [^{S} you'll never guess how many people ~~John invited to his party~~]] to his party.

c: * **[+ NP Ellipsis, – Sluicing]**

John invited [^{NP} [^{NP} ~~a surprising number of people~~] [^{S} you'll never guess how many people John invited to his party]] to his party.

d: * **[– NP Ellipsis, – Sluicing]**

John invited [^{NP} [^{NP} a surprising number of people] [^{S} you'll never guess how many people John invited to his party]] to his party.

The question is, then: why so? The basic intuition behind William Cantrall's suggestion is to eliminate the construction specific character of amalgamation as much as possible, and derive its effects from the interaction of other independent grammatical mechanisms.[54] However, nothing in this analysis explains why both sluicing and NP ellipsis are obligatory and must apply *in tandem*, as shown in (22).[55] If we aim to eliminate the construction-specific

[53] If we assume that the sluiced sentence adjoins to the left of the indefinite NP, the relevant example, whose ungrammaticality needs to be explained would be (i):

(i) * John invited you will never guess how many people a lot of people to his party.

[54] Actually, the idea of eliminating the theoretical status of specific syntactic constructions is not explicitly mentioned in Lakoff's (1974) presentation of William Cantrall's suggestion. But, as far as I can see, there is a 'principles-&-parameters flavor' inherent to that proposal.

[55] The same criticism applies to Romance, illustrated below with examples from Portuguese:

(i) João convidou ~~muita gente~~ você nunca vai adivinhar quantas pessoas ~~João convidou~~
John invited ~~*many people*~~ *you never will guess how-many people* ~~*John invited*~~
~~pra festa dele~~ pra festa dele.
~~*to+the party of+him*~~ *to+the party of+him*

(ii) * João convidou muita gente você nunca vai adivinhar quantas pessoas ~~João convidou~~
John invited many people you never will guess how-many people ~~*John invited*~~

character of amalgamation, and derive its effects from the interaction of other independent grammatical mechanisms, we must not have two allegedly distinct operations being parasitic on one another just by stipulation.

As far as NP-ellipsis is concerned, my criticism may not apply so obviously. In fact, Lakoff (1974) claims that the technical implementation suggested by William Cantrall has the advantage of also accounting for sentences like (23). Apparently, the basic difference between (22b) and (23) would be whether or not NP ellipsis applies.

(23) John invited a surprising number of people — you will never guess how many —to his party.

Notice, however, that the sluicing that takes place in this construction without NP-ellipsis goes a little bit further than what we see in (22b). As a matter of fact, the example in (22d), where the WH-phrase surfaces as **how many people**, is not acceptable, as opposed to the acceptable example in (23), where the WH-phrase surfaces as **how many**.

~~pra festa dele~~ pra festa dele.
~~to+the party of+him~~ to+the party of+him

(iii) * João convidou ~~muita gente~~ você nunca vai adivinhar quantas pessoas João convidou
John invited ~~many people~~ you never will guess how-many people John invited
pra festa dele pra festa dele.
to+the party of+him to+the party of+him

(iv) * João convidou muita gente você nunca vai adivinhar quantas pessoas João convidou
John invited many people you never will guess how-many people John invited
pra festa dele pra festa dele.
to+the party of+him to+the party of+him

Thus, the apparent advantage of this formalism (and its alleged unification power) does not resist closer scrutiny, as it is clear that there are two distinct types of sluicing, one for each construction. The relevance of this comparison lies on the fact that the very same type of sluicing involved in (22b) is also involved in the acceptable example in (22a).

Crucially, the type of sluicing found in (23) – where deletion/ellipsis also affects the head of the NP – is the same one independently found outside syntactic amalgams, as shown in (24).

(24) a: John invited a surprising number of people to his party. You will never guess how many ~~people John invited to his party~~.

b: John invited a surprising number of people —*You will never guess how many ~~people John invited to his party~~* —to his party.

If that same type of sluicing takes place in a genuine syntactic amalgam, the resulting structure is not acceptable, as shown in (25).

(25) * John invited [NP [NP ~~a surprising number of people~~] [S you will never guess how many ~~people John invited to his party~~]] to his party.

This contrast can be taken as evidence that (23) is a *bona fide* case of parenthetical construction, rather than a syntactic amalgam (and its prosodic structure seems to corroborate that). Not only does (23) not exhibit NP-ellipsis, but also what seems to be its 'invasive clause' exhibits a standard form of sluicing. Moreover, in such constructions, the parenthetical does not necessarily look like an 'incomplete' sentence at the surface, as shown in (26) and (27).

(26) a: John invited a surprising number of people to his party. You will never guess how many guests there are.

b: John invited a surprising number of people *(You will never guess how many guests there are)* to his party.

(27) a: John invited an amazing number of people to 40th birthday party. Apparently, there are two thousand guests in total.

b: John invited an amazing number of people *(apparently, there are two thousand guests in total)* to his 40th birthday party.

One may argue that both (22a) and (23) are genuine syntactic amalgams, and that their distinct types of sluicing follow from recoverability of deletion, as a consequence of the fact that (22a) exhibits NP-ellipsis and (23) does not.[56] In (22a), given that the direct object of the matrix clause — i.e. [^{NP} ~~a surprising number of people~~] — undergoes ellipsis, the token of **people** in the sluiced sentence is not recoverable, therefore it must not be deleted by sluicing, as illustrated in (28). Conversely, in (23), the matrix direct object — i.e. [^{NP} a

[56] I am thankful to Klaus Abels for discussion.

surprising number of people] — does not undergo ellipsis, which makes the token of **<u>people</u>** in the sluiced sentence recoverable, therefore it should be deleted by the sluicing mechanism, as illustrated in (29).

(28) a: John invited [^{NP} [^{NP} ~~a surprising number of people~~] [^{S} you'll never guess how many people ~~John invited to his party~~]] to his party.
b: * John invited [^{NP} [^{NP} ~~a surprising number of people~~] [^{S} you'll never guess how many ~~people John invited to his party~~]] to his party.

(29) a: * John invited [^{NP} [^{NP} a surprising number of people] [^{S} you'll never guess how many people ~~John invited to his party~~]] to his party.
b: John invited [^{NP} [^{NP} a surprising number of people] [^{S} you'll never guess how many ~~people John invited to his party~~]] to his party.

The apparent advantage of this analysis is that the two different types of sluicing seem to be derivable from the structural context and independent assumptions about recoverability of deletion. Notice, however, that this goal is not really being achieved, as it is still necessary to rely on a construction-specific mechanics that cannot be extended to account for ordinary cases of sluicing outside amalgams. This is so because, in order to derive the two different types of sluicing from NP-ellipsis, it is necessary to stipulate that this very operation of NP-ellipsis is a construction-specific mechanism, whose structural description is defined in terms of syntactic amalgams. That would be the only way to predict a grammaticality contrast between (30a) and (30b).

(30) a: John invited [^{NP} a surprising number of people] to his party.

You will never guess how many ~~people John invited to his party~~.

b: * John invited [^{NP} ~~a surprising number of people~~] to his party.

You will never guess how many people ~~John invited to his party~~.

If amalgamation really involved sluicing inside the 'invasive clause', and if the idiosyncratic 'reach' of that mandatory sluicing operation really followed from the optionality of NP-ellipsis, *modulo* recoverability of deletion, then, *ceteris paribus*, we would expect both examples in (30) to be acceptable. In (30a), NP-ellipsis does not apply, and sluicing must delete **people** in the second sentence of the mini-text. In (30b), NP-ellipsis applies, and sluicing does not delete **people** in the second sentence of the mini-text.

The fact, however, is that (30b) is not a legitimate structure. Therefore, this analysis based on recoverability of deletion lacks explanatory adequacy, as it assigns theoretical status to 'amalgamation constructions', which the NP-ellipsis rule would be sensitive to. Therefore, there is something else going on, which escapes from Lakoff's (1974) analysis. It cannot be just that NP-ellipsis is optional. The sluicing that takes place in the adjoined clause is somehow parasitic on NP-ellipsis.

Aside from the obligatoriness/optionality of NP-ellipsis and the details about how much of the target string of words is affected by sluicing, there is the issue of sluicing being obligatory inside syntactic amalgams, but optional otherwise, as shown in (31) and (32). Again, in order for the analysis under discussion to account for that, a construction-specific mechanism of sluicing would be required for syntactic amalgams, leading to explanatory inadequacy.

(31) *Obligatory Sluicing Inside Syntactic Amalgams*

a: John invited [NP [NP ~~a surprising number of people~~] [S you'll never guess how many people ~~John invited to his party~~]] to his party.

b: * John invited [NP [NP ~~a surprising number of people~~] [S you'll never guess how many people John invited to his party]] to his party.

(32) *Optional Sluicing Outside Syntactic Amalgams*

a: John invited [NP a surprising number of people] to his party. You will never guess how many ~~people John invited to his party~~.

b: John invited [NP ~~a surprising number of people~~] to his party. You will never guess how many people John invited to his party.

An alternative would be to assume that the sluiced sentence corresponding to the 'invasive clause' is not really an adjunct to an elliptical indefinite NP, but rather some sort of complex pre-nominal determiner or modifier to an overt head noun instantiated by **people**, as in (33).[57]

(33) John invited [^{NP} [^{S} you will never guess how many ~~people John invited to his party~~] [$^{N'}$ people]] to his party.

That way, the there would be nothing special about the sluicing that takes place inside syntactic amalgams, and no *ad hoc* NP-ellipsis rule would be need to be stipulated for syntactic amalgams.[58]

One problem with this analysis is that it cannot be extended to cases like (34), in which the WH-phrase in the supposedly sluiced sentence is a bare WH element rather than a complex phrase decomposable into a WH-determiner and a noun.

(34) John invited you'll never guess who to his party.

Ceteris paribus, this complex-determiner hypothesis wrongly predicts the generation of something like (35) instead of (34).

[57] I am thankful to Colin Phillips and Jonathan Bobaljik for pointing out this possibility to me.

[58] From that perspective, the possibility of having both (16) and (23) — repeated below as (i) and (ii) respectively — would correlate with the possibility of the sluiced sentence to behave either as a pre-pronominal complex determiner/modifier to N (cf. (i)) or as an adjunct to NP (cf. (ii)).

(i) John invited you'll never guess how many people to his party.

(ii) John invited a surprising number of people — you'll never guess how many — to his party.

(35) * John invited [^{NP} [^{S} you will never guess who ~~John invited to his party~~] [$^{N'}$person]] to his party.

A way out of this problem would be to resort to an *ad hoc* mechanism of ellipsis, as in (36). Again, this analysis is explanatorily inadequate, as no unified account of amalgams is achieved.

(36) John invited [^{NP} [^{S} you will never guess who ~~John invited to his party~~] [$^{N'}$ ~~someone~~]] to his party.

Moreover, this complex-determiner hypothesis also faces the problem of having to stipulate the obligatoriness of sluicing inside syntactic amalgams but not otherwise (cf. (31) and (32) above) in order to account for the constrast in (37).

(37) a: John invited [^{NP} [^{S} you will never guess how many ~~people John invited to his party~~] [$^{N'}$ people]] to his party.

b: * John invited [^{NP} [^{S} you will never guess how many people John invited to his party] [$^{N'}$ people]] to his party.

Another worry that arises from Lakoff's (1974) analysis is the following. It is claimed that amalgams involve a kernel sentence like (38a), which becomes something like (38b) after NP ellipsis. Eventually, after the adjunction of the apparently-sluiced clause, (38c) obtains.

(38) a: John invited **[^{NP} a surprising number of people]** to his party.

b: John invited **[^{NP} Δ]** to his party.

c: John invited [^{NP} **[^{NP} Δ]$_j$** [^{S} you will never guess **[^{NP}** how many people**]$_j$** ~~John invited to his party~~]] to his party.

The question, then, is: why is this the gap **[^{NP} Δ]** interpreted as **[^{NP} how many people]**? That is, why is the object of **invited** in (01) interpreted as *"a number of people **n**, such that you will never guess **n**"*, as indicated by the indices in (38)? (cf. Tsubomoto & Whitman 2000). There is nothing in Lakoff's (1974) analysis that accounts for this fact.[59]

Now, moving on to the empirical generalizations presented in chapter **II**, consider the paradigm in (61), which illustrates that invasive clauses cannot target non-ECM subject positions. The corresponding structures are given in (62).

(61) a: Tom said that Amy is dating *I forgot **who***.

b: * Tom said that *I forgot **who*** is dating Amy**.**

c: * Tom said that *I forgot **who*** was kissed by Amy at the party**.**

[59] Tsubomoto & Whitman's solution is formalized under an indexation-through-predication approach, with feature-percolation mechanisms. Although it makes the right predictions, such analysis is problematic from a minimalist perspective, given its reification of indices.

d: The conductor of the orchestra wants *you'll never guess* ***which musician*** to be in charge of the rehearsal while he will be out of town.

(62) a: [^{S} Tom said [$^{S'}$ that [^{S} Amy is dating [^{NP} [^{NP} *e*] [^{S} I forgot ***who***$_1$ ~~Amy is dating t$_1$~~]]]]]]

b: * [^{S} Tom said [$^{S'}$ that [^{S} [^{NP} [^{NP} *e*] [^{S} I forgot ***who***$_1$ ~~t$_1$ is dating Amy~~]]] is dating Amy]]]

c: * [^{S} Tom said [$^{S'}$ that [^{S} [^{NP} [^{NP} *e*] [^{S} I forgot ***who***$_1$ ~~t$_1$ was kissed by Amy at the party~~]]] was kissed by Amy at the party]]]

d: [^{S} [^{NP} the conductor of the orchestra] [^{VP} wants [^{NP} [^{NP} *e*] [^{S} you'll never guess [***which musician***]$_1$ ~~the conductor of the orchestra wants t$_1$ to be in charge of the rehearsal while he will be out of town~~]]] to be in charge of the rehearsal while he will be out of town]]

Without further stipulation, the sluicing-based analysis wrongly predicts that no such constraint on amalgamation can exist, as there is nothing in the corresponding structures for the examples in (61) which such constraint could piggyback on, unless one simply stipulates that the elliptical NPs to which sluiced sentences adjoin cannot occupy non-ECM subject positions. That, however, would be just a restatement of the facts.

Consider, now, the cross-linguistic difference between English and Romance presented in chapter **II**, with regards to cases where the 'clause invasion' affects the object of a preposition.

In English, the preposition must appear before the material that is supposedly adjoined to the NP which is the complement of that preposition, as shown in (63).

(63) a: John invited 300 people to you can imagine what kind of party.
b: ?* John invited 300 people you can imagine to what kind of party.

This has a straightforward account under the sluicing-based approach, as shown in (64).[60]

(64) John invited 300 people [^{PP} **to** [^{NP} [^{NP} *e*] [^{S} you can imagine **[what kind of party]**$_1$ ~~John invited 300 people to t$_1$~~]]]]

[60] Also straightforward would be the treatment of the unacceptability of structures like (i), mentioned at the end of **§II.8** as an apparent mystery for the sluicing approach to amalgamation, since the special type of sluicing involved in (i) — where the preposition is pronounced at the end of the string — is actually found in its non-amalgamated analogue in (ii).
(i) * John danced *I don't remember who with* at the party.
(ii) John danced at the party. But I don't remember **who**$_1$ ~~John danced~~ with ~~t$_1$ at the party~~**.**
Notice that, outside syntactic amalgams, this special type of sluicing obtains only when the WH-phase of the sluiced sentence corresponds to an adjunct in the previous sentence, as in (iii), where **danced** is used intransitively. In (iv), **danced** is used transitively, selecting **with someone** as its indirect object, and such structural context only licenses ordinary sluicing.
(iii) a: John danced at the party. But I don't remember who with**.**
b: * John danced at the party. But I don't remember who**.**
(iv) a: * John danced with someone at the party. But I don't remember who with**.**
b: John danced with someone at the party. But I don't remember who**.**
Extending this logic to syntactic amalgams, we would expect that an invasive clause that undergoes the special kind of sluicing — such as **I don't remember who with** — would require that the verb **danced** in the invaded clause be used intransitively. is being used intransitively. That being the case, there would be no elliptical NP in the indirect object position to begin with. Consequently, there would be no appropriate host where the sluiced invasive clause could adjoin to. Therefore, structures like (i) are correctly predicted to be ungrammatical.

In Romance, the opposite pattern obtains. The preposition must appear after the material that is supposedly adjoined to the NP which is the complement of that preposition, as shown in (65).

(65) a: * João convidou 300 pessoas pra você pode imaginar que tipo de festa.
John invited 300 persons to you can imagine what kind of party
b: João convidou 300 pessoas você pode imaginar pra que tipo de festa.
John invited 300 persons you can imagine to what kind of party

If we maintain the view that the substring **you can imagine what kind of party** (and its Romance equivalent) is a sluiced sentence that adjoins to an elliptical NP, we are forced to assume that, in the matrix clause, the preposition that takes that elliptical NP as its complement somehow must undergo ellipsis in Romance but not in English, as in (66).[61] Alternatively, we may say that, for some reason, the elliptical argument which the sluiced clause adjoins to is an NP in English, but a PP in Romance, as in (67).[62] Either way, an extra parametric difference is stipulated without independent evidence.

[61] This also applies to the complex-determiner analysis sketched in (33), as shown in (i) [= (01)]
(i) João convidou 300 pessoas [^{PP} ~~pra~~ [^{NP} [$^{\alpha}$ você pode imaginar pra que] [$^{N'}$ tipo de festa]]]
John invited 300 persons ~~to~~ you can imagine to what kind of party

[62] Notice that the alternative analysis in (67) has to be formalized in such a way that NPs can also be the target of ellipsis and adjunction whenever there is no PP involved, or else cases like (i) would not be accounted for. This undesirably complicates the analysis even further.
(i) John invited [^{NP} [^{NP} *e* [^{S} you'll never guess **who$_1$** ~~John invited t$_1$ to his party~~]] to his party.

(66) João convidou 300 pessoas [^{PP} ~~pra~~ [^{NP} [^{NP} *e*] [^{CP} você pode imaginar
John invited 300 persons ~~to~~ you can imagine
pra que tipo de festa ~~João convidou 300 pessoas~~]]]
to what kind of party ~~John invited 300 persons~~

(67) João convidou 300 pessoas [^{PP} [^{PP} *e*] [^{CP} você pode imaginar
John invited 300 persons you can imagine
pra que tipo de festa ~~João convidou 300 pessoas~~]]
to what kind of party ~~John invited 300 persons~~

The analysis in (66) deserves further discussion. As suggested above for (22a) and (23), one could, in principle, hypothesize that the existence of these two distinct forms of sluicing follows from recoverability of deletion coupled with standard assumptions about the parametric difference between the two languages with regards to pied-piping/preposition-stranding.

In Romance (cf. 66), the preposition inside the sluiced clause escapes is not affected by ellipsis in the sluicing process by virtue of it being pied-piped to spec/CP along with the WH-phrase. That way, the preposition in the invaded clause would be deleted under identity with the preposition inside the sluiced clause. In English (cf. 64), on the other hand, the preposition of the invasive clause is affected by ellipsis in the sluicing process by virtue of it being stranded inside the IP. That way, the preposition in the invaded clause cannot be deleted because it is unrecoverable.[63]

The problem with this analysis is that it lacks independent motivation. In standard cases of sluicing in (Brazilian) Portuguese, whereas the preposition must be pronounced in the sentence preceding the sluiced clause, it must be absent (at least at PF) in the sluiced clause, contrary to Merchant's (2001: 91-107) generalization about strong pied-piping languages, as shown in (68).[64]

(68) a: Bob deu dinheiro pra alguém. Mas eu não sei quem.
Bob gave money to someone. But I not know who.
b: * Bob deu dinheiro pra alguém. Mas eu não sei pra quem.
Bob gave money to someone. But I not know to who.

In (69), we see that deleting the preposition in the non-sluiced sentence and pronouncing the preposition in the sluiced sentence is not a legitimate alternative.

(69) a: * Bob deu dinheiro ~~pra~~ alguém. Mas eu nao sei pra quem.
Bob gave money ~~to~~ someone. But I not know to who
b: * Bob deu dinheiro ~~pra alguém~~. Mas eu nao sei pra quem.
Bob gave money ~~to~~ ~~someone~~. But I not know to who.

[63] This possibility was pointed out to me by Klaus Abels.

[64] *"Form-identity generalization II: Preposition-stranding. A language L will allow preposition stranding under sluicing iff L allows preposition stranding under regular wh-movement"* (Merchant 2001: 91)

Presumably, from this perspective, the idiosyncrasy of Brazilian Portuguese that causes Merchant's generalization to break down is that, for some unknown reason, the structure that serves as the input to the ellipsis operation involved in sluicing is not *quasi*-isomorphic to the overt clause. Rather, the input structure would consist of a copular sentence, like (70), which undergoes ellipsis and turns into the sluiced string in (71).

(70) Bob deu dinheiro pra [alguém]$_1$. Mas eu não sei quem é [essa pessoa]$_1$.
Bob gave money to someone. But I not know who is that person is.
'Bob gave money to a certain person. But I don't know who such person is'

(71) Bob deu dinheiro pra [alguém]$_1$. Mas eu não sei quem ~~é [essa pessoa]$_1$~~.
Bob gave money to someone. But I not know who is that person is.

On the other hand, invasive clauses within syntactic amalgams exhibit a structure that conforms to Merchant's generalization, as shown in (72). Notice that the preposition is pronounced only inside the parenthetic-like string, not in the core structure.

(72) a: Bob deu dinheiro pra *eu não sei quem.*
Bob gave money to I not know who.
b: * Bob deu dinheiro *eu não sei pra quem.*
Bob gave money I not know to who.

In a nutshell, the cross-linguistic variation with regards to the relative order of prepositions and invasive clauses — which correlates with the pied-piping/preposition-stranding distinction — poses a serious problem to the sluicing-based analysis of amalgamation.

Now, let us consider how the sluicing-based approach handles the facts about co-reference among NPs/DPs across different parts of a syntactic amalgam discussed in **§II.9**.

It is possible for an R-expression in the spine of the invaded clause to co-refer with a pronoun in the spine of the invasive clause, as shown in (73a). This co-reference pattern differs from what obtains in the corresponding hypotactic paraphrase in (73b), and mirrors what obtains in the corresponding paratactic paraphrase in (73c).

(73) a: **[Homer]**$_1$ drank **[he]**$_{1/2}$ doesn't even remember how many beers at the party.

b: **[He]**$_{*1/2}$ doesn't even remember how many beers **[Homer]**$_1$ drank at the party.

c: **[Homer]**$_1$ drank beers at the party. **[He]**$_{1/2}$ doesn't even remember how many.

If, on the other hand, the R-expression is in the spine of the invasive clause and the pronoun is in the spine of the invaded clause, co-reference is impossible, as in (74a). Again, the corresponding hypotactic paraphrase does not exhibit the same pattern (cf. 74b), whereas the corresponding paratactic paraphrase does (cf. 74c).

(74) a: **[He]**$_{*1/2}$ drank **[Homer]**$_{1}$ doesn't even remember how many beers at the party.

b: **[Homer]**$_{1}$ doesn't even remember how many beers **[he]**$_{1/2}$ drank at the party.

c: **[He]**$_{*1/2}$ drank beers at the party. **[Homer]**$_{1}$ doesn't even remember how many.

At first blush, the facts in (73) and (74) appear to receive a straightforward account under the sluicing-based approach to amalgamation. The structure of the two syntactic amalgams in (73a) and (73b) would be as in (75) and (76), respectively. Notice that the overt token of **Homer** is not c-commanded by **he** in (75), whereas in (76) it is. Thus, by Principle C, co-reference should be possible in (75) but not in (76).[65]

[65] Also straightforward are the cases where the pronoun is not in the spine of either the invasive or the invaded clause, but rather embedded inside a more complex NP/DP, as in (i) and (ii). Co-reference is legitimate in all possible combinations, which is compatible with Principle C, given that the pronoun does not c-command the R-expression in any of the examples.

(i) a: **[Homer]**$_{1}$ drank [^{NP} [^{NP} *e*] [^{S} I bet **[[his]**$_{1/2}$ **wife]** remembers how many beers ~~Homer drank at the party~~] at the party.

b: [^{S} I bet **[[his]**$_{1/2}$ **wife]** remembers [$^{S'}$ how many beers [^{S} **[Homer]**$_{1}$ drank at the party]]]

c: **[Homer]**$_{1}$ drank beers at the party. I bet **[[his]**$_{1/2}$ **wife]** remembers how many.

(ii) a: **[[His]**$_{1/2}$ **wife]** drank [^{NP} [^{NP} *e*] [^{S} I bet **[Homer]**$_{1}$ remembers how many beers ~~his wife drank at the party~~] at the party.

(75) [^{IP} **Homer** [^{VP} drank [^{NP} [^{NP} *e*] [^{S} **he** doesn't even remember [how many beers]$_1$ ~~Homer drank t$_1$ at the party~~]] at the party]]

(76) [^{IP} **He** [^{VP} drank [^{NP} [^{NP} *e*] [^{IP} **Homer** doesn't even remember [how many beers]$_1$ ~~he drank t$_1$ at the party~~]] at the party]]

In the hypotactic paraphrases, the opposite c-command relations obtain, as shown in (77) and (78). It follows, then, that Principle C would yield the opposite effects, as it does.[66]

(77) [^{CP} [^{IP} **He** doesn't even remember [^{CP} [how many beers]$_1$ [^{IP} **Homer** drank t$_1$ at the party]]]]

(78) [^{S} **Homer** doesn't even remember [$^{S'}$ [how many beers]$_1$ [^{S} **he** drank t$_1$ at the party]]]

However, nothing has been said so far about the NPs/DPs affected by sluicing inside the invaded clause. Let us consider (75) again, repeated below as (79).

b: [^{S} I bet **[Homer]$_1$** remembers [$^{S'}$ how many beers [^{S} **[[his]$_{1/2}$ wife]** drank at the party]]]

c: **[[His]$_{1/2}$ wife]** drank beers at the party. I bet **[Homer]$_1$** remembers how many.

[66] In both paratactic paraphrases ((73c) and (74c)), there is no c-command relation between the pronoun and the R-expression, which belong to two independent parallel sentences. Co-reference is possible in (73c) but not in (74c). Needless to say, this contrast does not follow from Principle C. Presumably, it follows from some post-LF condition on deixis at the pragmatic level.

(79) [^{S} **Homer** [^{VP} drank [^{NP} [^{NP} *e*] [^{S} **he** doesn’t even remember [how many beers]$_1$ ~~Homer drank t$_1$ at the party~~]] at the party]]

There are actually two tokens of **Homer** in the structure. One is overt, and occupies a position in the spine of the invaded clause, outside the c-command path of **he** (therefore no Principle C violation arises in case of co-reference). The other one is unpronounced due to sluicing, and is inside the invaded clause, so that it is c-commanded by the overt token of **Homer**. The fact that these two tokens of **Homer** co-refer, despite one c-commanding the other, is problematic, given that such co-reference constitutes a violation of Principle C. Alternatively, one may consider the hypothesis that the sluiced sentence does not contain a token of **Homer**. Rather, there would be a pronoun in that position, as in (80).

(80) [^{S} **Homer**$_1$ [^{VP} drank [^{NP} [^{NP} *e*] [^{S} **he**$_{1/2}$ doesn’t even remember [how many beers]$_1$ ~~**he**$_{1/*2}$ drank t$_1$ at the party~~]] at the party]]

The fact that the deeply embedded and unpronounced token of he co-refers with **Homer** does not constitute a violation of any binding principle. The problem, however, is that, unlike the higher and overt token of **he**, which may or may not co-refer with **Homer**, the lower and unpronounced token of **he** must co-refer with **Homer**. Such mandatory co-reference does not follow from anything in Binding Theory, and thus constitutes a construction-specific property of amalgams that remains unexplained under the sluicing-based approach.

Another potential problem for the sluicing-based approach relates to the facts in (81) and (82) below.

In (81a), there is an epithet NP/DP (i.e. **the idiot**) in the spine of the invaded clause, and a proper name (i.e. **Homer**) in the spine of the invasive clause. Co-reference between the two is impossible. The same pattern obtains in both the hypotactic and the paratactic paraphrases, as shown in (81b) and (81c).

(81) a: **[The idiot]**$_{*1/2}$ drank **[Homer]**$_{1}$ doesn't even remember how many beers at the party.

b: **[Homer]**$_{1}$ doesn't even remember how many beers **[the idiot]**$*_{1/2}$ drank at the party.

c: **[The idiot]**$_{*1/2}$ drank beers at the party. **[Homer]**$_{1}$ doesn't even remember how many.

In (82a), on the other hand, it is **Homer** that is in the spine of the invaded clause, whereas **the idiot** is in the spine of the invasive clause. It is possible for them to co-refer, unlike what happens in both the hypotactic and the paratactic paraphrases, as shown in (82b) and (82c).

(82) a: **[Homer]**$_{1}$ drank **[the idiot]**$_{1/2}$ doesn't even remember how many beers at the party.

b: **[The idiot]**$_{*1/2}$ doesn't even remember how many beers **[Homer]**$_{1}$ drank at the party.

c: **[Homer]**$_{1}$ drank beers at the party. **[The idiot]**$_{1/2}$ doesn't even remember how many.

The pattern in (81a) has a straightforward explanation under the sluicing-based approach to amalgamation. The corresponding structure would be as in (83), where the epithet **the idiot** is c-commanded by the proper name **Homer**. Under the standard assumption that epithets are subject to *Principle C* (cf. Lasnik 1976, 1991), co-reference between **Homer** and **the idiot** is correctly predicted to be impossible.

(83) [^{S} **Homer** [^{VP} drank [^{NP} [^{NP} *e*] [^{S} **[the idiot]** doesn't even remember [how many beers]$_{1}$ ~~Homer drank t$_{1}$ at the party~~]] at the party]]

The same logic would apply to the hypotactic paraphrase, whose structure would be as sketched in (84). There, it is also the case **the idiot** c-commands **Homer**.

(84) [^{S} **Homer** doesn't even remember [$^{S'}$ [how many beers]$_{1}$ [^{S} **[the idiot]** drank t$_{1}$ at the party]]]

The pattern in (82a), however, poses a problem to the sluicing-based approach to amalgamation. The corresponding structure would be as in (85), where **Homer** is c-commanded by **the idiot**. By the same logic applied to (81a), co-reference between these two R-expressions should be impossible, *modulo* Principle C. But that is not the case.

(85) [^{S} **[the idiot]** [^{VP} drank [^{NP} [^{NP} *e*] [^{S} **Homer** doesn't even remember [how many beers]$_1$ ~~[the idiot] drank t_1 at the party~~]] at the party]]

Notice that, in the corresponding hypotactic paraphrase, co-reference between **Homer** and **the idiot** is impossible, as expected under standard assumptions about Principle C and c-command.

(86) [^{S} **Homer** doesn't even remember [$^{S'}$ [how many beers]$_1$ [^{S} **[the idiot]** drank t_1 at the party]]]

Finally, the sluicing-based analysis reveals itself problematic in face of the fact that invasive clauses systematically behave like matrix clauses, to the extent that they may exhibit certain grammatical patterns that are licensed only in matrix clauses, like auxiliary-inversion/do-support and imperative mood, as shown in (87) and (88).

(87) [Bob told me that Amy danced with [*do you know who?*] at the party]

(88) [Bob told me that Amy danced with [*guess who!*] at the party]

The problem with this is that, under the sluicing-based approach, those very clauses exhibiting auxiliary-inversion/do-support and imperative mood are analyzed as embedded clauses, as shown in (89) and (90).

(89) [^{S} Bob told me that Amy danced with [^{NP} *e* [^{S} do you know [$^{S'}$ **who$_1$** [^{S} ~~Amy danced with **t$_1$** at the party~~]]]] at the party]

(90) [^{S} Bob told me that Amy danced with [^{NP} *e* [^{S} guess [$^{S'}$ **who$_1$** [^{S} ~~Amy danced with **t$_1$** at the party~~]]]] at the party]

It is rather mysterious, then, as to why those alleged embedded clauses of syntactic amalgams can behave like matrix clauses, but other embedded clauses cannot.

Another instance of the same problem can be observed in the distribution of empty categories in Brazilian Portuguese. As shown in **§II.10**, in Brazilian Portuguese, gaps in the position of a (3rd person) subject are licensed only in certain specific kinds of embedded clauses, as in (91), but never in matrix clauses, as in (92).[67]

(91) a: Maria$_1$ não se lembra quantos homens ela$_{1/2}$ beijou na festa.
Mary$_1$ not REFL remember how+many men she$_{1/2}$ kissed at+the party.
'Mary$_1$ doesn't remember how many men she$_{1/2}$ kissed at the party'

b: Maria$_1$ não se lembra quantos homens *e*$_{1/*2}$ beijou na festa.
Mary$_1$ not REFL remember how+many men ∅$_{1/*2}$ kissed at+the party.
'Mary$_1$ doesn't remember how many men she$_{1/*2}$ kissed at the party'

(92) a: Maria$_1$ beijou muitos homens na festa. Ela$_1$ nem se lembra quantos.
Mary kissed many men at+the party. She not+even REFL remember how+many
'Mary kissed many men at the party. She doesn't even remember how many'

b: * Maria$_1$ beijou muitos homens na festa. *e*$_1$ nem se lembra quantos.
Mary kissed many men at+the party. ∅ not+even REFL remember how+many
'Mary kissed many men at the party. She doesn't even remember how many'

[67] For an exhaustive and detailed presentation of this empirical generalization, and for an explanation on why it holds, I refer the reader to Rodrigues (2002, 2004), who analyses those gaps as traces (≡ deleted copies) of movement, whose antecedent is the subject of the matrix clause.

Crucially, the invasive clauses of syntactic amalgams behave exactly as matrix clauses, as no gap in subject position is possible there, as shown in (93).

(93) a: Maria$_1$ beijou ela$_1$ nem se lembra quantos homens na festa.
Mary kissed she not+even REFL remember how+many men at+the party
'Mary kissed she doesn't even remember how many men at the party'

b: * Maria$_1$ beijou *e*$_1$ nem se lembra quantos homens na festa.
Mary kissed ∅ not+even REFL remember how+many men at+the party
'Mary kissed she doesn't even remember how many men at the party'

If invasive clauses are taken to be embedded clauses adjoined to an (elliptical) NP/DP, as in (94), then there would be no reason for subject gaps not to be licensed in those domains.

(94) a: [^{S} Maria beijou [^{NP} [^{NP} *e*] [^{S} ela nem se lembra [quantos homens]$_1$
Mary kissed she not+even REFL remember how+many men
~~Maria beijou t$_1$ na festa~~]] na festa]
~~Mary kissed at+the party~~ at+the party

b: * [^{S} Maria beijou [^{NP} [^{NP} *e*] [^{S} [^{NP} *e*] nem se lembra [quantos homens]$_1$
Mary kissed not+even REFL remember how+many men
~~Maria beijou t$_1$ na festa~~]] na festa]
~~Mary kissed at+the party~~ at+the party

Notice that, in Brazilian Portuguese, those gaps in subject position are attested in embedded clauses that adjoin to NPs/DPs, as shown below (cf. Rodrigues 2004: chapter 4).

(95) a: O susto de João$_1$ quando *e*$_1$ chegou em casa foi grande.
the shock of John when arrived at home was big
'John's shock when he arrived at home was huge.'

b: [^{NP} [^{NP} o susto [^{PP} de [^{NP} João]$_1$]] [$^{S'}$ quando *e*$_1$ chegou em casa]] foi
the shock of John when arrived at home was
grande
big

(96) a: Você perdeu a cara de João$_1$ quando *e*$_1$ viu Maria chegando.
you missed the face of John when saw Maria arriving
'You missed John's face when he saw Maria arriving.'

b: Você perdeu [^{NP} [^{NP} a cara [^{PP} de [^{NP} João]$_1$]] [$^{S'}$ quando *e*$_1$ viu Maria
you missed the face of John when saw Maria
chegando]]
arriving

III.3. An Alternative Neo-Conservative Analysis

Consider, now, an alternative analysis of syntactic amalgams which does not involve duplication of any chunk of structure. The basic intuition is that examples like (97) are derived from a combination of transformations that apply to input structures like (98).

(97) John invited you will never guess how many people to his party.

(98) [^{IP} you will never guess [^{CP} [^{DP} how many people]$_1$ John invited t$_1$ to his party]]

Interestingly, this idea involves much less structure than argued by Lakkof (1974). What we have in (98) is in fact a proper subset of the syntactic material involved in Lakoff's (1974) formalization.

In fact, this possibility was already mentioned (but not pursued) by Lakoff, who credited Avery Andrews for the insight.

> Presumably the residual S "John invited to his party" would be raised as in S-lifting (see Ross, 1973), and "you'll never guess how many people" moved (by some miracle) back into the right place.
>
> Lakoff (1974: 321)

Under the classical transformational approach, this idea of deriving (97) from (98) may seem to require too much extra machinery, with back-and-forth ‘miraculous’ movements. But given the tools of the Principle-&-Parameters framework, the basic mechanics is actually rather straightforward, as shown in **§III.3.1** below, though the details are not trivial at all, as shown in **§III.3.2**.

III.3.1. The Mechanics: Remnant Movement

Syntactic amalgams may be analyzed in terms of remnant movement (Muller 1998), which may be implemented in two different ways, as shown in **§III.3.1.1** and **§III.3.1.2**.

III.3.1.1. M-Scrambling, WH-Movement and IP-Topicalization

According to this technical implementation, the generation of syntactic amalgams via remnant movement would involve the following steps.

We start building the structure from the bottom upwards, up to the point in (99).

(99) [^{CP} [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} [^{DP} how many people] t$_1$ [^{PP} to his party]]]]]

Then we move both **[^{DP} how many people]** and **[^{PP} to his party]** to the left periphery of the embedded clause. The movement of **[^{DP} how many people]** is straightforward, targeting the specifier of CP, whereas **[^{PP} to his party]** undergoes some M-Scrambling-like operation, targeting a position somewhere above the specifier of IP, and below the specifier of CP. For expository reasons, I will refer to such movement as targeting the specifier of a hypothesized functional category XP, projected in between CP and IP, as in (100).

(100) [^{CP} [^{DP} how many people]$_4$ [^{XP} [^{PP} to his party]$_3$ [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} t$_4$ t$_1$ t$_3$]]]]

After that, we keep merging new elements from the bottom upwards, up to the point that (100) is embedded inside a higher IP, like in (101).

(101) [^{IP} you will never guess [^{CP} [^{DP} how many people]$_4$ [^{XP} [^{PP} to his party]$_3$
[^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} t$_4$ t$_1$ t$_3$]]]]]

Finally, the entire IP of the embedded clause moves to some specifier in the CP domain of the matrix clause, where it presumably is assigned a topic-like status, as in (102).

(102) [^{CP} [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} t$_4$ t$_1$ t$_3$]]]$_5$ [$^{C'}$ C [^{IP} you will never guess
[^{CP} [^{DP} how many people]$_4$ [^{XP} [^{PP} to his party]$_3$ t$_5$]]]]

Crucially, this is an instance of remnant movement. That is, the IP that undergoes topicalization contains two traces of phrases left behind in the left periphery of the embedded clause.

III.3.1.2. WH-Movement with Pied-Piping of VP and IP-Topicalization

Alternatively, the derivation of syntactic amalgams may be taken to be as follows.

The computational system starts building the structure from the bottom upwards, till the structure in (103) obtains.

(103) [^{CP} [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} [^{DP} how many people] t$_1$ [^{PP} to his party]]]]]

Then, the WH-phrase **[^{DP} how many people]** moves to the specifier of the embedded CP, pied-piping the entire VP,[68] yielding (104).

(104) [^{CP} [^{VP} [^{DP} how many people] t$_1$ [^{PP} to his party]]$_3$ [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ t$_3$]]]

The rest of the derivation is trivial. New elements are merged, and the phrase marker grows from the bottom upwards, up to the point that (104) is embedded inside a higher IP, as in (105).

[68] This pied-piping would be optional. As will become clearer later on, if pied-piping occurs, the final result is (i), whereas, if it doesn't, we get (ii). I'll come back to this issue on **§III.3.2** below.
(i) John invited you will never guess how many people to his party.
(ii) John invited to his party you will never guess how many people.

(105) $[^{CP}$ $[^{IP}$ you will never guess $[^{CP}$ $[^{VP}$ $[^{DP}$ how many people] t_1 $[^{PP}$ to his party]]$_3$ $[^{IP}$ John$_2$ $[^{vP}$ t_2 invited$_1$ t_3]]]]]

Finally, we move the entire IP of the embedded clause to the topic position, as in (106).

(106) $[^{CP}$ $[^{IP}$ John$_2$ $[^{vP}$ t_2 invited$_1$ t_3]]$_4$ $[^{C'}$ C $[^{IP}$ you will never guess $[^{CP}$ $[^{VP}$ $[^{DP}$ how many people] t_1 $[^{PP}$ to his party]]$_3$ t_4]]]]

Again, this is an instance of remnant movement. The IP that undergoes topicalization contains a trace of the VP left behind in the CP of the embedded clause.

III.3.2. Some Good News

There are some clear advantages of a remnant-movement approach to syntactic amalgams over Lakoff's (1974) original analysis.

First of all, we do not need to worry about the nature of those elliptical indefinite NPs/DPs, since they actually do not exist. Hence, no deletion rule is needed, and no condition on such a rule needs to be postulated or derived in order for the theory not to overgenerate structures like (107) [=(22b)], where the deletion does not take place.

(107) * John invited a surprising number of people you will never guess how many people to his party.

With no elliptical NPs to worry about, then it becomes obvious why the object of **invited** in (01) — repeated below as (108) — is interpreted as *"a number of people **n**, such that you will never guess **n**"*.

(108) John invited you'll never guess how many people to his party.

This is so simply because in the structure which the syntactic amalgam originates from — i.e. (109) — the object of the verb **guess** is a clause whose verb is **invited**; and the object of **invited** is **[^{DP} how many people]** itself, instead of an elliptical NP whose proper interpretation would require an extra mechanism to obtain. In other words, the verb **invited** takes as its complement an indirect-question which has **how many people** occupying the specifier of its CP.

(109) [^{TP} you will never guess [^{CP} [^{DP} how many people]$_1$ John invited t_1 to his party]]

Finally, as far as sluicing goes, no questions arise, as there is actually no sluicing. After all, there is no embedded sentence adjoined to **[^{DP} how many people]** to begin with, where sluicing could possibly apply. That predicts that (110) [=(22c)] should be ungrammatical. In Lakoff's (1974) analysis, something else has to be said about the obligatoriness of sluicing in such cases, as well as about 'how far' the deletion goes.

(110) * John invited you will never guess how many people John invited to his party to his party.

It seems, then, that the remnant-movement approach to syntactic amalgams solves the problems faced by Lakoff's (1974) analysis by denying that the problems exist. Once a different structure is assumed as the input to transformations, none of the problematic issues pointed in **§III.2.2** arise.

Interestingly, the remnant-movement approach to syntactic amalgams automatically accounts for the pattern in (111), not mentioned by Lakoff (1974).

(111) a: John invited you will never guess how many people to his party.

b: John invited to his party you will never guess how many people.

c: * John invited how many people to his party you will never guess.

d: * John invited how many people you will never guess to his party.

The grammaticality of both (111a) and (111b) follows from the fact that the indirect object **to his party** may or may not be part of the phrase that is topicalized, yielding either sentence as the output.

The generation of (111a) would be as discussed above in **§III.3.2.1** or **III.§3.2.2**, depending on the technical implementation adopted. When the embedded IP is topicalized, it carries the indirect object with it.

On the other hand, the generation of (111b) would be as indicated in (112), regardless of the technical implementation adopted. First, the WH-phrase moves to the specifier of the first CP above it (cf. 112a); then the resulting clause gets embedded within another sentence (cf. 112b); and eventually the embedded CP containing the trace of the moved WH-phrase is moved to a topic position in the CP domain of the matrix clause, as an instance of remnant movement (cf. 112c).

(112) a: [^{CP} [^{DP} how many people]$_4$ [^{TP} John$_2$ [^{vP} t$_2$ invited$_1$
[^{VP} t$_4$ t$_1$ [^{PP} to his party]]]]]

b: [^{TP} you will never guess [^{CP} [^{DP} how many people]$_4$
[^{TP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} t$_4$ t$_1$ [^{PP} to his party]]]]]]

c: [^{CP} [^{TP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} t$_4$ t$_1$ [^{PP} to his party]]]]$_5$
[^{TP} you will never guess [^{CP} [^{DP} how many people]$_4$ t$_5$]]]

Under the technical implementation presented in **§III.3.1.1**, this amounts to saying that the optional scrambling does not take place; whereas, under the technical implementation presented in **§III.3.1.2**, this amounts to saying that the optional pied-piping of the VP doesn't take place when the WH-phrase is moved.

The ungrammaticality of (111c) and (111d) can be accounted for by applying the same logic.

In order to generate (111c) or (111d), we need a derivation in which there is no overt movement of the WH-phrase to the specifier of the embedded CP. That way, whatever principle requires this movement to be overt in English is getting violated. The (non-convergent) derivation of (111c) would be as in (113).

(113) a: [^{CP} [^{IP} John$_2$ [^{vP} t_2 invited$_1$ [^{VP} [^{DP} how many people] t_1 [^{PP} to his party]]]]]

b: [^{IP} you will never guess [^{CP} [^{IP} John$_2$ [^{vP} t_2 invited$_1$ [^{VP} [^{DP} how many people] t_1 [^{PP} to his party]]]]]]

c: [^{CP} [^{IP} John$_2$ [^{vP} t_2 invited$_1$ [^{VP} [^{DP} how many people] t_1 [^{PP} to his party]]]]$_3$ [^{IP} you will never guess [^{CP} t_3]]]

As for (111d), its (non-convergent) derivation would be as in (114) under the scrambling approach given in **§III.3.1.1**.[69]

[69] Under the pied-piping approach, a derivation of (111d) is actually inconceivable.

(114) a: [^{CP} [^{PP} to his party]$_3$ [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} [^{DP} how many people] t$_1$ t$_3$]]]]

b: [^{IP} you will never guess [^{CP} [^{PP} to his party]$_3$ [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} [^{DP} how many people] t$_1$ t$_3$]]]]]

c: [^{CP} [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ [^{VP} [^{DP} how many people] t$_1$ t$_3$]]]]
[^{IP} you will never guess [^{CP} [^{PP} to his party]$_3$ t$_4$]]

III.3.3. The Problem of Postulating an Additional Unmotivated Movement

One of the core properties of remnant movement constructions is that all movements involved should be independently motivated (cf. Müller 1998). This is not what happens in the derivations sketched in **§III.3.1.1** and **§III.3.1.2** above.

The movement of the WH-phrase to the specifier of the lower CP is independently motivated, as the data in (115) indicates.

(115) a: How many people did John invite to his party?

b: I wonder how many people John invited to his party.

The movement of the lower IP to the topic position somewhere in the CP-shell of the matrix clause also seems to be independently motivated, although the data supporting this view (e.g. (116)) is not so conclusive.

(116) John invited two-hundred people to his party, I guess.

It is not immediately obvious that the displacement of a sentential constituent in (116) reallyinvolves IP-topicalization, rather than topicalization of the lower CP.[70]

At any rate, even if (116) really is IP-topicalization, the analysis, as a whole, is still far from trivial. The scrambling-like movement of **[^{PP} to his party]** postulated in **§III.3.1.1** is not attested in constructions other than syntactic amalgams, as exemplified by (117).

(117) * She will never know that [^{PP} to his party]$_2$ John invited a lot of people t_2.

One might say, along the lines of Müller (2000), that such scrambling-like movement of **[^{PP} to his party]** discussed in **§III.3.1.1** is parasitic on the movement of the WH-phrase **[^{DP} how many people]** to the specifier of CP, and is required in

[70] In fact, the data in (i) and (ii) below supports the view that what happens in (116) is CP-topicalization rather than IP-topicalization.

(i) a: I believe (that) Amy gave all her money to Tom.
b: Amy gave all her money to Tom, I believe [^{DP} THAT].
c: * Amy gave all her money to Tom, I believe [^{C} that].
d: That Amy gave all her money to Tom, I believe.

(ii) a: You know if/whether Amy gave all her money to Tom.
b: * Amy gave all her money to Tom, you know if/whether.

I am thankful to Andrew Nevins for discussion on this matter.

order to satisfy a principle of shape conservation, which demands that the original linear order of the arguments inside the VP must be kept constant when they are extracted.

Putting aside issues about the *ad hoc* character of such a principle, the main empirical problem with this idea is that not only is (117) ungrammatical, but also (118) and (119), where that the linear order of the arguments inside the VP is kept constant after they are extracted.

(118) * [^{DP} how many people]$_1$ [^{PP} to his party]$_2$ did John invite t_1 t_2?

(119) * I wonder [^{DP} how many people]$_1$ [^{PP} to his party]$_2$ John invited t_1 t_2.

Moreover, sentences like (120) are grammatical, despite the fact that the linear order of the arguments is not conserved.[71]

(120) John invited to his party you will never guess how many people.

It is, thus, not clear how such a condition on representations would work, and how it would interact with other grammatical principles, to yield the desired results.

[71] In Lakoff's (1974) system, that could be achieved by adjoining the embedded sluiced clause to the VP, rather than to the elliptical indefinite NP, which, of course, poses many questions regarding the freedom of.

(i) [^{S} John [^{VP} [^{VP} invited [$^{NP}\Delta$] to his party] [^{S} you'll never guess [^{NP} how many people]$_1$ ~~John invited t_1 to his party~~]]]

Technically, one could play with formal definitions in such a way that the scrambling-like movement is parasitic on both the WH-movement and the remnant IP-topicalization. But that would face the conceptual problem of massive globality/look-ahead, since, given the *Extension Requirement* (Chomsky 1995: 190, 327-328), **[^{PP} to his party]** should move before both **[^{DP} how many people]** and **[^{IP} John invited t_{WH} t_{PP}]** move, since **[^{PP} to his party]** occupies a lower specifier than the other two phrases. What that amounts to saying is that a certain movement operation ***x*** is parasitic on other two movement operations ***y*** and ***z***, which that have not taken place yet when ***x*** applies. Not even the triggers of ***y*** and ***z*** are present at that stage.

As for the movement of the WH-phrase and the scrambled PP, the reverse derivational order (i.e. first the WH and then the PP) is in principle available if we assume, with Richards (1997) and Castillo & Uriagereka (2000), that tucking-in is allowed (see also Chomsky 2000: 135-138). It is less trivial to assume that the movement of the scrambled PP derivationally follows the topicalization of the remnant IP. Besides tucking-in, that would imply movement out from a copy in the tail of the chain.[72]

In any case, if we assume a bottom-up system in which extension holds in its strongest form, it seems that we will always need to postulate the existence of at least one movement that is not independently available. Therefore, the attempt

[72] See Nunes & Uriagereka (2000) for arguments against movement of chain-tails.

to derive syntactic amalgams without appealing to construction-specific principles fails.

We should also worry about this order-conservation principle being sensitive to linear order if we take linear order to be established only when syntax interfaces phonology (Chomsky 1995: 334-340), on the basis of (a version of) Kayne's (1994) LCA. As Müller (2000) himself admits, the relevant notion has to be precedence, not c-command, for reasons that I will not discuss here. One potential problem with this approach is that precedence relations are being established for non-terminals, therefore a crucial aspect of the Antisymmetry Theory (Kayne 1994) gets missed; namely, that linearization applies to all and only terminal elements.

One could also assume a system that interfaces PF and LF in cycles (cf. Uriegereka 1999; Chomsky 2000, 2001b, 2001c; Guimarães 1999; *inter alia*), therefore allowing linearization to be computed for local domains. But if precedence is a property of PF objects only, and if syntactic structure is lost when PF objects are generated, we need a way of avoiding pronunciation of traces — so to speak — because, once a domain has been spelled-out, everything inside it would be stuck there as far as phonology goes. In other words, we need something like Chomsky's (2000, 2001b, 2001c) notion of *phase*s with edges that are accessible to the next round of syntactic operations, and are mapped to PF together with the material in the next cycle. One could say, then, that the scrambled PP in (121) [=(100)] above uses the edge of its phase to escape from the

VP and keep the order of arguments constant just in case the next phase involves the movement of the remnant IP, therefore requiring both objects to be adjacent.

(121) $[^{CP}$ $[^{DP}$ how many people$]_4$ $[^{XP}$ $[^{PP}$ to his party$]_3$ $[^{IP}$ John$_2$ $[^{vP}$ t_2 invited$_1$ $[^{VP}$ t_4 t_1 t_3]]]]

This can be a local computation, with no look-ahead. But what if nothing in the next phase requires such adjacency? In principle, we should expect the PP to be pronounced at the edge of the embedded CP, since the previous phase has already been spelled-out without pronunciation of either argument. But that would wrongly predict that we should get (123a) and (123b) instead of (122a) and (122b), respectively. This is so because the PP is moved to the edge of its phase anyway, no matter whether there is a higher phase, and no matter whether something in a higher phase will require the PP to be adjacent to the WH-phrase.

(122) a: $[^{DP}$ how many people$]_1$ did John invite t_1 $[^{PP}$ to his party]?

b: I wonder $[^{DP}$ how many people$]_1$ John invited t_1 $[^{PP}$ to his party]

(123) a: * $[^{DP}$ how many people$]_1$ $[^{PP}$ to his party$]_2$ did John invite t_1 t_2?

b: * I wonder $[^{DP}$ how many people$]_1$ $[^{PP}$ to his party$]_2$ John invited t_1 t_2.

Similar problems arise under the pied-piping approach given in **§III.3.1.2**.

First of all, why should we pied-pipe the whole VP when moving the WH-phrase? Is this a legitimate configuration? How do we handle the optionality? Presumably, we don't want to assume *ad hoc* pied-piping features for that.

Also, what is it that allows the pied piping in constructions like (124) [=(104)], but not in cases like (125)[73]?

(124) [^{CP} [^{VP} [^{DP} how many people] t$_1$ [^{PP} to his party]]$_3$ [^{IP} John$_2$ [^{vP} t$_2$ invited$_1$ t$_3$]]]

(125) a: * [^{VP} [^{DP} how many people] t$_1$ [^{PP} to his party]]$_2$ did John invite$_1$ t$_2$?

b: * I wonder [^{VP} [^{DP} how many people] t$_1$ [^{PP} to his party]]$_2$ John invited$_1$ t$_2$.

Finally, the movement of the VP to the specifier of CP is independently problematic. Crucially, we have to assume that the head of VP has been moved to *v*P, otherwise, we wrongly predict the generation of (126), whose structure would be as in (127), where the verb ends up inside the phrase that is in the specifier of the lower CP, at the right periphery of the sentence.

(126) * John did, you will never guess how many people invite to his party.

(127) * [^{CP} [^{IP} John did t$_1$]$_2$ [$^{C'}$ C [^{IP} you will never guess [^{CP} [^{VP} how many people invite to his party]$_1$ [$^{C'}$ C t$_2$]]]]]

[73] As far as PF is concerned, (123) and (125) are identical pairs, but two different syntactic structures are being postulated.

As a consequence, the movement of the lower VP to the specifier of the lower CP is itself an instance of remnant movement, in which the trace of the phrase being moved is the head of that very phrase. The problem is that such an instance of remnant movement is illicit, and not independently available (cf. Takano 2000), as exemplified in (127), which contrasts with (128), (129) and (130).

(127) * $[^{VP}$ $[^{DP}$ a book] $[^{V'}$ t_1 $[^{PP}$ to Mary$]]]_3$ ... $[^{IP}$ $[^{DP}$ John$]_2$ $[^{I'}$ I $[^{vP}$ t_2 $[^{v'}$ v+gave$_1$ $t_3]]]]$

(128) $[^{DP}$ a book$]_3$ ... $[^{IP}$ $[^{DP}$ John$]_2$ $[^{I'}$ I $[^{vP}$ t_2 $[^{v'}$ v+gave$_1$ $[^{VP}$ t_3 $[^{V'}$ t_1 $[^{PP}$ to Mary]]]]]]

(129) $[^{PP}$ to Mary$]_3$ ... $[^{IP}$ $[^{DP}$ John$]_2$ $[^{I'}$ I $[^{vP}$ t_2 $[^{v'}$ v+gave$_1$ $[^{VP}$ $[^{DP}$ a book] $[^{V'}$ t_1 $t_3]]]]]]$

(130) $[^{vP}$ t_2 $[^{v'}$ v+give$_1$ $[^{VP}$ $[^{DP}$ a book] $[^{V'}$ t_1 $[^{PP}$ to Mary$]]]]]_3$ ... $[^{IP}$ $[^{DP}$ John$]_2$ $[^{I'}$ did $t_3]]$

III.3.4 Two Alternative Implementations of the Remnant-Movement Analysis

The two technical implementations presented in **III.3.2** are minimally different versions of essentially the same analysis, with basically the same virtues and problems. As shown in **III.3.3**, they both fail to achieve explanatory adequacy, as an *ad hoc* type of movement operation needs to be stipulated just for syntactic amalgams. In this section, I will briefly consider two more alternative technical implementations to the remnant movement analysis of amalgamation, which apparently do not face such a problem. In one case, there is actually an extra movement of a PP, but it is taken to be just one more instantiation of the more general process of *Rightward Movement*. In the other case, no additional scrambling-like movement of PPs is stipulated (and its effects is taken to be derived from independently available mechanisms of 'chain pronunciation').

III.3.4.1. Rightward Movement

One potential way of technically implementing the remnant movement analysis of amalgamation without facing the problem pointed out in **III.3.3** is to postulate that the movement of the rightmost PP in the problematic cases above is an instance of *Rightward Movement* (Ross 1967, 1973), which is arguably a stylistic/optional operation wildly available in natural languages, and not a construction-specific sub-rule of amalgamation. (cf. Akmajian 1975; Johnson 1986; Postal 1998; McCloskey 1999; Sabbagh 2003).

Abstracting away from technical details about the inner-workings of *Rightward Movement*, this approach predicts that (131) [= (111b)] and (132) [= (111a)] can be both derived by the same grammar, depending on whether or not the optional rightward movement of the PP out of the lower VP applies.

(131) John invited to his party you will never guess how many people.

(132) John invited you will never guess how many people to his party.

The respective derivations would be as in (133) and (134).

(133) *Derivation Without Rightward Movement*

a: [$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited [$^{\mathrm{DP}}$ how many people] [$^{\mathrm{PP}}$ to his party]]]]

b: [$^{\mathrm{CP}}$ **[$^{\mathrm{DP}}$ how many people]$_1$** [$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited **t$_1$** [$^{\mathrm{PP}}$ to his party]]]]]

c: [$^{\mathrm{IP}}$ you will [$^{\mathrm{VP}}$ never guess [$^{\mathrm{CP}}$ [$^{\mathrm{DP}}$ how many people]$_1$
[$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited t$_1$ [$^{\mathrm{PP}}$ to his party]]]]]]]

e: [$^{\mathrm{CP}}$ **[$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited t$_1$ [$^{\mathrm{PP}}$ to his party]]]$_3$** [$^{\mathrm{IP}}$ you will [$^{\mathrm{VP}}$ never
guess [$^{\mathrm{CP}}$ [$^{\mathrm{DP}}$ how many people]$_1$ **t$_3$**]]]]

(134) *Derivation With Rightward Movement*

a: [$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited [$^{\mathrm{DP}}$ how many people] [$^{\mathrm{PP}}$ to his party]]]]

b: [$^{\mathrm{CP}}$ **[$^{\mathrm{DP}}$ how many people]$_1$** [$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited **t$_1$** [$^{\mathrm{PP}}$ to his party]]]]]

c: [$^{\mathrm{CP}}$ [$^{\mathrm{CP}}$ [$^{\mathrm{DP}}$ how many people]$_1$ [$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited t$_1$ **t$_2$**]]]]
[$^{\mathrm{PP}}$ to his party]$_2$]

d: [$^{\mathrm{IP}}$ you will [$^{\mathrm{VP}}$ never guess [$^{\mathrm{CP}}$ [$^{\mathrm{CP}}$ [$^{\mathrm{DP}}$ how many people]$_1$
[$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited t$_1$ **t$_2$**]]]] **[$^{\mathrm{PP}}$ to his party]$_2$**]]]

e: [$^{\mathrm{CP}}$ **[$^{\mathrm{IP}}$ John [$^{v\mathrm{P}}$ invited t$_1$ t$_2$]]** [$^{\mathrm{IP}}$ you will [$^{\mathrm{VP}}$ never guess

$[^{CP}$ $[^{CP}$ $[^{DP}$ how many people$]_1$ **t_3**]] **$[^{PP}$ to his party$]_2$**]]]]

III.3.4.2. Chain-Internal Selective Deletion of Copies

Working under the framework of the *Copy Theory of Movement* (Chomsky 1995: chapter 4), Wilder (1995) and Boskovic (2001), among others, have explored the possibility that, in certain special circumstances, the PF-deletion of chain links (i.e. copies of the moved element) may not target the entire lower copy, as usual. Rather, deletion would affect the node(s) corresponding to a given substring at the lower copy, and the node(s) corresponding to complementary substring at the higher copy.

Abstracting away from details of how chain-internal selective deletion works, this approach might in principle be applicable to the case in point, and the prediction would be that (135) [= (131)] and (136) [= (133)] can be both derived by the same grammar, depending on whether there is chain-internal selective deletion or ordinary deletion of the chain formed by the two copies of the topicalized IP.

(135) John invited to his party you will never guess how many people.

(136) John invited you will never guess how many people to his party.

The respective derivations would be as in (137) and (138).

(137) *Derivation With Canonical PF-Deletion of Copies*

a: [^{IP} John [^{vP} invited [^{DP} how many people] [^{PP} to his party]]]]

b: [^{CP} **[^{DP} how many people]** [^{IP} John [^{vP} invited [^{DP} ~~how many people~~] [^{PP} to his party]]]]]

c: **[^{IP} you will [^{VP} never guess** [^{CP} [^{DP} how many people] [^{IP} John [^{vP} invited [^{DP} ~~how many people~~] [^{PP} to his party]]]]]**]]**

d: [^{CP} **[^{IP} John [^{vP} invited [^{DP} ~~how many people~~] [^{PP} to his party]]]** [^{IP} you will [^{VP} never guess [^{CP} [^{DP} how many people] ~~[^{IP} John [^{vP} invited [^{DP} how many people] [^{PP} to his party]]]~~]]]]]

(138) *Derivation With Canonical Chain-Internal Selective PF-Deletion of Copies*

a: [^{IP} John [^{vP} invited [^{DP} how many people] [^{PP} to his party]]]]

b: [^{CP} **[^{DP} how many people]** [^{IP} John [^{vP} invited [^{DP} ~~how many people~~] [^{PP} to his party]]]]]

c: **[^{IP} you will [^{VP} never guess** [^{CP} [^{DP} how many people] [^{IP} John [^{vP} invited [^{DP} ~~how many people~~] [^{PP} to his party]]]]]**]]**

d: [^{CP} **[^{IP} John [^{vP} invited [^{DP} ~~how many people~~] [^{PP} ~~to his party~~]]]**

[^{IP} you will [^{VP} never guess [^{CP} [^{DP} how many people] [^{IP} ~~John~~ [^{vP} ~~invited~~ [^{DP} ~~how many people~~] [^{PP} to his party]]]]]]]

III.3.4.3. New Issues that Arise from the Alternative Analyses

The two alternative analyses in **§III.3.4.1** and **§III.3.4.2** both seem to be successful attempts to dispense with *ad hoc* movement, consequently not assigning any special theoretical status to amalgamation. However, each of these analyses also raises its own nontrivial issues, which also relate to 'last resort'.

The rightward-movement-based analysis, on the one hand, equates the extra movement of a PP — necessary to account for cases like (132)/(136) — with the allegedly more general operation of *Rightward Movement*. Although this analysis has the virtue of potentially unifying various phenomena into a single mechanics, this can also be a problem, to the extent that the very notion of rightward movement itself is not straightforward in minimalist grounds. Its optional nature is incompatible with the minimalist assumption that movement is a last resort operation, which is a crucial aspect of any theory based on the notion of economy of derivations.[74]

[74] Moreover, if Kayne's (1994) *Antisymmetry Theory* is correct, any instance of rightward movement should be discarded from the outset, unless one formalizes it in terms of optional leftward movement of the PP obligatorily followed by an extra (remnant) movement of a heavy constituent containing the trace of the movement PP to a position right above the landing site of

One may hypothesize, then, that rightward movement, when it happens, is triggered by the need to satisfy some additional requirement, caused by the presence of some extra feature or functional projection, which is not always present. That, of course, is not an explanation, but merely a way of encoding the facts in our meta-language, unless we can detect some interpretive effect associated with rightward movement, such as focalization or topicalization, which would constitute evidence for the existence of such extra movement-triggering devices. That might be true of some cases for which rightward movement has been proposed, but there seems to be no sign that this is the case with syntactic amalgams.

On the other hand, the analysis based on chain-internal selective deletion denies the existence of the extra movement of a PP, deriving its effects from mechanisms that are arguably necessary for independent reasons. The rational behind the idea of chain-internal selective deletion is that canonical PF-deletion of chain links is the default strategy by virtue of it being the most economical strategy, whereas non-canonical deletion is a more costly strategy that the system applies as a last resort, only in special structural contexts demanding certain prosodic patterns that depend on non-canonical linearization to obtain (cf. Boskovic 2001, for details).

The problem with extending this logic to the treatment of syntactic amalgamation is that there seems to be no such 'special circumstances' or

the moved PP. This, of course, raises even more issues related to last resort, as it is unclear what would optionally trigger the movement of the PP in the first place, and what would obligatorily trigger the movement of the remnant constituent if and only if the movement of the PP.

'additional prosodic demands' present in the structures for which chain-internal selective deletion is being postulated. Therefore this analysis is also problematic with regards to last resort.

In any event, even if either of these two analyses turns out to be non-problematic with respect to last resort once the relevant details are worked out, it still does not immediately follow that either one represents a significant improvement over the other remnant-movement-based approaches presented in **§III.3.2**. This is so because the rightward-movement analysis and the non-canonical linearization analysis are both based on remnant-movement mechanics, to the same extent that the two analyses in **§III.3.2** are. Putting aside the extra movement of a PP in some cases, all those analyses share the key property of taking syntactic amalgams to involve movement of a WH-phrase out of an embedded IP,[75] followed by the movement of that whole (remnant) IP to a topic position in the left periphery of the matrix clause. If this is on the right track, we expect this alleged movement of IP to be subject to the usual constraints on movement, otherwise the remnant movement approach will suffer from the problem assigning a construction-specific status to syntactic amalgams, just like the sluicing based approach does.

In the next section, I will argue that some of the facts presented in chapter **II** are incompatible with any of the versions of the neo-conservative (remnant-

[75] In the case of cleft-amalgams, there would be no WH-phrase to begin with. But the logic is the same, as there would be a DP dislocated to some position in the left periphery of the embedded clause, followed by the movement of the remnant IP to a topic position in the left periphery of the matrix clause.

movement-based) approach, as they would require further stipulation to rule in instances of movement that would otherwise violate well-known constraints on movement.

III.3.5 Further Problems for the Remnant Movement Approach

III.3.5.1. Embedded Amalgams

Empirical problems arise when any version of the remnant-movement analysis is applied to more complex cases like (139).

(139) I believe that Amy gave all her money to you know who.

Details of technical implementation aside, there are two possible ways of analyzing cases like (139) under the remnant-movement approach. Either (140) or (141) could potentially be the derivation that generates (139).

(140) a: **building the embedded clause**

[^{IP} Amy gave all her money to who]

b: **local WH-movement**

[^{CP} who$_1$ [^{IP} Amy gave all her money to t$_1$]]

c: **building the intermediate clause**

[^{IP} you know [^{CP} who$_1$ [^{IP} Amy gave all her money to t$_1$]]]

d: **remnant-movement of the embedded IP to a topic position**

[^{CP} [^{IP} Amy gave all her money to t$_1$]$_2$ [^{IP} you know [^{CP} who$_1$ t$_2$]]]

e: **building the matrix clause**

[^{CP} I believe that [^{CP} [^{IP} Amy gave all her money to t$_1$]$_2$ [^{IP} you know [^{CP} who$_1$ t$_2$]]]]

(141) a: **building the lowest embedded clause**

[^{IP} Amy gave all her money to who]

b: **local WH-movement**

[^{CP} who$_1$ [^{IP} Amy gave all her money to t$_1$]]

c: **building the highest embedded clause**

[^{IP} I believe that [^{CP} who$_1$ [^{IP} Amy gave all her money to t$_1$]]]

d: **successive cyclic WH-movement**

[^{CP} who$_1$ [^{IP} I believe that [^{CP} t$_1$ [^{IP} Amy gave all her money to t$_1$]]]]

e: **building the matrix clause**

[^{IP} you know [^{CP} who$_1$ [^{IP} I believe that [^{CP} t$_1$ [^{IP} Amy gave all her money to t$_1$]]]]]

f: **remnant-movement of the highest embedded IP to a topic position**

[^{CP} [^{IP} I believe that [^{CP} t$_1$ [^{IP} Amy gave all her money to t$_1$]]] [^{IP} you know [^{CP} who$_1$ t$_2$]]]

The problem is that, while either of these alternative derivations can account for the word-order, neither accounts for the meaning.

Given standard assumptions about semantic compositionality, the output of the derivation in (140) should be about the speaker's belief in the listener's knowledge of the fact that Amy gave money to someone. Similarly, the sentence generated from the derivation in (141) should be about the listener's knowledge of the speaker's belief in the fact that Amy gave money to someone.

As a matter of fact, neither of these possibilities corresponds to the actual meaning of (139), in which the listener's knowledge has nothing to do with the speaker's belief, and vice versa. In fact, there are two parallel messages in (139). One of them concerns the speaker's belief in the fact that Amy gave money to someone. The other one concerns the listener's knowledge of the fact that Amy gave money to someone.

III.3.5.2. Absence of Island Effects

Another piece of evidence against all remnant-movement-based approaches comes from the fact that it is possible for amalgams to occur inside syntactic domains that are typical islands for extraction (cf. Tsubomoto & Whitman 2000), as in (142), as discussed in **§II.5**.

(142) a: * I don't remember [when]$_1$ John lives in [a house]$_2$ {that he built *e*$_2$ t$_1$}

b: John lives in [a house]$_2$ {that he built *e*$_2$ I don't remember when}

This is a counter-example for the remnant-movement analysis, since it would require a derivation like (143), which involves an extraction of a relative-clause island in step (b).

(143) a: [^{IP} John lives in [^{NP} [^{NP} a house]$_2$ [^{CP} that he built e_2 when$_1$]]]

b: [^{CP} **when$_1$** [^{IP} John lives in [^{NP} [^{NP} a house]$_2$ [^{CP} that he built e_2 **t$_1$**]]]]

c: [^{IP} I don't remember [^{CP} when$_1$ [^{IP} John lives in [^{NP} [^{NP} a house]$_2$ [^{CP} that he built e_2 t$_1$]]]]]

d: [^{CP} **[^{IP} John lives in [^{NP} [^{NP} a house]$_2$ [^{CP} that he built e_2 t$_1$]]]$_3$** [^{IP} I don't remember [^{CP} when$_1$ **t$_3$**]]]

III.3.5.2. Multiple Amalgamation

Yet another problem for the remnant-movement approach concerns multiple amalgamation. Consider (144).

(144) John invited *you will never guess how many people* to *you can imagine what kind of party*.

In this case, it is impossible to apply any of the movement-based analysis successfully, unless we assume that the system overlooks/forgives violations of the relevant locality constraints on WH-movement in two derivational steps. The derivation for (144) would be as in (145).[76]

(145) a: [^{IP} John invited [^{DP} how many people] [^{PP} to [^{DP} what kind of a party]]]

b: ☹ [^{CP} [^{DP} what kind of a party]$_1$ [^{IP} John invited [^{DP} how many people] [^{PP} to t$_1$]]]

c: [^{IP} you can imagine [^{CP} [^{DP} what kind of a party]$_1$ [^{IP} John invited [^{DP} how many people] [^{PP} to t$_1$]]]]

d: [^{IP} [^{PP} to t$_1$]$_2$ [^{IP} you can imagine [^{CP} [^{DP} what kind of a party]$_1$ [^{IP} John invited [^{DP} how many people] t$_2$]]]]

e: ☹ [^{CP} [^{DP} how many people]$_3$ [^{IP} [^{PP} to t$_1$]$_2$ [^{IP} you can imagine [^{CP} [^{DP} what kind of a party]$_1$ [^{IP} John invited t$_3$ t$_2$]]]]]

f: [^{IP} you will never guess [^{CP} [^{DP} how many people]$_3$ [^{IP} [^{PP} to t$_1$]$_2$ [^{IP} you can imagine [^{CP} [^{DP} what kind of a party]$_1$ [^{IP} John invited t$_3$ t$_2$]]]]]]

g: [^{CP} [^{IP} John invited t$_3$ t$_2$] [^{IP} you will never guess [^{CP} [^{DP} how many people]$_3$ [^{IP} [^{PP} to t$_1$]$_2$ [^{IP} you can imagine [^{CP} [^{DP} what kind of a party]$_1$ t$_4$]]]]]]

[76] An additional issue with (24) is the scrambling-like movement of the **[^{PP} to ~~[^{DP} what kind of party]~~]** in step (d), which is not independently motivated. However, as discussed in footnote 6, it is possible to get rid of this extra *ad hoc* movement under an alternative implementation of the remnant-movement analysis.

Notice that, in step (b), the WH-phrase **what kind of party** moves to the lowest spec/CP crossing the other WH-phrase **how many people**, despite the fact that **how many people** is closer to the target than **what kind of party** is. Conversely, in step (e), **how many people** moves to the intermediate spec/CP crossing over **what kind of party**, despite the fact that **what kind of party** is closer to the target than **how many people** is.

These two instances of non-local WH-movement are problematic in themselves, given the standard assumptions about UG principles – strongly supported by cross-linguistic empirical generalizations about locality effects in WH-movement – demanding that **how many people** should move at that step (cf. Rizzi's (1990) *Relativized Minimality*, Chomsky's (1995) *Minimal Link Condition*). Besides, this absence of locality effects in WH-movement are not independently attested in a less convoluted version of the sentence (i.e. without the IP-topicalization movements), as indicated by the unacceptability of (146).

(146) * [^{S3} you will never guess [how many people]$_1$ [^{S2} you can imagine [what kind of party]$_2$ [^{S1} John invited t$_1$ to t$_2$]]]

Moreover, even putting aside syntactic locality matters, the actual meaning of (144) is not compatible with the structure postulated in the analysis in (145), given standard assumptions about semantic compositionality. In (145), the clause corresponding to the invitation event is the complement of the verb **imagine**; and the clause corresponding to the imagining event is the complement of the verb **guess**. Therefore, this analysis wrongly predicts a meaning in which what is being guessed is something about an event of imagining that concerns an invitation event. But this is not what (144) means. Instead, the meaning of (144) is such that what is being guessed is something about the invitation event itself, which is also what is being imagined. The events of imagining and guessing are independent.

In a nutshell, examples containing multiple 'clause invasion' constitute strong empirical evidence against the remnant movement analysis to amalgamation.

Appendix to Chapter III

1. Avery Andrews's Case

(01) John invited you will never guess how many people to his party.

(02) John invited you will never guess how many people to you can imagine what kind of a party at it should be obvious where.

(03) For all contexts C, if (i) & (ii) & (iii) & (iv), then (v):

i: S_1 is an indirect question with S_0 as its complement S;

ii: S_2 is the ith phrase marker in a derivation D whose logical structure is conversationally entailed by the logical structure of S_1 in context C;

iii: NP_1 is an NP in S_2, such that S_2 minus NP_1 is identical to S_0;

iv: S_1 has the force of an exclamation;

v: relative to context C, S_1 minus S_0 may occur in place of NP_1 in the i+1th phrase marker of derivation D.

2. Larry Horn's Case

(04) John is going to I thinks it's Chicago on Saturday.

(05) John is going to I thinks it's Chicago on, I'm pretty sure he said it was Saturday to deliver a paper on Was it morpholexemes?

(06) For all contexts C, if (i) & (ii) & (iii) & (iv), then (v):

i: S_1 is a sentence with an embedded cleft-sentence with S_0 as its relative clause;

ii: S_2 is the i^{th} phrase marker in a derivation D whose logical structure is conversationally entailed by the logical structure of S_1 in context C;

iii: NP_1 is an NP in S_2, such that S_2 minus NP_1 is identical to S_0 minus the relative pronoun;

iv: S_1 is a hedged assertion of the content of S_2;

v: relative to context C, S_1 minus S_0 may occur in place of NP_1 in the $i{+}1^{th}$ phrase marker of derivation D.

3. Performative Predicate Modifiers

(07) Since the President said you were to take orders from me, get me the missing tapes.

(08) For all contexts C, if (i) & (ii), then (iii):

i: S_0 is modified by the reason-clause S_1;

ii: S_2 is the i^{th} phrase marker in a derivation D such that the logical structure of S_0 is either a felicity condition for, or a called-for response to, a logical structure S_3 which is conversationally entailed by the logical structure of S_2 in context C;

iii: relative to context C, S_1 may occur as a reason-clause modifier of S_2 in the $i+1^{th}$ phrase marker of derivation D.

example: S_0 = I have authority to give you orders.

S_1 = The president said you were to take orders from me.

S_2 = I would appreciate it if you would supply me with the missing tapes.

S_3 = I order you to get me the missing tapes.

4. Mark Liberman’s because-cases

(09) I’m afraid the Knicks are going to win, because who on the Celts can possibly handle Frazier?

(10) For all contexts C, if (i) & (ii) & (iii), then (iv):

i: the sentence consisting of S_0 modified by the reason-clause S_1 is true in C;

ii: S_4 conversationally entails an assertion of S_1 in C;

iii: S_2 is the i^{th} phrase-marker in a derivation D such that the logical structure of S_0 is a felicity condition for, or a called-response to, a logical structure S_3 which is conversationally entailed by the logical structure of S_2 in context C;

iv: relative to context C, S_4 may occur as a reason-clause modifier of S_2 in the i^{th} phrase-marker of derivation D.

example: S_0 = I believe that the Knicks are going to win.

S_1 = No one on the Celts can possibly handle Frazier.

S_2 = I’m afraid the Knicks are going to win.

S_3 = The Knicks are going to win.

S_4 = Who on the Celts can possibly handle Frazier?

5. Mark Liberman's or-cases

(11) i: You better get out, or somebody'll slug you.

ii: I think you'd better get out, or I'm afraid I'll have to throw you out.

(12) For all contexts C, if (i) & (ii) & (iii), then (iv):

i: the sentence consisting of S_0 modified by the reason-clause IF NOT S_0, THEN S_1 is true in C;

ii: S_4 conversationally entails an assertion of S_1 in C;

iii: S_2 is the i^{th} phrase-marker in a derivation D such that the logical structure of S_0 is a felicity condition for, or a called-response to, a logical structure S_3 which is conversationally entailed by the logical structure of S_2 in context C;

iv: relative to context C, S_4 may occur disjoined to the right of S_2 in the i^{th} phrase-marker of derivation D.

6. Tag Questions[77]

(13) You couldn't open the door, could you?

[77] No rule for Tag Questions was formalized by Lakoff (1974).

IV

Overlapping Computations, Dynamic Phrase-Structure, and Shared Constituency

As said in **§I.2**, I am conducting this research and searching for answers to questions about syntactic amalgamation biased by the *desiderata* of the Minimalist Program (Chomsky 1993, 1995, 2000a, 2000b, 2001a, 2001b; Martin & Uriagereka 2000; Uriagereka 1998, 2001, 2002). The main *desideratum* of Minimalism is that grammatical patterns follow from principles optimal design. In the limit, that amounts to saying that every formal property of a linguistic expression is a consequence of economy principles, and every substantive property follows from interface demands.

In what follows, I advocate for a derivational approach to Syntax, as in mainstream Minimalism. However, I departure from this tradition with regards to the 'directionality of derivations', in that I propose that tree-growth proceeds in a top-to-bottom fashion, through a structure-building mechanism where constituency is heavily dynamic. Moreover, I advocate for non-standard representations, with structure-sharing and multiply-rooted phrase markers. That way, constructions that appear to be of a paratactic nature can be seen as a product of syntax itself, pushed to the limit.

IV.1. The Input to The Computational System

Following Chomsky (1995: 225-228), I assume that the inputs to syntactic derivations are 'initial arrays of lexical items', conventionally called *Numerations*, and defined as sets of lexical tokens.[78] The role that the numeration plays in the system is the one of a 'reference set' that establishes a local domain where convergence and economy are evaluated. For instance, the sentence in (01) — whose syntactic structure is arguably something along the lines of (02) — is taken to be generated from the numeration in (03).

(01) Homer kissed many women at the party.

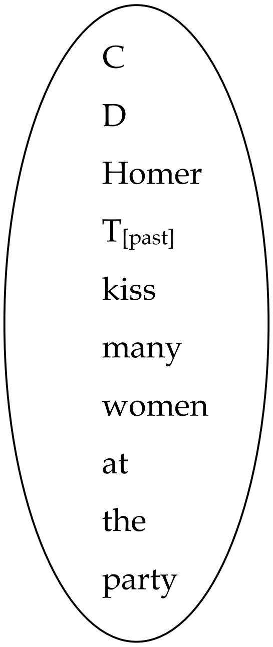
(02) [$^{\text{CP}}$ C [$^{\text{TP}}$ [$^{\text{DP}}$ D Homer] [$^{\text{T}'}$ T [$^{\text{VP}}$ ~~[$^{\text{DP}}$ D Homer]~~ [$^{\text{V}'}$ [$^{\text{V}'}$ kissed [$^{\text{DP}}$ many women]] [$^{\text{PP}}$ at [$^{\text{DP}}$ the party]]]]]]]

(03) {C, D, Homer, T$_{[\text{past}]}$, kiss, many, women, at, the, party}

[78] It has been often assumed that the members of each numeration are not just tokens of lexical items, but rather ordered pairs like <***x***,***y***> (standardly notated as x_y), where ***x*** is a type of lexical item, and ***y*** is an index that consists of positive integer which determines how many tokens of ***x*** are available for the computational system.. Technically speaking, this amounts to saying that the numeration is not an ordinary set, but rather a 'multiset' (Chomsky 1995: 225-228; Uriagereka 1998: 289-297; Gärtner 2002: 56-61). As far as I can see, this technicality is not relevant for the present purposes (if relevant at all), as the difference between $N = \{X_1, Y_2, Z_3\}$ and $N = \{X, Y_a, Y_b, Z_a, Z_b, Z_c\}$ is obviously merely notational. In what follows, I will adopt the second notation, dropping token indices for the sake of exposition unless they are necessary.

The mathematical object in (03) can be equivalently presented by the Venn diagram notation in (04).

(04)



A syntactic derivation, then, is a complex function that maps the numeration (e.g. (05a)) into a phrase marker (e.g. (05b)), in a step-by-step fashion.

(05) a: {C, D, Homer, $T_{[past]}$, kiss, many, women, at the party}

↓

successive applications of structure-building operations

↓

b: [^{CP} C [^{TP} [^{DP} D Homer] [$^{T'}$ T [^{VP} ~~[^{DP} D Homer]~~ [$^{V'}$ [$^{V'}$ kissed [^{DP} many women]] [^{PP} at [^{DP} the party]]]]]]]

The standard take on how this mapping takes place is that the items of the numeration are introduced into the derivational workspace Σ, each one by a distinct application of an operation of the computational system called *Select*. Those elements are integrated into the LF phrase marker that is being built in Σ, through multiple applications of the operations *Merge* and *Move*,[79] which assemble complex phrases in a recursive fashion.[80]

One potential conceptual problem with this approach is that it assigns a theoretical status to the notion of Numeration, which may apparently correspond to a tacit recognition of an extra level of representation besides PF and LF. Thus, the Numeration would be as an unwelcome 'residue of D-structure' that goes against the minimalist *desideratum* that the grammar only has levels of representations that are interface levels.[81]

[79] In mainstream Minimalism, Move is understood as a combination of three distinct (but related) operations: Copy, Merge and PF-Delete (Chomsky 1995: chapter 4). First, a given constituent **X** that is already integrated into the LF phrase marker under construction is copied, then that new copy of **X** is merged with some other constituent (the root node of the spine of the tree). Eventually, all copies but the highest one are deleted at PF. Recently, Chomsky (2000, 2001a, 2001b) has proposed to decompose Move even further into Agree and Pied-Pipe, such that the former would be the feature-checking operation per se, whereas the latter would be an EPP-driven mechanism whose inner workings involve Copy, Merge, and PF-Delete.

[80] Some of these steps are logically ordered with respect to others (e.g. in order for the system to merge **at** with **[^{DP} the party]**, it must be the case that the complex phrase **[^{DP} the party]** exists in the first place, which presupposes the anteriority of the merging of **the** with **party**), while others aren't (e.g. the merging of **many** with **women**, and the merging of **D** with **Homer**).

[81] Of course, if it can be argued on independent grounds, that such D-structure-like level of representation interfaces with some module of the cognitive system, then it would be justified on minimalist grounds. As pointed out by Uriagereka (forthcoming, chapter 1), the core minimalist assumption with regards to levels of representations is not that that PF and LF are the only levels. Rather, the assumption is that the grammar only has interface levels.

Prima Facie, this may not be much of a problem as long as no well-formedness condition is posited on Numerations. However, to some extent, that seems inevitable, since, no matter how 'flat' and 'unstructured' it is, it must satisfy some formal condition or other, in order to count as a set-theoretical object of a given kind. Needless to say, that is already a condition on well formedness of the Numeration.

At any rate, it is virtually conceptually necessary to have some function that creates lexical tokens from lexical types, in order to feed the derivation. After all, phrase markers are made of tokens, while the Lexicon is a collection of types.

Under the standard assumption that lexicon is a collection of morphemes, and that words are combinations of morphemes, the null hypothesis is that such mapping function is nothing but standard morphology.

Taking seriously the idea of Numerations being sets has some interesting consequences, which will play a major role in this dissertation, in the treatment of parataxis.

First, consider the simpler (arguably hypotactic) constructions in (06) and (08), and their respective numerations in (07) and (09).

(06) His wife just found out that Homer kissed many women at the party.

(07)

C
his
wife
just
$T_{[past]}$
find-out
that
D
Homer
$T_{[past]}$
kiss
many
women
at
the
party

(08) I couldn’t even count [how many women]$_1$ Homer kissed t_1 at the party.

(09)

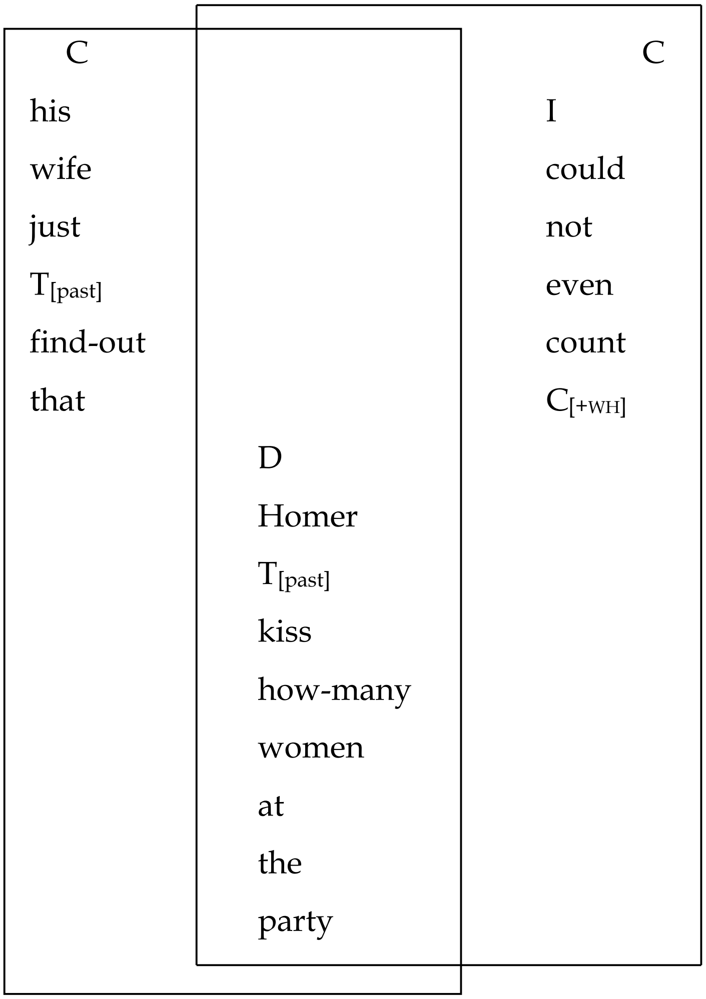
C
I
could
not
even
count
$\text{C}_{[+\text{WH}]}$
D
Homer
$\text{T}_{[\text{past}]}$
kiss
how-many
women
at
the
party

Consider, now, the more complex construction in (10), which, pre-theoretically speaking, can be seen as the result of some paratactic process that somehow collapses (07) and (09) together.

(10) His wife just found out that Homer kissed I couldn't even count how many women at the party.

Since nothing in Set Theory prevents two or more numerations from intersecting and sharing some lexical tokens, this option is in principle available.[82] In this dissertation, I will explore the hypothesis that the input to the syntactic computation(s) that generate(s) constructions like (10) is as shown in the Venn diagram in (11).

(11)



[82]. Regardless of the possibility of intersecting numerations, Set Theory also allows indefinitely many kinds of set-theoretical arrangements of lexical tokens that do not constitute inputs that the computational system can handle. Thus, we independently need to commit to an axiomatization that determines which of those sets count as legitimate numerations. In this context, ruling out intersecting numerations would require an additional axiom just for that purpose, which would be an unnecessary complication to the theory, unless there were strong empirical evidence for such a prohibition. Thus, a system that allows intersecting numerations is the null hypothesis; and I take the facts presented here as evidence for it.

The claim is that such intersections allow local computations to interfere with one another to some extent, with paratactic effects emerging from syntax pushed to limit, as it will be shown in detail in **§V**.

Finally, there is one issue about the idea of inputs as intersecting numerations which deserves further comment. The claim being made here is that whenever there is an intersection between two or more numerations, the syntactic computations that combine the lexical tokens of each numeration will necessarily be integrated into a unified larger computation.

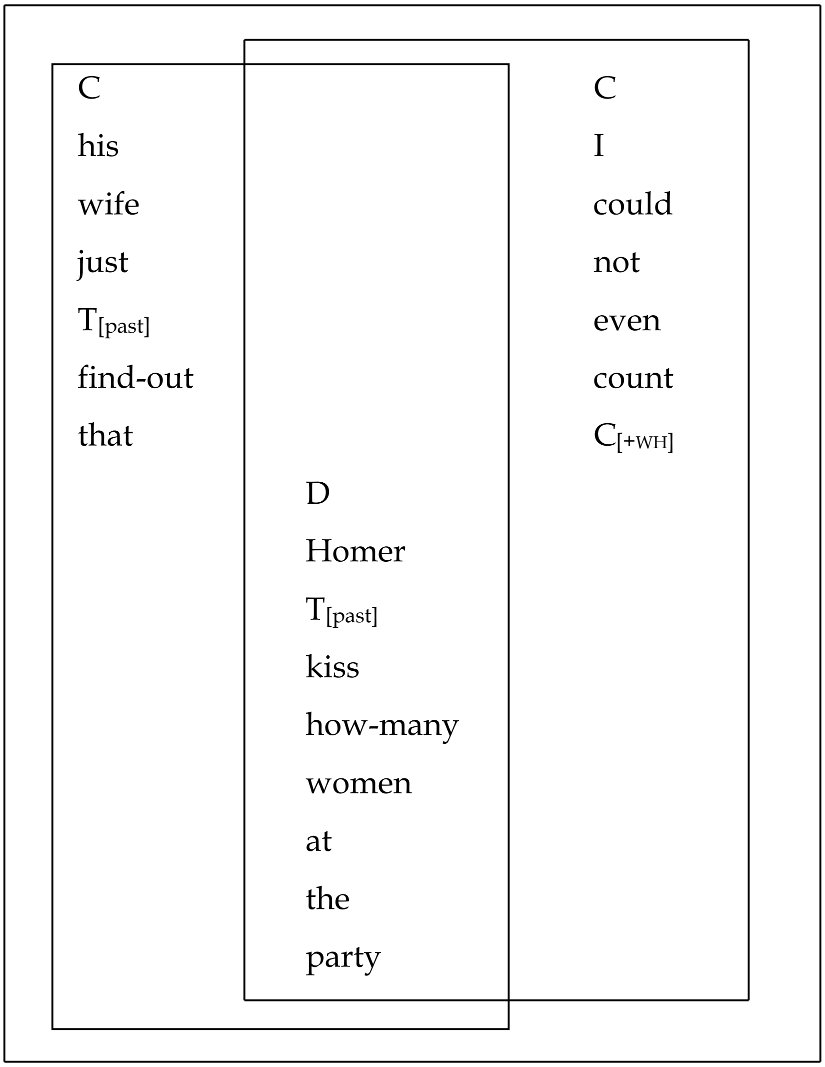
That does not follow from set theory alone. In principle, nothing seems to prevent the computational system from focusing on only one numeration and simply ignoring the other one(s), despite the intersection.[83]

One way out of the problem is to redefine what counts as an input to a syntactic derivation. Instead of a numeration (i.e. a set of lexical tokens), the input can be defined as a 'super-numeration', i.e. a set of sets of lexical tokens. That way, there would be a single formal object containing all the numerations that are supposed to 'go together', and nothing else. This would guarantee that the computations corresponding to each numeration would be treated as subcomputations of a larger computation.

From that perspective, (11) would be revised as in (12). For consistency, the same concept would apply to simpler constructions involving only one numeration, as in (13), which is the revised version of (07).

[83] Maybe this is not a problem. Depending on which items are in that numeration, the derivation would, in the best case, produce an ordinary sentence instead of a syntactic amalgam. In the worst case, the derivation would crash or terminate prematurely.

(12)


C
his
wife
just
$T_{[past]}$
find-out
that
C
I
could
not
even
count
$C_{[+WH]}$
D
Homer
$T_{[past]}$
kiss
how-many
women
at
the
party

(13)

C
his
wife
just
$\mathrm{T}_{\mathrm{[past]}}$
find-out
that
D
Homer
$\mathrm{T}_{\mathrm{[past]}}$
kiss
many
women
at
the
party

In the rest of this dissertation, I will keep using the simpler notation, as in (07) and (11), for expository reasons.

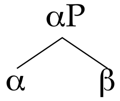
IV.2. Structure Building and Structure Preservation

In most variations of mainstream Minimalism, it is assumed that syntactic representations are built derivationally, through recursive applications of the operation merge, conceived as in (14) below.

(14) *Merge*[84]

a: input: α & β (such that both α and β are syntactic objects)

b: output:



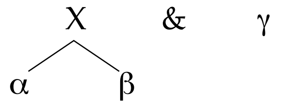
It has been assumed, without much discussion, that both inputs to Merge (i.e. α and β above) must be 'root nodes' of independent subtrees by definition, so that trees always grow on their outer edges.

This has been explicitly stated as the *Extension Condition*, which basically requires that, at any given derivational step ***t***, only constituents that are root nodes (i.e. maximal projections in the relational, bare-phrase-structure sense) can undergo merge in ***t***.

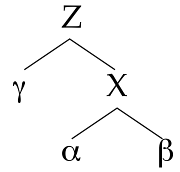
Therefore, abstracting away from linear order at PF, if (15a) is the input, the output must be (15b), not (15c).

[84] Formally speaking:
(i) input: α & β
output: K = {L, {α, β}}, such that L is the label of K, which corresponds to the head to the element that projects (in this case, α)

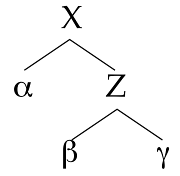
(15) a: input:



b: output:



c: * output:



In (15b), γ remains outside of X in the output. The new constituent Z that is created (and whose daughters are γ and X) is the new root node. Therefore, the internal structure of all constituents in the input is completely preserved in the output.

In (15c) γ is inserted inside X, as a new sister of β. The new constituent Z that is created (and whose daughters are γ and β) is not a root node. Rather, it is the new sister of α (which is no longer a sister of β). Therefore, the internal structure of X is not preserved from the input to the output. One of its daughters (i.e. β) is replaced with another one (i.e. Z). X was the root node in the input and it remains the root node in the output. Strictly speaking, what happens in (15c) is that γ merges with β, not with X.[85] Richards (1998) refers to this second type of Merge as *Tucking-in*.

[85] Thus, it is a little misleading to describe the input to this operation as being X & γ. Rather, it is β & γ, such that β is a daughter of X.

Clearly, the operation in (15b) obeys the Extension Condition, whereas the operation in (15c) does not.

From a conceptual point of view, the Extension Condition is motivated on minimalist grounds. Given derivational economy, it is not surprising that the computational system always chooses to build structure in a monotonic fashion, so that the internal structure of every constituent built in previous derivational steps is fully preserved. New constituents are created, but no constituent is destroyed. The intuitive idea behind this is as follows: why bother building a constituent at one point if it will be destroyed later? Therefore, this monotonicity appears to be part of an optimal solution, as if the system was designed by a "super-engineer", to put it in Chomsky's (2000) metaphorical terms. The Extension Condition might be seen, then, as a mere instantiation of a general economy condition on derivations. If merge applies only at the root, then every bit of structure built at any point is guaranteed to be in the final output (= LF).[86]

Uriagereka (2002) refers to this general economy condition as 'Law of Conservation of Patterns'.

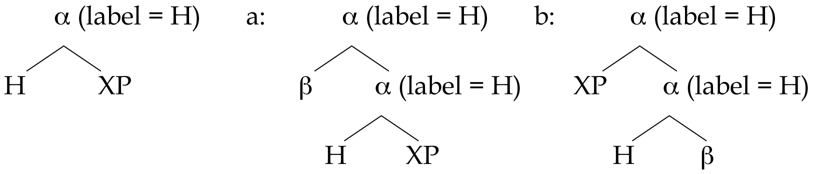
> It is hard to find examples of "loss of information", among other things because, on the average, linguistic processes are highly conservative. familiar constraints on recoverability of deletion operations, or what Chomsky calls the "inclusiveness" of derivations (...), can obviously be expressed in terms of some Law of the Conservation of Patterns (...). The same law, however, normally prevents us from teasing apart a computational and a representational approach.
>
> (Uriagereka 2002: 14)

[86] For a precise formalization of this argument, see Watanabe (1995),

However, the issue is not uncontroversial. Although intuitively appealing, a 'conservation law' of this kind is not *a priori* required by derivational systems, as a matter of logic. Changing structure is something that only derivational systems can do. Therefore, it is worth exploring a system that allows merge operations like in (15c). This is justified on methodological grounds, as a potential way to conclude something about the 'representationalism *versus* derivationalism' dilemma (as already hinted in Uriagereka's (2002) quote above).

In this regard, Chomsky (2000: 136) wrote:

> The new object K formed by Merge of β to α retains the label L of α, which projects. There are two reasonable possibilities, illustrating the ambiguity of cyclicity (...):
>
> (i) a: α is unchanged;
> b: β is as close to α as possible.
>
> Suppose we have the L[exical] I[tem] H with selectional feature F, and XP satisfying F. Then first Merge yields α = {XP, H}, with label H. Suppose we proceed to second Merge, merging β to α. In this case β is either extracted from XP (Move) or is a distinct syntactic object (pure Merge). There are two possible outcomes, depending on choice of K in (ii).
>
> (...)
>
> (ii)



> The desired outcome is (ii-a), not (ii-b); that has always been assumed without discussion. (...)
>
> But the reasons are not entirely obvious. Each outcome satisfies a reasonable condition: [(ii-a)] satisfies the familiar *Extension Condition* (ii-a); [(ii-b)] satisfies the condition of *Local Merge* [(ii-b)].
>
> One possibility is to stipulate that the *Extension Condition* always holds: operations preserve existing structure. Weaker assumptions suffice to bar (ii-b) but still allow Local Merge under other conditions.

Moreover, many recent minimalist works based on bottom-up derivations defend the necessity of certain grammatical mechanisms that, in one way or another, involve massive overwriting and changing in constituency relations along the derivational history, such as non-cyclic merge (Richards 1998; Castillo & Uriagereka 2000) and movement by lowering (Boskovic & Takahashi 1998), all of which would involve some variant of (15c). The common feature of all these approaches is that the Extension Condition should be relaxed, as suggested by Chomsky (1998) in the quote above.

I will not dispute the empirical advantages systems that allow tucking-in.[87] Actually, in the next section, I will go as far as endorsing the proposal that every instance of merge is, by definition, tucking-in.

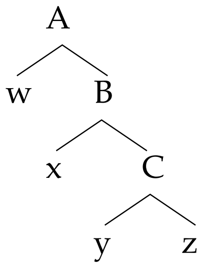
In what follows, I claim that 'merge at the root' (cf. 15b) and 'tucking in' (cf. 15c) are formally too different to be just two possible instantiations of the same structure building operation.

[87] One good example of the empirical and conceptual benefit of incorporating tucking-in into the syntactic machinery is found in the work by Castillo & Uriagereka (2000) on successive cyclicity. The phenomenon of long distance WH-movement defies the current minimalist desideratum of *last resort*, to the extent that it requires the stipulation of an *ad hoc* feature in the intermediate COMP whose only purpose is to trigger the very movement it tries to explain. By allowing tucking-in, the authors are able to straightforwardly reduce long distance movement to local movement, reconciling the Tree Adjoining Grammar approach to successive cyclic movement (cf. Frank 2002 and references therein) with the Minimalist framework. Basically, WH-movement happens in a strictly local fashion, and then the whole higher clause is built afterwards, by tucking-in lexical tokens one by one, as shown in (i):

(i) a: [^{CP} [^{IP} Mary [^{VP} loves who]]]
b: [^{CP} who$_1$ [^{IP} Mary [^{VP} loves t$_1$]]]
c: [^{CP} who$_1$ [^{CP} that [^{IP} Mary [^{VP} loves t$_1$]]]]
d: [^{CP} who$_1$ [^{VP} thinks [^{CP} that [^{IP} Mary [^{VP} loves t$_1$]]]]]
e: [^{CP} who$_1$ [^{IP} John [^{VP} thinks [^{CP} that [^{IP} Mary [^{VP} loves t$_1$]]]]]]
f: [^{VP} wonder [^{CP} who$_1$ [^{IP} John [^{VP} thinks [^{CP} that [^{IP} Mary [^{VP} loves t$_1$]]]]]]]
g: [^{IP} I [^{VP} wonder [^{CP} who$_1$ [^{IP} John [^{VP} thinks [^{CP} that [^{IP} Mary [^{VP} loves t$_1$]]]]]]]]
h: [^{CP} [^{IP} I [^{VP} wonder [^{CP} who$_1$ [^{IP} John [^{VP} thinks [^{CP} that [^{IP} Mary [^{VP} loves t$_1$]]]]]]]]]

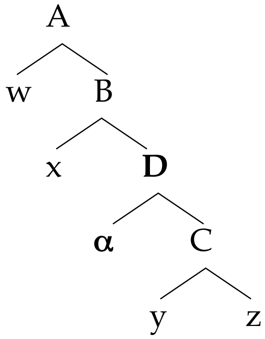
Consider the instance of tucking-in in (16), and let us, then, scrutinize its inner workings.

(16) a: input



& α

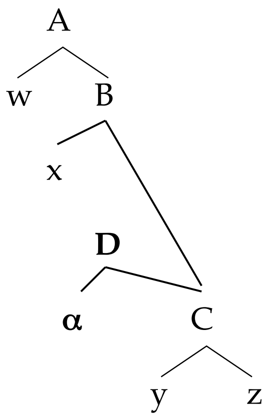
b: output:



Apparently, what happens in (16) is that the internal structure of B changes, whereas all other nodes remain unaffected. In (16a), the daughters of B are a and C. In (16b), C ceases to be a daughter of D, and becomes a daughter of the new constituent D, which is the new daughter of B, replacing C.

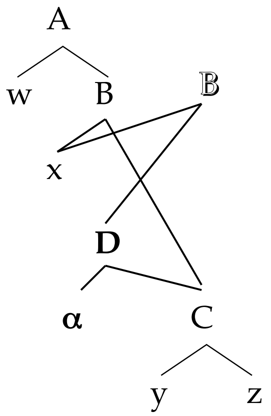
But what does that amount to, formally speaking? If the external element α is merely merging to C, then the output should simply be as in (17a).

(17) a:



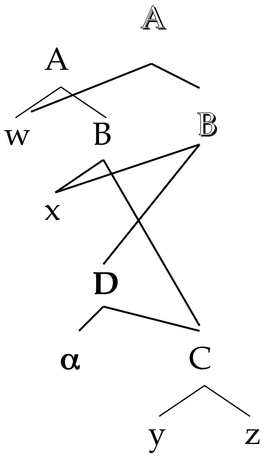
This is obviously not the same as (16b). Among many other things, (16b) 7differs from (17a) to extent that x and D are sisters in (16b) but not in (17a). Therefore, aside from the mere merge of α and C, one extra step towards the representation in (16b) would be merging x and the new constituent D, so that they become sisters, as shown in (17b).

(17) b:



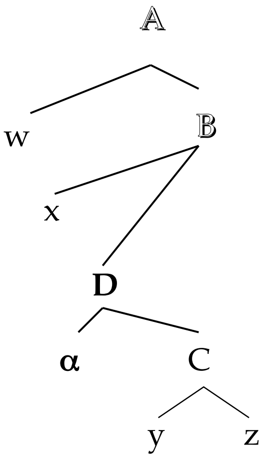
Once this is done, a new ‘incarnation’ of B is created in parallel to the already existing B. Notice that, in (17b), w is the sister of the old incarnation of B, the one that still has C as a daughter. However, in the target structure, w is the sister of the new B, the one that has D as a daughter. Therefore, yet one more merge operation is necessary. The new B and w merge, creating a new incarnation of A, as shown in (17c).

(17) c:



This is not yet the target structure. In order for the desired configuration to obtain, the old constituents that have been 'cloned' during the derivation must somehow be eliminated. Whatever the elimination procedure is, the result is that the old incarnations of A and B (and their respective motherhood relations) disappear from the structure, finally yielding (17d), which coincides with (16b) above.

(17) d:



In conclusion, tucking-in is not simply 'merge at a non-toot node'. It is a complex combination of applications of merge coupled with some structure elimination mechanism. And the deeper in the phrase marker that tucking-in occurs, the more rebuilding will be involved to achieve the desired target representations, as more and more nodes will have to be 'cloned' or eliminated in the process.

Another possibility is that tucking-in is a completely different operation to begin with, where all parts of the inner-workings described above come together as an automaton. This is the approach that I will take in the following sections. The question, then, is whether both structure building procedures exist in the computational system's toolbox, or only one of them. Needless to say, *ceteris paribus*, Occam's Razor would lead us to a theory where only one of these two possibilities exist. But this is, in the end, an empirical matter. In this dissertation, I commit to the view that every structure building operation is tucking-in, by definition. Thus, I claim that there is no such thing as an *Extension Condition* in the grammar. To the contrary, the system always builds new structure by partially destroying old structure (in a very constrained way). Consequently, constituency is heavily dynamic. What is a constituent at one derivational step may or may not be a constituent at subsequent steps. The technical details of such system will be presented in the next sections. In **§V**, the explanatory adequacy of such system (vis-à-vis the empirical facts given in **§II**) will be shown.

IV.3. Structure Building and the Directionality of Derivations

IV.3.1. Derivationalism *versus* Representationalism

Mainstream minimalism assumes the syntactic component of UG to be a derivational system, which builds syntactic structure step-by-step, from the bottom upwards, as in (18).

(18) a: [^{Z} d e]

b: [^{X} c [^{Z} d e]]

c: [^{Y} b [^{X} c [^{Z} d e]]]

d: [^{W} a [^{Y} b [^{X} c [^{Z} d e]]]]

From this perspective, the formal properties of phrases and sentences are taken to be effects of how syntactic structure is built. Therefore, what makes a given syntactic structure grammatical or ungrammatical is not much the structural properties of the final representation that obtains at the end of the derivation (e.g. (18d)). Rather, its entire derivational history must be in accordance with the principles of derivational economy (cf. Chomsky 1995, 2000; Collins 1997; Kitahara 1997; *inter alia*).[88/89]

[88] Chomsky (1995: section 9) makes a big deal out of the contrast in (i) and (ii), pointing out that the LF representation of (ii) is perfect, but it is ungrammatical because its derivational history involves a step which violates *Merge-Over-Move*, when the specifier of the embedded TP is filled with **[^{DP} a man]** instead of **[^{DP} there]**. The intuition is that Merge is just Merge, while Move is Copy+Merge+Delete; therefore, the system always prefers to merge **[^{DP} there]** at that derivational stage since it is the most economical strategy (less operations), eventually yielding (i). Strictly speaking, ungrammatical structures are not filtered or rejected at the interfaces. Rather, the

An alternative approach is to take syntax to be a representational system (Brody 1995, 1997, 1998; *inter alia*). Under that view, instead of the derivation in (18), all we have is the representation in (18d), generated in a single step. What determines whether it is grammatical or ungrammatical is a set of declarative rules that state which structural properties any given phrase marker must or must not have (e.g. binary branching, endocentricity, format of chains, etc.).

As Chomsky (2000: 98-99) points out, these two perspectives are very hard to tease apart. Arguments go in both directions, and, in most cases, analyses are fully intertranslatable from one framework to the other.

> The issue is reminiscent of old questions about morphological processes ("item-and-process" vs. "item-and-arrangement", etc.) and grammatical transformations. Thus, does a transformation map an input structure to an output structure, or is it an operation on the "output" that expresses properties of the "input"? It is unclear whether these are real questions; on the surface they look like the question whether $25 = 5^2$ or $5 = \sqrt{}25$. If the questions are real, they are subtle. (...) The apparent alternatives seem to be mostly intertranslatable, and it is not easy to tease out empirical differences, if any.

Cornell (1999) goes even further, and claims that these two apparently opposite approaches are two sides of the same coin, and must co-exist in any (transformational) theory of grammar.

economy principles prevent them from being generated in the first place. See Chomsky (1998, 1999) and Castillo, Drury & Grohmann (1999) for discussion of more complex examples of Merge-Over-Move.

(i) $[^{DP}$ there$]_1$ seems t_1 to be $[^{DP}$ a man$]$ in the room.
(ii) * $[^{DP}$ there$]$ seems $[^{DP}$ a man$]_1$ to be t_1 in the room.

[89] Uriagereka (1998, 1999) and Epstein, Groat, Kawashima & Kitahara (1998) are examples of radically derivational systems, with no levels of representations whatsoever (i.e. there is a phonological component and a semantic component, but no PF and LF levels of representations).

> A transformational grammar should have *both* a derivational and a representational interpretation, connected by soundness and completeness results.

Chomsky (2000: 99) even ends up admitting that his choice for a derivational approach is somewhat arbitrary.

> I will adopt the derivational approach as an expository device, though I suspect it may be more than that.

A new approach to this issue is offered by Phillips (1996, 2003). He proposes a derivational system that works in a top-to-bottom fashion,[90] rather

[90] Phillips himself never used the terminology 'top-to-bottom' to refer to this kind of system. He uses 'left-to-right' instead. The reason for it is that, in this framework, derivations do not work like the ones of classical transformational grammar (Chomsky 1955 [1975], 1957, 1965; *inter alia*), where non-terminal nodes were considered substantive entities that exist independently from the terminals they end up dominating; with the terminals being introduced after their dominating non-terminals (i.e. the whole being introduced before its parts). As opposed to that, Phillips-style derivations have terminals being introduced first (just like in *Bare Phrase Structure* (Chomsky 1994, 1995)), and non-terminals do not exist as primitive items of the 'alphabet of formatives', rather, they emerge as byproducts of merge operations on terminals (or, by recursion, on less complex non-terminals). On the basis of this, Phillips-style derivations are considered 'locally bottom-up' by Drury (1998a, 1998b), who prefers the terminology 'root-first derivations', also adopted by Richards (1999, 2002).

Phillips (2002) expressed his concern with this terminological issue as follows: *"A 'top-down' parser is a parser that begins with a root node, such as 'S', and then expands the root node by projecting its daughters, and then projects the daughters of each of those nodes, and so on until it arrives at the terminal nodes. A 'bottom-up' parser works in the opposite direction, starting with the terminals and ending at the root node. This terminology is well established in the computational linguistics literature. Notice that neither of these systems is subject to any linear order constraints; the only constraint is that they build structure either from top-to-bottom or from bottom-to-top. A number of authors have described incremental left-to-right systems of the kind that I have proposed as 'top-down' systems (...) This is unfortunate, since a left-to-right system is not a top-down system in the normal sense."*

I myself have used the terminology 'top-down' before, when referring to Phillips-style derivational systems (cf. Guimarães 1999, 2001), and I accept Phillips' (2002) criticism on that. However, I disfavor the term 'left to right' in the context of this research, since it masks the role played by 'tucking in' in the system, which essentially makes the tree undergo endogenous growth (with new branches emerging from the inside). Instead of 'top down', I here adopt the term 'top-to-bottom', since the later is not loaded (i.e. not associated with the concatenation-algebra re-writing formalism traditionally referred to 'top down'), and — unlike 'left-to-right'— it transparently expresses the idea that what is higher in the phrase marker accesses the derivation before what is lower.

than from the bottom upwards. Once the directionality of derivation is reversed, we start making predictions that, *ceteris paribus*, no representational approach can make. Moreover, these predictions seem to be by and large consistent with the facts, confirming Chomsky's suspicion that the derivational approach is more than just an expository device.

This idea has been explored, in different ways, by Drury (1998a, 1998b), Richards (1999, 2002), Schneider (1999), Terada (1999), Heider (2000), and Guimaraes (1999a, 1999b, 2001, 2003b, 2003c).

From that perspective, we would have the derivation in (19) instead of the one in (18).

(19) a: [^{W} a b]
b: [^{W} a [^{X} b c]]
c: [^{W} a [^{X} b [^{Y} c d]]]
d: [^{W} a [^{X} b [^{Y} c [^{Z} d e]]]]

This is what I will call the *Generalized Tucking-in* approach to structure building. From that perspective, constituency is partially destroyed <u>at every derivational step</u>.[91] For instance, **[^{W} a b]** is a constituent at step (19a); and, at step (19b) the construction of the constituents **[^{X} b c]** and **[^{W} a [^{X} b c]]** ends up destroying **[^{W} a b]**.

[91] In the example in (19), all non-terminals have their internal constituency changed at every step. This does not happen to specifiers, which never get destroyed, as will be shown shortly.

As mentioned in **§IV.2**, recent research has pointed out that there is some empirical evidence for tucking-in operations in syntax (cf. note 10). Moreover, as also discussed in **§IV.2**, one we scrutinize tucking-in in its the inner workings, it becomes clear that it is not merely 'merge at a non-root node', but rather a completely distinct structure building operation. In this context, it is a legitimate methodological move to hypothesize that tucking in is the only structure-building device of syntax, therefore pushing the idea of 'mutant constituency' to the limit to see which predictions are made when 'partial destruction' is taken to be inherent to the basic mechanisms of 'construction'. This is exactly what Phillips's (1996, 2003) program is all about.

One very powerful argument for a 'generalized tucking-in' derivational approach like (19) goes back to Phillips' (1996, 2003) work on conflicting constituency tests.

An old and embarrassing puzzle in Generative Grammar is the fact that, in some cases, multiple constituency tests give different and conflicting results when applied to the same sentence. For instance, take the sentence in (20).

(20) John gives candy to children in libraries on weekends.

As Phillips (1996: 24-25) shows, some tests, like negative polarity (cf. 21) and coordination (cf. 22) point to a right-branching VP structure, along the lines of the 'Larsonian shell' in (23).[92]

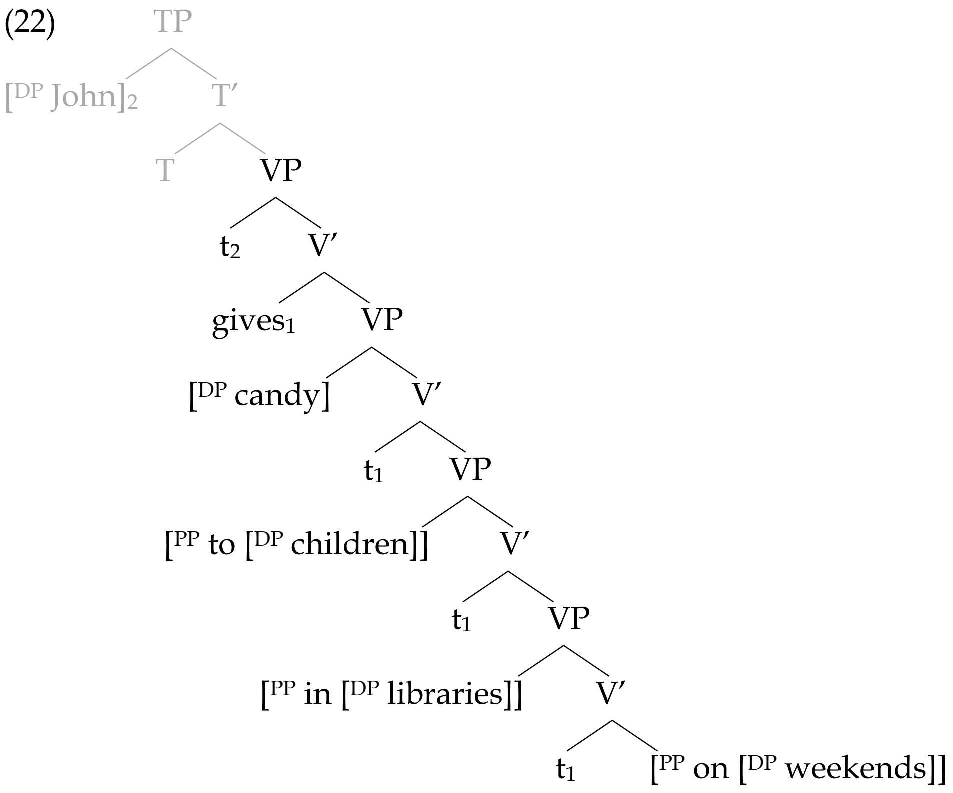
(21) a: John gave nothing to any of my children in the library on his birthday.

b: John gave candy to none of my children in any library on his birthday.

c: John gave candy to children in no library on his birthday.

d: * John gave candy to any of my children in no library on his birthday.

(22)



[92] This is based on the standard assumption that the licensing of a negative polarity requires c-command.

On the other hand, some movement tests, like VP-topicalization (cf. 23), point to a predominantly left-branching structure along the lines of (24).

(23) a: John intended to give candy to children in libraries on weekends...
and [give candy to children in libraries on weekends]$_1$ he did t_1 .

b: John intended to give candy to children in libraries...
and [give candy to children in libraries]$_1$ he did t_1 on weekends.

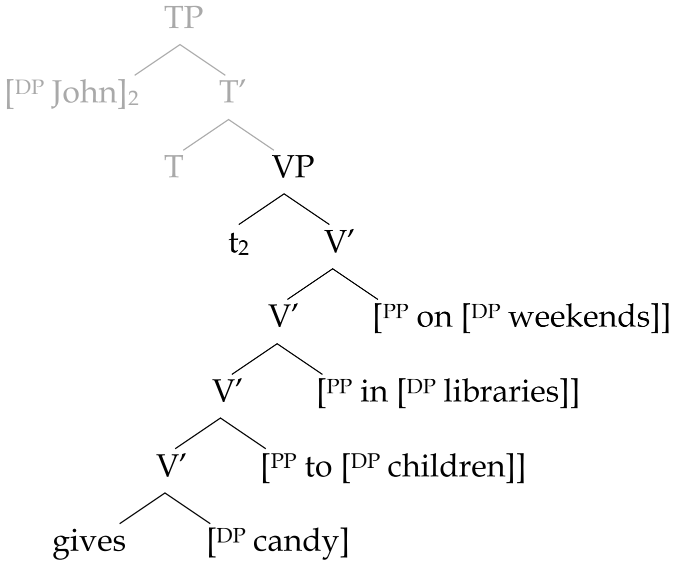
c: John intended to give candy to children...
and [give candy to children]$_1$ he did t_1 in libraries on weekends.

d: John intended to give candy...
and [give candy]$_1$ he did t_1 to children in libraries on weekends.

e: * and [to children in libraries]$_1$ he did t_1 give candy on weekends.

f: * and [in libraries on weekends]$_1$ he did t_1 give candy to children.

(24)



This conflict is even more accentuated in cases where the very same sentence exhibits symptoms of both left-branching and right-branching structure (cf. Pesetsky 1995: 230 *apud* Phillips 1996: 27). This is attested in (25), where the fronted fragments of VP require a structure like (24), whereas the binding relations between an NP inside the fronted fragment and an anaphor outside it require a structure like (22).

(25) a: ... and [give the books to **[them]**$_j$ in the garden]$_1$ he did t$_1$
on **[each other's]**$_j$ birthdays.
b: ... and [give the books to **[them]**$_j$]$_1$ he did t$_1$ in the garden
on **[each other's]**$_j$ birthdays.

This is, in principle, a serious paradox that defies many concepts behind the notion of constituency, which are the standard in Generative-Transformational Grammar for the treatment of long distance dependencies.

The conflict between (21-22) and (23-24) is the following. In order for the relevant c-command relations required to license NPIs in (21) to obtain, it must be the case that the substrings in (26a-d) are constituents, whereas the ones in (27a-c) are not. On the other hand, in order for the relevant movement operations to take place in (23), it must be the case that the substrings in (26a-c) are not constituents, whereas the ones in (27a-d) are.

(26) a: on weekends

b: in libraries on weekends

c: to children in libraries on weekends

d: gives candy to children in libraries on weekends

(27) a: gives candy

b: gives candy to children

c: gives candy to children in libraries

d: gives candy to children in libraries on weekends

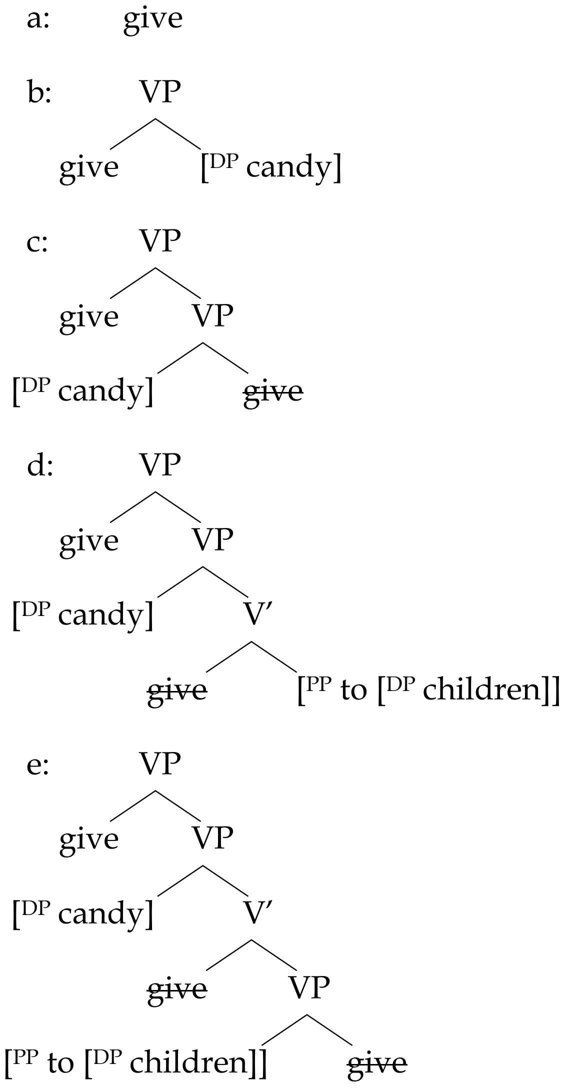
From a representational perspective, these two possibilities are mutually exclusive. The same holds for derivational systems where structure-building is fully conservative, as in (18) above.

Phillips' (1996, 2003) great insight is that such paradox completely disappears once we take all those structures above as to be generated in what I call the 'generalized tucking-in' fashion, along the lines of (19) above. In such a system, it is possible to 'have the cake and eat it too', having both (22-25) and (24-26) in the same derivation, which is absolutely crucial to account for cases like (25).

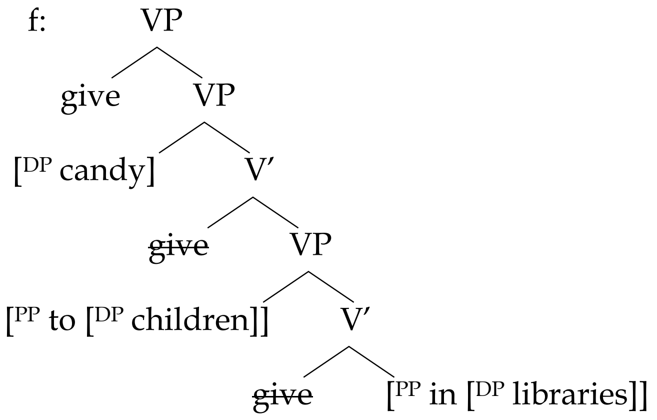
In derivations of the sort sketched in (19), constituency is dynamic. A given substring can be a constituent at a given derivational step t (and, hence, undergo some transformation), and later, at a derivational step t+n, that constituent may be destroyed, so that one of its daughters forms a constituent with together with another chunk of structure, with consequences for some other grammatical process that applies at that stage.

With regards to the specific problem above, Phillips' (1996, 2003) solution to the paradox above can be summarized as follows. Abstracting away from the VP-internal subject position, the VP of such constructions would be derived along the lines sketched in (28).

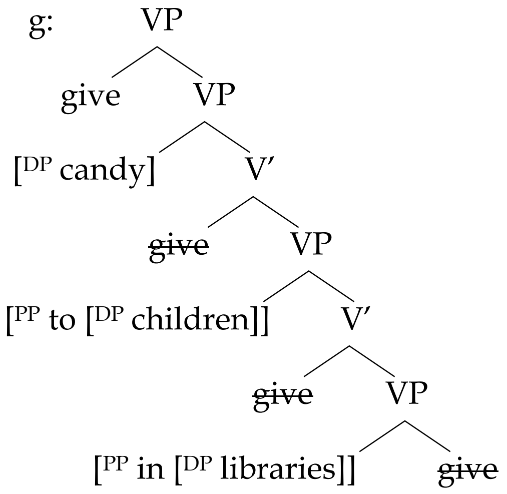
(28)



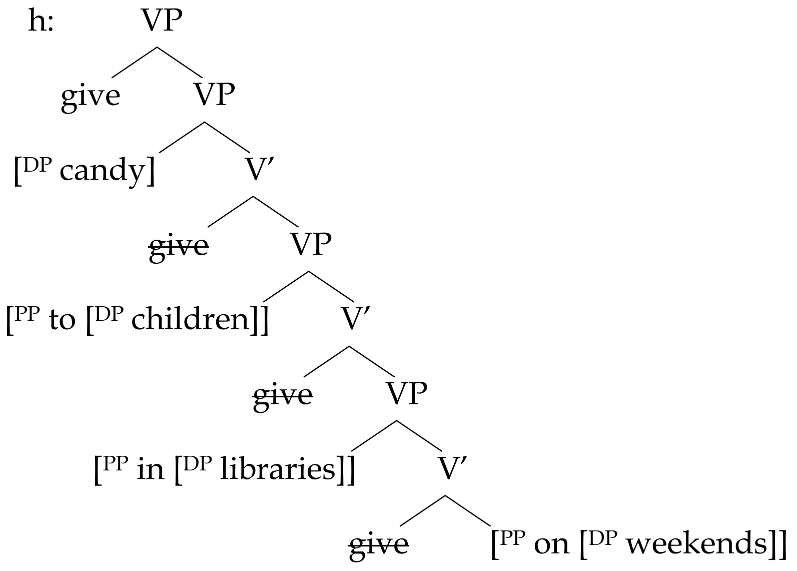
f:


VP
give
VP
[DP candy]
V′
~~give~~
VP
[PP to [DP children]]
V′
~~give~~
[PP in [DP libraries]]

g:

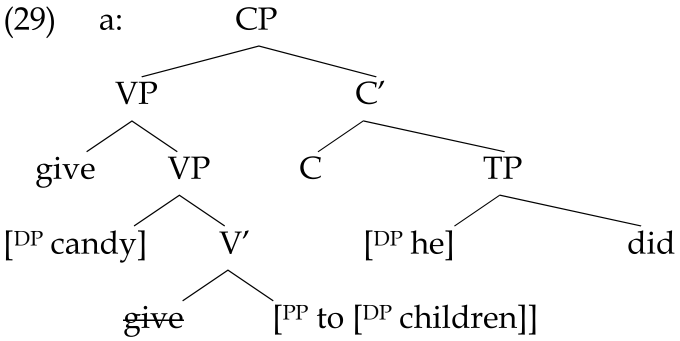

VP
give
VP
[DP candy]
V′
~~give~~
VP
[PP to [DP children]]
V′
~~give~~
VP
[PP in [DP libraries]]
~~give~~

h:

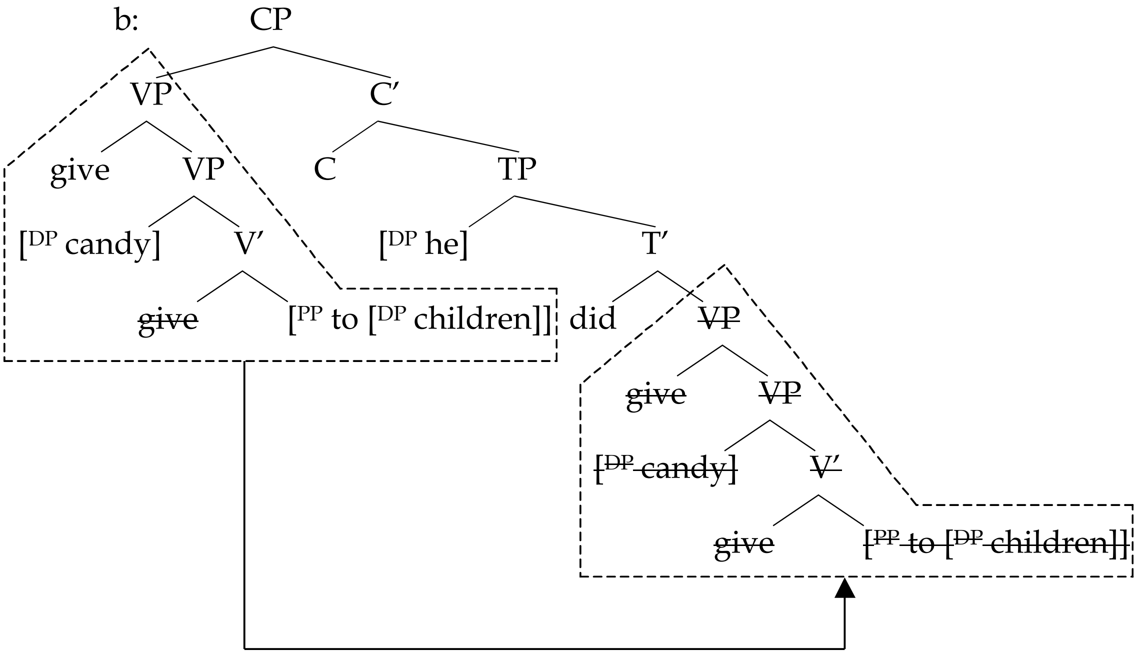

VP
give
VP
[DP candy]
V′
~~give~~
VP
[PP to [DP children]]
V′
~~give~~
VP
[PP in [DP libraries]]
V′
~~give~~
[PP on [DP weekends]]

Since, in such kind of system, derivations proceed essentially from top to bottom, and from left to right, any fronted VP would be generated in surface position (e.g. spec/CP, spec/TopP) and subsequently lowered to its 'D-structure position', so to speak, as sketched in (29).[93]

(29) a:

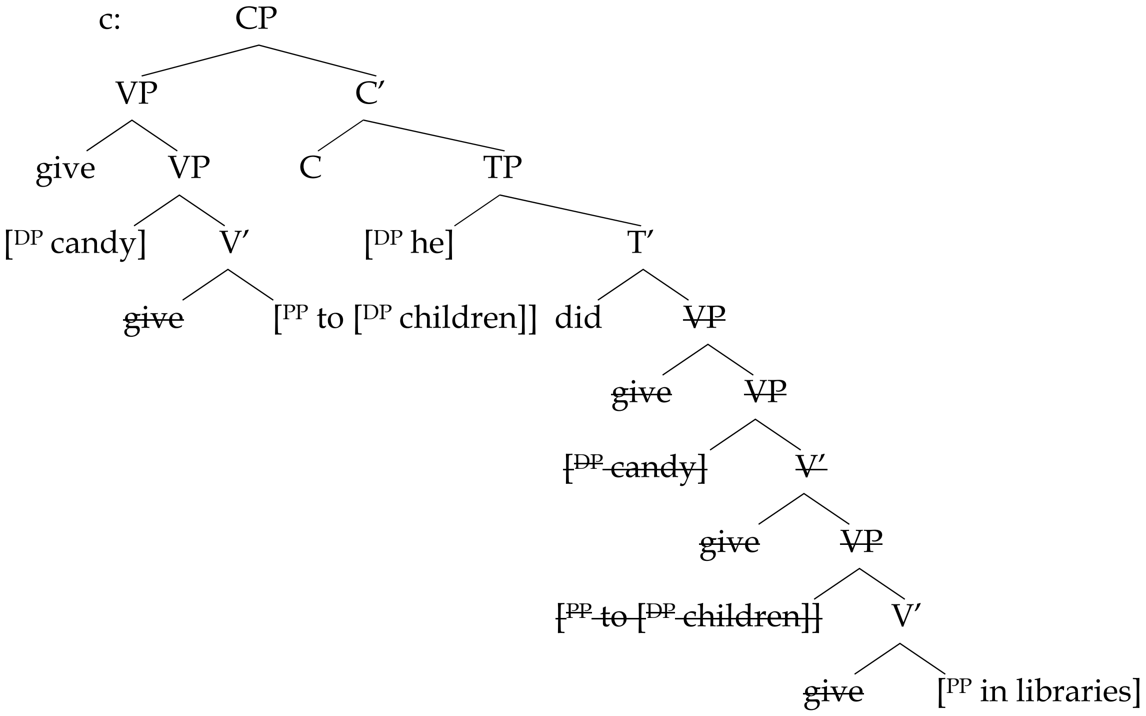


b:

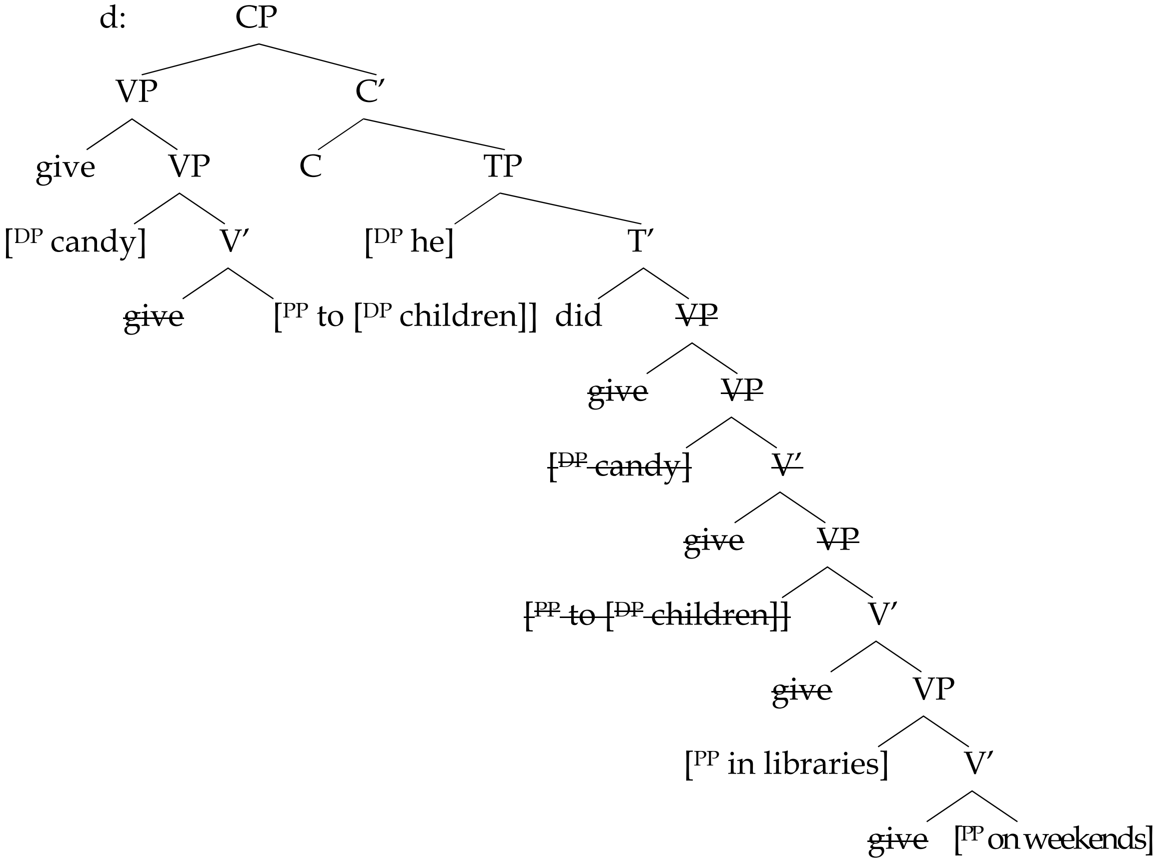


[93] According to Phillips (1996, 2003), such lowering involves making a silent copy of the relevant element, and then merging that silent copy in the lower position of the chain. This assumption will be revised in the next subsection.

c:


CP
VP
C′
give
VP
C
TP
[DP candy]
V′
[DP he]
T′
~~give~~
[PP to [DP children]]
did
~~VP~~
~~give~~
~~VP~~
~~[DP candy]~~
~~V′~~
~~give~~
~~VP~~
~~[PP to [DP children]]~~
V′
~~give~~
[PP in libraries]

d:


CP
VP
C′
give
VP
C
TP
[DP candy]
V′
[DP he]
T′
~~give~~
[PP to [DP children]]
did
~~VP~~
~~give~~
~~VP~~
~~[DP candy]~~
~~V′~~
~~give~~
~~VP~~
~~[PP to [DP children]]~~
V′
~~give~~
VP
[PP in libraries]
V′
~~give~~
[PP on weekends]

What happens in (29) is that, at the point where VP-topicalization takes place, **give candy to children** is a constituent (as required by the movement-based test). However, by the end of the derivation, that same substring no longer a constituent (as required by the c-command-based test). Basically, the construction of the VP begins in spec/CP, and gets interrupted at some point. After the chain is formed, the construction of the VP continues.

In a nutshell, the diagnostics from any movement-related tests is an accurate snapshot of an early derivational stage, whereas the diagnostics from any movement-related tests is an accurate snapshot of a late derivational stage.

This is essentially the derivational mechanics that I will adopt in this dissertation for the treatment of syntactic amalgamation. The next section concerns the technical details of its implementation.

Before moving on to the technical details, however, it is worth emphasizing that the 'generalized tucking-in' mechanics adopted here is actually not as much 'destructive' as it seems to be at first blush. Although monotonicity does not hold in its strongest form in these systems (because some constituency is destroyed in the course of the derivation), there is a weaker sense in which the computation can still be seen as monotonic[94].

If we focus on the core grammatical relations encoded in the phrase markers, rather than on the constituency integrity, we can see that new relations are established incrementally in the course of the derivation without eliminating

[94] This idea of considering monotonicity with respect to syntactic relations (rather than to the integrity of phrase geometry) is inspired on Weinberg's (1995) work on parsing.

any of the previously established ones. This is true of both bottom-up and top-to-bottom derivations, as shown in (30) and (31), respectively.

(30)

constituency	precedence among terminals	asymmetric c-command	dominance
[^{Z} d e]	<d,e>		<Z,d>, <Z,e>
[^{X} c [^{Z} d e]]	<d,e>, <c,d>, <c,e>	<c,d>, <c,e>	<Z,d>, <Z,e>, <X,c>, <X,d>, <X,e>, <X,Z>,
[^{Y} b [^{X} c [^{Z} d e]]]	<d,e>, <c,d>, <c,e>, <b,c>, <b,d>, <b,e>	<c,d>, <c,e>, <b,c>, <b,d>, <b,e>, <b,Z>	<Z,d>, <Z,e>, <X,c>, <X,d>, <X,e>, <X,Z>, <Y,b>, <Y,X>, <Y,c>, <Y,Z>, <Y,d>, <Y,e>
[^{W} a [^{Y} b [^{X} c [^{Z} d e]]]]	<d,e>, <c,d>, <c,e>, <b,c>, <b,d>, <b,e>, <a,b>, <a,c>, <a,d>, <a,e>	<c,d>, <c,e>, <b,c>, <b,d>, <b,e>, <b,Z>, <a,b>, <a,X>, <a,c>, <a,Z>, <a,d>, <a,e>	<Z,d>, <Z,e>, <X,c>, <X,d>, <X,e>, <X,Z>, <Y,b>, <Y,X>, <Y,c>, <Y,Z>, <Y,d>, <Y,e>, <W,a>, <W,Y>, <W,b>, <W,X>, <W,c>, <W,Z>, <W,d>, <W,e>

(31)

constituency	precedence among terminals	asymmetric c-command	dominance[95]
[^{W} a b]	<a,b>		<W,a>, <W,b>
[^{W} a [^{X} b c]]	<a,b>, <a,c>, <b,c>	<a,b>, <a,c>	<W,a>, <W,b>, <W,X>, <W,c>, <X,b>, <X,c>
[^{W} a [^{X} b [^{Y} c d]]]	<a,b>, <a,c>, <b,c>, <a,d>, <b,d>, <c,d>	<a,b>, <a,c>, <b,c>, <b,d>, <a,Y>, <a,d>	<W,a>, <W,b>, <W,X>, <W,c>, <X,b>, <X,c>, <W,d>, <W,Y>, <X,d>, <X,Y>, <Y,c>, <Y,d>
[^{W} a [^{X} b [^{Y} c [^{Z} d e]]]]	<a,b>, <a,c>, <b,c>, <a,d>, <b,d>, <c,d>, <a,e>, <b,e>, <c,e>, <d,e>	<a,b>, <a,c>, <b,c>, <b,d>, <a,Y>, <a,d>, <c,d>, <c,e>, <b,Z>, <b,e>, <a,Z>, <a,e>	<W,a>, <W,b>, <W,X>, <W,c>, <X,b>, <X,c>, <W,d>, <W,Y>, <X,d>, <X,Y>, <Y,c>, <Y,d>, <W,e>, <W,Z>, <X,e>, <X,Z>, <Y,e>, <Y,Z>, <Z,d>, <Z,e>

IV.3.2. Merge

The most basic substantive notion involved in the combinatorial system proposed here is the syntactic atom, defined as in (32).

[95] In order for dominance to be taken as monotonic in a top-down system, we must first work out some details, and define syntactic objects in a more flexible and intuitive way, such that, for example, **[^{W} a b]**, **[^{W} a [^{X} b c]]**, **[^{W} a [^{X} b [^{Y} c d]]]**, and **[^{W} a [^{X} b [^{Y} c [^{Z} d e]]]]** would all be taken as successive 'reincarnations' of the same phrase, since they all have the same label.

(32) *Syntactic Atom***:**

A syntactic atom is a lexical token, which is formed by a π-particle (relevant only to the phonological component), a λ-particle (relevant only to the semantic component) somehow linked to each other.

For instance, **drum = *drum* ↔#drum#**, where: (i) ***drum*** is the λ-particle of **drum**, *i.e.* its semantic material; (ii) **#drum#** is the π-particle of **drum**, *i.e.* its phonological material; and (iii) **↔** is whatever lexical device (substantive or formal) arbitrarily links ***drum*** and **#drum#** to each other.

In this system, phrases are taken to be organizations of λ-particles, rather than organizations of syntactic atoms.

(33) *Phrase*:

K is a phrase if and only if either (i) or (ii):

a: K is a λ-particle;

b: K = {L, {*x*, *y*}}, such that both ***x*** & ***y*** are phrases,
and L (the label of K) corresponds to the head of either ***x*** or ***y***.

Complex phrases are recursively built through the operation Merge, defined as in (34), where **[**^{A} ***x y*****]** is the already existing structure and ***z*** is the incoming element to be inserted within **[**^{A} ***x y*****]**.

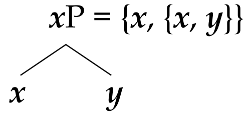
(34) *Merge*: (preliminary definition, to be refined later)

input: {A, {*x*, *y*}} & *z*

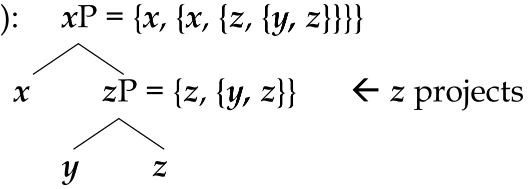
output: $\{A, \{x, \{B, \{y, z\}\}\}\}$

By the definition in (34), there is no constraint on the label of the new constituent formed inside the existing structure. In principle, either the new element being tucked-in projects (as in (35a)), or its sister does (as in (35b)).

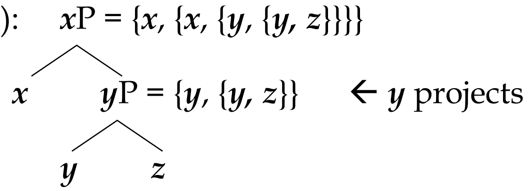
(35) input:



output (a):



output (b):

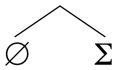


In order for this mechanics to work, we need some phrase already there in the derivational workspace in order to introduce the (λ-particles of the) first two syntactic atoms selected from the numeration. This is so because the input is explicitly defined as including a branching 'host phrase', such that one of its daughters becomes the sister of the incoming element in the output.

I assume, then, that the system has a starting axiom, which consists of an operation that applies before all others, introducing the phrase in (36) in the derivational workspace.

(36) *Starting Axiom*:

$\Sigma P = \{\Sigma, \{\varnothing, \Sigma\}\}$



The phrase ΣP of my system is analogous to the node **S** of Chomsky (1955 [1975], 1957, 1965) and to the abstract terminal of Kayne (1994: 36-38). I take Σ to be an 'assertion terminal'. In fact, every time that a speaker says, for instance, *"the earth isn't flat"*, (s)he is not just saying that the earth is not flat. Rather, (s)he is asserting that (s)he believes that the earth not being flat is an actual fact about the real world[96]. In other words, (s)he is committing to the truth of the uttered sentence (see Echepare (1997) on this matter). My take is that this 'commitment' is syntactically represented/instantiated by ΣP[97]. The symbol ∅ stands for the empty set, which is there, as a sister of Σ, just to guarantee the appropriate syntactic configuration.[98] Occasionally, this starting axiom may be omitted from the notation for expository reasons, but I want the reader to keep in mind that it is always present, or else no derivation could start.

Nothing has been said yet about linear order. Given the definitions in (33) and (34), inputs and outputs of Merge are set-theoretical objects which do not

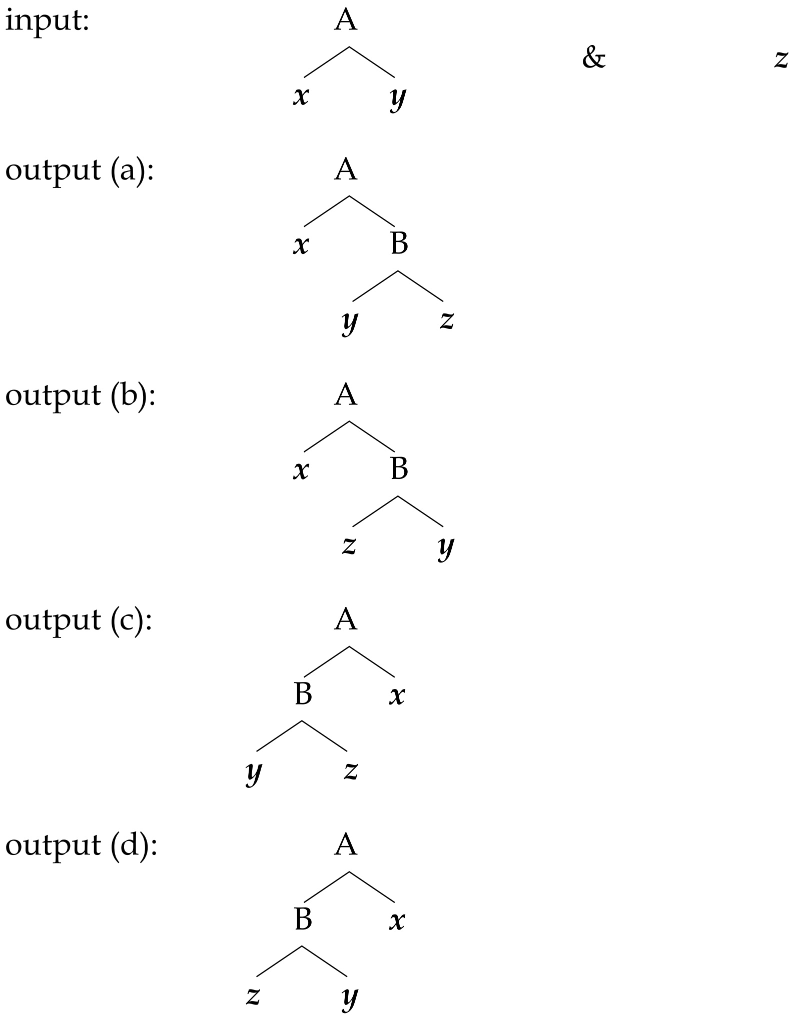
[96] Or, at least, (s)he wants the interlocutor to believe that.

[97] Of course, there is nothing (neither inside nor outside the system) requiring this 'commitment' to be syntactically represented. But it is also true that nothing in principle excludes this possibility.

[98] One may raise an objection to (36), pointing out that it does not count as a phrase in the technical sense of (33) above, given that ∅ is not, in principle, a λ-particle. There are two ways to go about this. Either we simply stipulate that ∅ is a λ-particle, or we leave (36) as it is, and take it to be the locus of 'Göedelian incompleteness' of this theory.

encode precedence relations, or even any other kind of asymmetry between sisters. Thus, in principle, (34) would be compatible with either logically possible ordering pattern in (37).

(37)



In Phillips' (1996, 2003) system, the mechanics of merge is constrained in such a way that only the output in (37a) is possible. This is achieved through the postulation of the two additional axioms below.

(38) *Merge Right*:

A new element can merge only with a (compatible) node that is at the right edge of the structure.

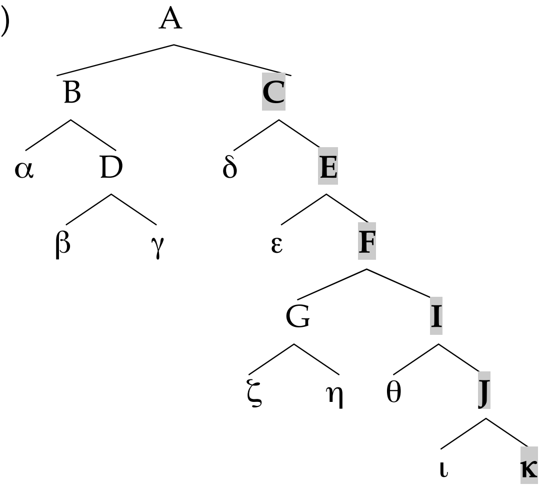
(39) *Branch Right*:

A new element must immediately follow the node it is merged with (i.e. its sister).

On the one hand, *Merge Right* forces that the constituent targeted as the sister of the new incoming element be one of the rightmost branches (as in (37a)), not any left-branch (as in (37c) and (37d)). *Branch Right*, on the other hand, forces that the new element be pronounced after its sister (as in (37a)), not before it (as in (37b) and (37d)).

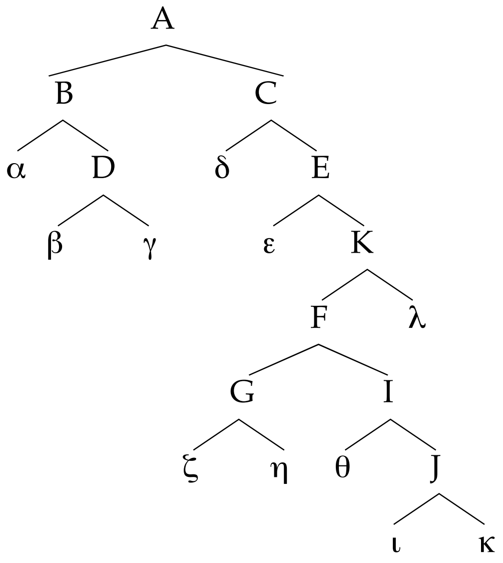
Consider a more complex case now. Suppose that the phrase marker currently in the derivational workspace is the one in (40). At this point, the lexical token **λ** is selected from the numeration. By *Merge Right*, the nodes **C**, **E**, **F**, **I** and **κ** are all and only the legitimate candidates for being the sister of **λ** in the next step. This is so because these are all and only the nodes at the right edge of the structure. Among these possibilities, the system will choose one that is compatible with being a sister of **λ** as far as convergence matters are concerned (i.e. thematic and feature-checking requirements). Even if **λ** can potentially be a sister of another node not at the right edge of the structure (e.g. D), such ‘merge left’ type of attachment is not allowed, as a constraint on derivations.

(40)

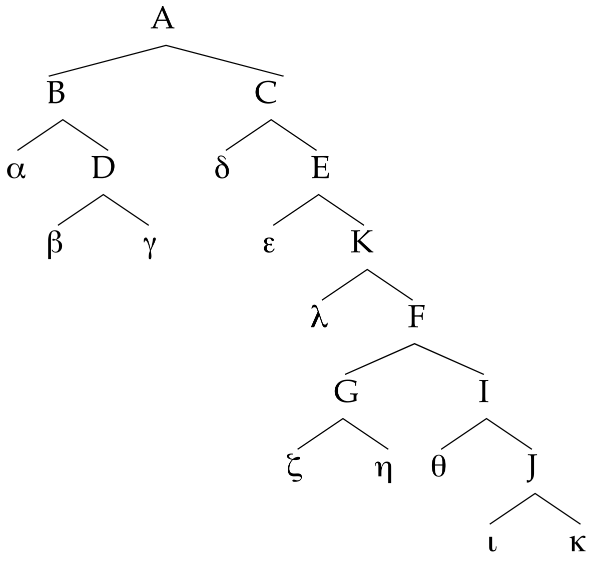


Suppose that the compatible node in this case is **F**. By *Branch Right*, the output must be as in (41a), rather than as in (41b).

(41) a:



b: *



Notice that both *Merge Right* and *Branch Right* make explicit reference to linear order relations in the phrase marker itself, which raises an important issue.

If precedence is not encoded in the syntactic representations — as I explicitly assume in (33)—, then it must be the case that linear order is established extrinsically, through some mapping function.

In Phillips' (1996, 2003) system, the linear order of each terminal node relatively to all the others is established upfront, at the very moment that it he lexical token corresponding to that terminal node is selected from the Numeration.[99] In a nutshell, the PF-string of terminal nodes is a direct reflex of the order in which lexical tokens access the derivational workspace. This is why Phillips labels his model 'Incremental Left-to-Right Syntax'.

Notice, however, that this leads to a serious 'alignment problem', as there is no way in which the computational system could possibly know which nodes are possible targets for merge, simply because the notion of 'rightmost' is not definable for the phrase-marker. Consequently, in the limit, any given string of terminals could correspond to pretty much any hierarchical structure, and vice-versa.

A way out of this problem would be to assume that precedence relations are indeed encoded in the phrase marker, which seems to be what Phillips (1996, 2003) tacitly assumes. From that perspective, the set-theoretical objects corresponding to phrases wouldn't be as in (42a), which encodes only one type of asymmetry between the sister nodes (i.e. which one 'projects' its categorical properties to the mother node). Rather, they would be something more or less along the lines of (42b), which encodes two types of asymmetry between sister nodes (i.e. which one 'projects' its categorical properties to the mother node, and which one (immediately) precedes the other).

[99] I am bringing the notion of Numeration into the picture for commensurability purposes. However, Phillips (1996, 2003) does not commit himself to Numerations.

(42) a: xP = {x, {*x*, *y*}}

b: xP = {x, <*x*, *y*>}

That way, notions such as ‘rightmost’ are definable within phrase markers, and structure-building operations can be sensitive to them, so that the ‘alignment problem’ goes away.

However, this technical solution relies on redundancy. Precedence relations are determined upfront, outside of the phrase marker, and then they are redundantly encoded in the phrase marker by the structure-building mechanism.

This ‘redundancy problem’ can be easily avoided if we assume that order and hierarchy are related through some mapping function that is external to the structure-building mechanism.

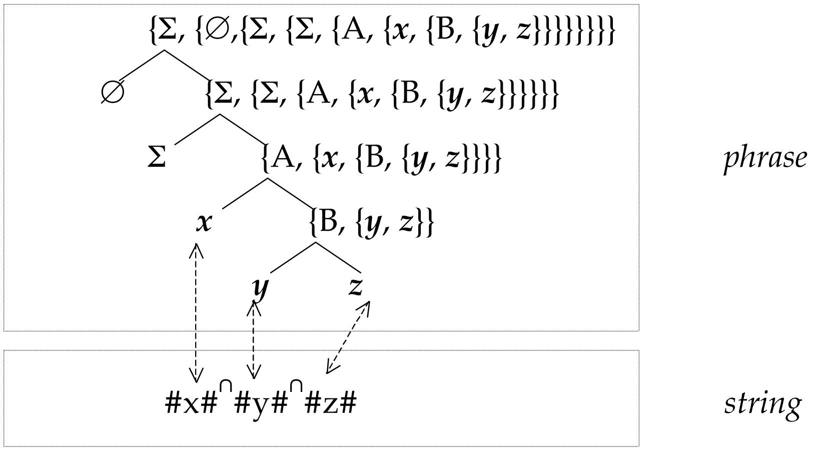
This can be implemented through some ‘Linearization’ mapping function from hierarchical structure to a string of terminals, as has been standardly assumed in mainstream Minimalism (based on some version of Kayne’s (1994) Linear Correspondence Axiom). Alternatively, we can conceive the reverse mapping function. That is, a string of terminals can be mapped to a hierarchical structure through some ‘Hierarchization’ procedure. It is the second possibility that I explore in this dissertation.

I assume that syntax generates two distinct kinds of syntactic objects in the same derivational workspace. On one hand, if the phonological component demands a string of sounds, then the syntactic component has to generate it. On the other hand, if the semantic component demands a hierarchical structure with part/whole relations, then the syntactic component has to generate that as well. In a sense, this is very similar to what we had in the old days of generative grammar, when every phrase structure rule was, by definition, the establishment of both hierarchical and precedence relations (*e.g.* VP → V$^\frown$NP)[100]. The difference is that, in the system I propose here, these two properties are factored into two parallel (sub)representations, one of each satisfying a distinct bare output condition.

I suggest the following way of conceiving these 2-dimensional syntactic objects in more formal terms. Given three syntactic atoms $\underline{\mathbf{x}}$, $\underline{\mathbf{y}}$ & $\underline{\mathbf{z}}$, such that their λ-particles are ***x***, ***y*** & ***z*** respectively; their π-particles are #x#, #y# & #z# respectively; and such that they have been introduced in the derivation in the following order: 1st = $\underline{\mathbf{x}}$, 2nd = $\underline{\mathbf{y}}$ & 3rd = $\underline{\mathbf{z}}$; the complete structure generated by the syntactic component for this small (sub)derivation is (43), where actual labels are not specified for expository reasons.

[100] That is, **V**$^\frown$**NP** "is a" **VP** (hierarchy), and **V** immediately precedes **NP** (order)..

(43)


{Σ, {∅,{Σ, {Σ, {A, {x, {B, {y, z}}}}}}}}
∅
{Σ, {Σ, {A, {x, {B, {y, z}}}}}}
Σ
{A, {x, {B, {y, z}}}}
x
{B, {y, z}}
y
z
phrase
#x#⁀#y#⁀#z#
string

But nothing has been said so far about how these phrases and strings are supposed to go together. I propose that what link them to each other is a version of the Linear Correspondence Axiom, here conceived not as a grammaticalization of a *bare output condition* in the spirit of Higginbotham (1983) and Chomsky (1994, 1995)[101], but as a constraint on the shape of phrase markers, in a way closer to Kayne's (1994) original proposal. In fact, I endorse Drury's (1998) assumption that precedence is not obtained from c-command. Rather, precedence is THE primitive relation of UG, and c-command is somehow parasitic on it.[102] For him (as well as for me), Kayne's (1994: 38) basic idea about

[101] Higginbotham's (1983) and Chomsky's (1995) idea is that the nature of the A-P performance system(s) demands that, for each sentence, all words must be temporally linearly ordered in order to be pronounceable. In Guimarães (1998: 54-55), I question that assumption, arguing that there is strong evidence (from radical phonetic co-articulation) that the A-P system can handle simultaneity. As a matter of logic, nothing would prevent the A-P system from taking an instruction to pronounce two or more words at once, and doing it by "calculating" the resultant forces for the combination of all movements required to pronounce all words together. If human language does not work like that, it is – in my view – because precedence is a grammatical primitive.

[102] Drury (1999) ends up classifying the derivations of top-down/left-to-right systems with these properties as "π-derivations", where π stands for precedence, [Colin] Phillips, and PF.

the relation between order and structure is better understood as the interaction between the axioms in (44) and (45).

(44) *Derivational Correspondence Axiom* (adapted from Drury (1998a/b))

Given any two syntactic atoms **<u>x</u>** & **<u>y</u>** (where **#x#** & **#y#** are their respective π-particles), if **<u>x</u>** accesses the derivation before **<u>y</u>**, then **#x#** phonetically precedes **#y#**.

(45) *Linear Correspondence Axiom*

Given any two syntactic atoms **<u>x</u>** & **<u>y</u>** (where **#x#** & **#y#** are their respective π-particles, and ***x*** & ***y*** are their respective λ-particles), if **#x#** precedes **#y#**, then it must be the case that ***x*** asymmetrically c-commands ***y***.

This is in accordance with Phillips's (1996, 2003) idea that derivational time equals real time, under the hypothesis that the parser and the grammar are the very same engine. Although I agree with Drury (1998) that c-command is parasitic on precedence (rather than the other way around), I do not endorse his proposal of defining c-command in terms of precedence. In this regard, I am more conservative and assume, that Kayne's (1994) LCA is, like the name says, a <u>correspondence</u> axiom (perhaps, motivated by parsing considerations), which requires that these two (independently definable) relations must 'go together' throughout the derivation, as in (45).

This formalism works fine for simple cases like (46), as shown in (47).

(46) a: [^{IP} *she*$_1$ [$^{I'}$ *was* [^{VP} *shot* t$_1$]]]

b: #she#$^\frown$#was#$^\frown$#shot#

(47)

PRECEDENCE	ASYMMETRIC C-COMMAND
#she# precedes **#was#**	*she* asymmetrically c-commands *was*
#she# precedes **#shot#**	*she* asymmetrically c-commands *shot*
#was# precedes **#shot#**	*was* asymmetrically c-commands *shot*

However, the LCA gets violated in structures with complex phrases at non-complement positions like (48), as shown in (49).[103]

(48) a: [^{IP} [^{DP} *the*$_1$ [^{NP} *man*]] [$^{I'}$ I [^{VP} *wonders* [^{CP} *if*

[^{IP} [^{DP} *the*$_2$ [^{NP} *woman*]]$_1$ [$^{I'}$ *was* [^{VP} *shot* t$_1$]]]]]]]

b: #the#$^\frown$#man#$^\frown$#wonders#$^\frown$#if#$^\frown$#the#$^\frown$#woman#$^\frown$#was#$^\frown$#shot#

[103] Here, I am abstracting away from 'the bottom-of-the-tree problem', which arises when the two lowest terminals of a (sub)tree are phonologically active. Since they mutually c-command each other, satisfaction of the LCA is impossible. For the sake of exposition, I am assuming two vacuous projections (i.e. [^{NP} *man*] and [^{NP} *woman*]), so that *the*$_1$ asymmetrically c-commands *man*, and *the*$_2$ asymmetrically c-commands *woman*. See Guimarães (2000) on the matter.

(49)

PRECEDENCE	ASYMMETRIC C-COMMAND
#the$_1$# precedes **#man#**	*the$_1$* asymmetrically c-commands *man*
#the$_1$# precedes **#wonders#**	* **no correspondence**
#the$_1$# precedes **#if#**	* **no correspondence**
#the$_1$# precedes **#the$_2$#**	* **no correspondence**
#the$_1$# precedes **#woman#**	* **no correspondence**
#the$_1$# precedes **#was#**	* **no correspondence**
#the$_1$# precedes **#shot#**	* **no correspondence**
#man# precedes **#wonders#**	* **no correspondence**
#man# precedes **#if#**	* **no correspondence**
#man# precedes **#the$_2$#**	* **no correspondence**
#man# precedes **#woman#**	* **no correspondence**
#man# precedes **#was#**	* **no correspondence**
#man# precedes **#shot#**	* **no correspondence**
#wonders# precedes **#if#**	*wonders* asymmetrically c-commands *if*
#wonders# precedes **#the$_2$#**	*wonders* asymmetrically c-commands *the$_2$*
#wonders# precedes **#woman#**	*wonders* asymmetrically c-commands *woman*
#wonders# precedes **#was#**	*wonders* asymmetrically c-commands *was*
#wonders# precedes **#shot#**	*wonders* asymmetrically c-commands *shot*
#if# precedes **#the$_2$#**	*if* asymmetrically c-commands *the$_2$*
#if# precedes **#woman#**	*if* asymmetrically c-commands *woman*
#if# precedes **#was#**	*if* asymmetrically c-commands *was*
#if# precedes **#shot#**	*if* asymmetrically c-commands *shot*
#the$_2$# precedes **#woman#**	*the$_2$* asymmetrically c-commands *woman*
#the$_2$# precedes **#was#**	* **no correspondence**
#the$_2$# precedes **#shot#**	* **no correspondence**
#woman# precedes **#was#**	* **no correspondence**
#woman# precedes **#shot#**	* **no correspondence**
#was# precedes **#shot#**	*was* asymmetrically c-commands *shot*

The bottom line is that the LCA gets violated every time a (phonologically active) new terminal is merged in a position not asymmetrically c-commanded by all (phonologically active) preceding terminals.

Since structures like (48a) do exist, the inevitable conclusion is that, in such cases, the grammar has to have extra device to satisfy the LCA incrementally. Moreover, minimalist assumptions force this extra device to be something that we already need for independent reasons. Such device is Spell-Out.

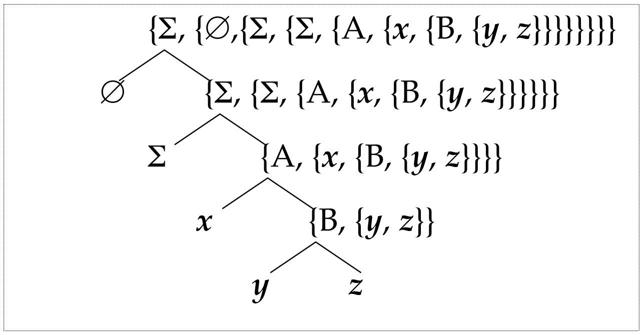
(50) **<u>Spell-Out</u>**:

Remove the current string of π-particles from the derivational syntactic workspace, and deliver it to the phonological component, for morphophonological and prosodic computation, and further pronunciation.

The task of Spell-Out, then, is to break the link (↔) between the λ-particle and the π-particle of all syntactic atoms present in the derivational workspace, removing all π-particles and delivering them to the phonological component, while leaving all λ-particles untouched, as well as the phrases formed by them[104]. Thus, if Spell-Out applies to the object we have in (43), then (51) obtains.

[104] Perhaps, this is what Chomsky (1995: 229) had in mind when he said that *"Spell-Out strips away from Σ* [*i.e.* the current syntactic structure, MG] *those elements relevant only to π* [*i.e.* the sound-related interface, MG]*, leaving the residue* Σ_L*, which is mapped to λ* [*i.e.* the meaning-related interface, MG] *by operations of the kind used to form Σ*".

(51)


{Σ, {∅,{Σ, {Σ, {A, {x, {B, {y, z}}}}}}}}
∅
{Σ, {Σ, {A, {x, {B, {y, z}}}}}}
Σ
{A, {x, {B, {y, z}}}}
x
{B, {y, z}}
y
z

🡐 **in the Syntax**

$\#x\#^\frown\#y\#^\frown\#z\#$

🡐 **out to Phonology**

Inspired by Drury (1998), I assume that the way top-down systems satisfy the LCA is by applying Spell-Out as many times as necessary, as first proposed by Uriagereka (1998, 1999) for bottom-up derivations. Every time the string of π-particles is removed from the syntactic derivational workspace and sent to the phonological component, there is no longer a problem with merging a new terminal in a position not asymmetrically c-commanded by all (phonologically active) preceding terminals. When this happens, the LCA is vacuously satisfied, since the π-particles of all (phonologically active) terminals preceding the new element are gone.

Given the minimalist *desiderata*, the null hypothesis is that the Spell-Out operation is restricted by economy. Therefore it can not apply freely. Rather, it is a last resort strategy (cf. Uriagereka 1999).

(52) *'Minimize Spell-Out' Corollary*:

Minimize the instances of Spell-Out as much as possible (*i.e.* do not apply Spell-Out unless it is strictly necessary for convergence).

Let us go back to the example under discussion to see how this formalism works. Starting from the top, the first phrase to be built is the subject of the main clause.

(53) a: $[^{\Sigma P}\ \varnothing\ [^{\Sigma'}\ \Sigma\ \textit{the}]]$

b: #the#

(54) a: $[^{\Sigma P}\ \varnothing\ [^{\Sigma'}\ \Sigma\ [^{DP}\ \textit{the}\ [^{NP}\ \textit{man}]]]]$

b: #the#$^{\cap}$#man#

Right at this point, the current string of π-particles (i.e. **#the#**$^{\cap}$**#man#**) has to be spelled-out, otherwise, a problem of lack of correspondence between precedence and c-command will arise as soon as the next phonologically active terminal accesses the derivation.

(55) a: ~~$[^{\Sigma P}\ \varnothing\ [^{\Sigma'}\ \Sigma\ [^{DP}\ \textit{the}\ [^{NP}\ \textit{man}]]]]$~~

b: $\varnothing$

Then the Infl head accesses the derivation, merging with **<u>the man</u>**. The LCA is vacuously satisfied at this point, as there is no phonologically active terminal in the derivational workspace.

(56) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ~~ [^{IP} ~~[^{DP} *the* [^{NP} *man*]]~~ I]~~]]~~

b: ∅

The system keeps building the rest of the structure, inserting new terminals step-by-step, from the top downwards, as follows:

(57) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ~~ [^{IP} ~~[^{DP} *the* [^{NP} *man*]]~~ [$^{I'}$ I *wonders*]]~~]]~~

b: #wonders#

(58) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ~~ [^{IP} ~~[^{DP} *the* [^{NP} *man*]]~~ [$^{I'}$ I [^{VP} *wonders if*]]]~~]]~~

b: #wonders#$^{\frown}$#if#

(59) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ~~ [^{IP} ~~[^{DP} *the* [^{NP} *man*]]~~ [$^{I'}$ I [^{VP} *wonders* [^{CP} *if the*]]]]~~]]~~

b: #wonders#$^{\frown}$#if#$^{\frown}$#the#

(60) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ~~ [^{IP} ~~[^{DP} *the* [^{NP} *man*]]~~ [$^{I'}$ I [^{VP} *wonders* [^{CP} *if*
[^{DP} *the* [^{NP} *woman*]]]]]]~~]]~~

b: #wonders#$^{\frown}$#if#$^{\frown}$#the#$^{\frown}$#woman#

At this point, it is time for **<u>was</u>** to access the derivation, turning the temporary structural complement of **<u>if</u>** into the definitive specifier of **<u>was</u>**. But before that happens, it is necessary to spell-out the current string (i.e. # **#wonders#**$^{\cap}$**#if#**$^{\cap}$**#the#**$^{\cap}$**#woman#**), to avoid a violation of the LCA, since **#the#** and **#woman#** would precede **#was#** despite neither *the* nor *woman* participating in c-command relations with *was*.

(61) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{IP} [^{DP} *the* [^{NP} *man*]] [$^{I'}$ I [^{VP} *wonders* [^{CP} *if*~~
~~[^{DP} *the* [^{NP} *woman*]]]]]]]~~

b: ∅

Once that is done, the subordinate Infl can finally access the derivation.

(62) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{IP} [^{DP} *the* [^{NP} *man*]] [$^{I'}$ *is* [^{VP} *wondering* [^{CP} *if*~~
[^{IP} ~~[^{DP} *the* [^{NP} *woman*]]~~ *was*]~~]]]]]]~~

b: #was#

The derivation goes on, and the subordinate verb is introduced as in (63). After that, the syntactic subject of the passive clause moves to its theta position, as in (65). This operation has no impact on PF, and no consequences to the LCA, since the substring corresponding to that phrase (i.e. **#the#**$^{\cap}$**#woman#**) is no longer in the derivational workspace.

(63) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{IP} [^{DP} *the* [^{NP} *man*]] [$^{I'}$ I [^{VP} *wonders* [^{CP} *if*~~

[^{IP} ~~[^{DP} *the* [^{NP} *woman*]]~~ [$^{I'}$ *was shot*]]]]]]]]

b: #was#$^{\cap}$#shot#

(64) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{IP} [^{DP} *the* [^{NP} *man*]] [$^{I'}$ I [^{VP} *wonders* [^{CP} *if*~~

[^{IP} t$_1$ [$^{I'}$ *was* [^{VP} *shot* ~~[^{DP} *the* [^{NP} *woman*]]~~$_1$]]]]]]]]]

b: #was#$^{\cap}$#shot#

Eventually the last string (i.e. **#was#$^{\cap}$#shot#**) is sent to the phonological component, as in (65); and the computational system is done with the derivation.

(65) a: ~~[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{IP} [^{DP} *the* [^{NP} *man*]] [$^{I'}$ I [^{VP} *wonders* [^{CP} *if*~~

~~[^{IP} t$_1$ [$^{I'}$ *was* [^{VP} *shot* [^{DP} *the* [^{NP} *woman*]]]]]]]]]]]~~

b: ∅

The global PF representation of this sentence, then, would be the concatenation of all spell-out strings in (66), yielding the longer string in (67).

(66) a: #the#$^{\cap}$#man#

b: #wonders#$^{\cap}$#if#$^{\cap}$#the#$^{\cap}$#woman#

c: #was#$^{\cap}$#shot#

(67) #the#⁀#man#⁀#wonders#⁀#if#⁀#the#⁀#woman#⁀#was#⁀#shot#

It is not obvious, though, as to why the strings in (66) should get concatenated exactly as in (67). There are other logical possibilities, as shown in (68), but for some reason, only one of them is a legitimate linearization.

(68) a: #the#⁀#man#⁀#wonders#⁀#if#⁀#the#⁀#woman#⁀#was#⁀#shot#
b: * #the#⁀#man#⁀#was#⁀#shot#⁀#wonders#⁀#if#⁀#the#⁀#woman#
c: * #wonders#⁀#if#⁀#the#⁀#woman#⁀#was#⁀#shot#⁀#the#⁀#man#
d: * #wonders#⁀#if#⁀#the#⁀#woman#⁀#the#⁀#man#⁀#was#⁀#shot#
e: * #was#⁀#shot#⁀#the#⁀#man#⁀#wonders#⁀#if#⁀#the#⁀#woman#
f: * #was#⁀#shot#⁀#wonders#⁀#if#⁀#the#⁀#woman#⁀#the#⁀#man#

My proposal is that, for any derivation, only one out of all logically possible options of concatenation of strings is actually available. The system has no choice to make due to an automaton that determines that, given any two strings of terminals X & Y, X precedes Y if and only if X accesses the phonological component before Y does. Moreover, under the assumption that 'the parser is the grammar' (in the technical sense of Phillips 1996, 2003) — which I have been tacitly assuming here —, when we talk about the order of operations in a given derivation, we are talking about real time. Thus, the concatenation of strings at PF works in a 'first-come-first-serve basis'. Therefore, this way of linearizing strings with respect to each other *"is thus seen to be ultimately related to the asymmetry of time"*[105].

Moreover, given the dynamics of the system, even if all strings eventually end up concatenated to each other in a single (longer) string, it is natural to expect these partial PF representations to behave as a prosodic domains, which define intonational phrasing, stress patterns, cliticization, and related grammatical processes. Therefore, (69) would be a better notation than (67). This seems to be the null hypothesis, since each string accesses the phonological component in an independent cycle.

(69) [#the#⁀#man#]⁀[#wonders#⁀#if#⁀#the#⁀#woman#]⁀[#was#⁀#shot#]

That said, notice that the machinery presented so far is not enough to avoid overgeneration of all unattested orders. For instance, once we reverse the standard LCA, and combine this assumption with multiple spell-outs, it seems that we are missing the generalization that specifiers must precede their corresponding heads. Given the possibility of spelling-out strings to allow new elements to be merged in a position not c-commanded by some preceding terminals, it does not seem to matter whether the system merges the specifier before its sister, as sketched in (70), or the other way around, as sketched in (71). Either way, multiple spell-out allows satisfaction of the LCA. Therefore, it seems

[105] The passage in italics is taken from Kayne (1994: 38), and used here to mean something slightly different from what the author meant.

that we are wrongly predicting that both (70) and (71) are legitimate derivations.[106]

(70) a: $[^{\Sigma P}$ ∅ $[^{\Sigma'}$ Σ $[^{DP}$ the $[^{NP}$ girl $[^{PP}$ from $[^{DP}$ D Korea]]]]]]

#the#$^\frown$#girl#$^\frown$#from#$^\frown$#Korea#

b: $[^{\Sigma P}$ ∅ $[^{\Sigma'}$ Σ $[^{IP}$ $[^{DP}$ the $[^{NP}$ girl $[^{PP}$ from $[^{DP}$ D Korea]]]] $[^{I'}$ will $[^{VP}$ kiss $[^{DP}$ D Max]]]]]]

[#the#$^\frown$#girl#$^\frown$#from#$^\frown$#Korea#]$^\frown$[#will#$^\frown$#kiss#$^\frown$#Max#]

(71) a: $[^{\Sigma P}$ ∅ $[^{\Sigma'}$ Σ $[^{I'}$ will $[^{VP}$ kiss $[^{DP}$ D $[^{NP}$ Max]]]]]]
#will#$^\frown$#kiss#$^\frown$#Max#

b: * $[^{\Sigma P}$ ∅ $[^{\Sigma'}$ Σ $[^{IP}$ $[^{I'}$ will $[^{VP}$ kiss $[^{DP}$ D Max]]] $[^{DP}$ the $[^{NP}$ girl $[^{PP}$ from $[^{DP}$ D Korea]]]]]]]
[#will#$^\frown$#kiss#$^\frown$#Max#]$^\frown$[#the#$^\frown$#girl#$^\frown$#from#$^\frown$#Korea#]

It seems, then, that we need to further constrain of *Merge*, so that we can capture the effects of Phillips' (1996, 2003) *Merge Right* and *Branch Right*, but this has to be done without the redundancy discussed above.

[106] In both (70) and (71), I am abstracting away from the VP internal subject position for expository purposes.

This goes back to the issue of the nature of tucking-in, discussed in **§IV.2**. As argued in Chomsky (2000: 136 (cf. quote in **§IV.2**)), tucking-in is the optimal way of satisfying the condition of *Local Merge*, which I define as in (72).

(72) *Sisterhood Condition on Syntactic Relations*

The establishment of any feature-checking or thematic relation between a head H and a phrase α requires that H and α be sisters at the relevant derivational step.

Besides that, we need a device that restricts the set of possible targets for *Merge*, as in (73).

(73) *'Active Node' Condition on Merge* (preliminary definition, to be refined)

A syntactic node **α** is *active* at a given derivational step **t** iff (i) or (ii):

i: **α** was tucked into the phrase marker at step **t**–1;

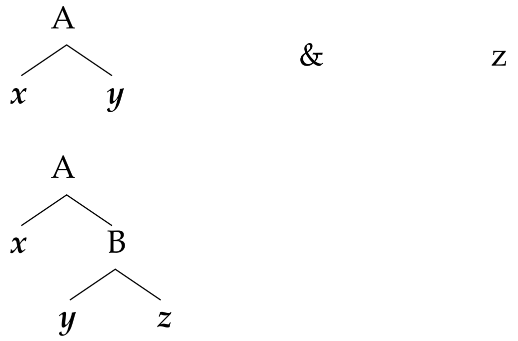
ii: The set of nodes dominated by **α** at step **t** –1 is a proper subset of the set of nodes dominated by **α** at step **t**.

The intuition behind the formal definition in (73) is that, in order for any constituent to be 'active' as a potential target of merge, and 'attract new sisters', it must be somehow 'new'. Any lexical token that was just integrated to the phrase marker step before is, by definition, the newest thing in the derivational workspace. In addition to that, each and every syntactic node that dominates an active node would also count as 'new'. This is so because of the 'anti-extension' requirement built into the very definition of merge, which causes the whole to change once its parts change.

For instance, in (74), although the node **A** in the input and the node **A** in the output are, for all intents and purposes, two incarnations of the same formal object (by virtue of them having the same label), they do not have the exact same internal structure. In the input, the daughters of **A** are *x* and *y*. In the output, the daughters of **A** are *x* and the new constituent **B**, but *y* is no longer a daughter of **A**.

(74) input:

output:

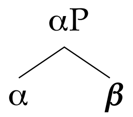


As discussed in **§IV.3.1** above, dominance relations are established in a monotonic fashion in a 'generalized tucking-in' system. Thus, when z gets tucked-in, there is a change in the set of dominance relations with respect to A (i.e. A dominated only *x* and *y* in the input, and dominates *z*, *y*, *z* and **B** in the output). It is this increment in the set of dominance relations that makes a node 'new'. Therefore, ***x*** and ***y*** remain 'old' after ***z*** is tucked in, which means that, by (73), neither ***x*** nor ***y*** can be the target of a new application of Merge (i.e. they cannot get a new sister).

Now, let us see how the conditions (72) and (73), together, conspire to yield the desired representations.

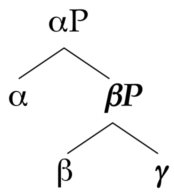
Consider the phrase marker in (75a) as the starting point (with αP being equivalent to the starting axiom introduced above, and β being the first element merged).

(75) a:



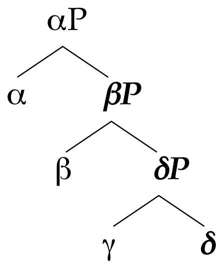
Suppose that a new element γ merges inside αP, as a sister to β, such that β projects, as in (75b).

(75) b:



At this point, γ is a (temporary) complement to β. Whether or not there will be some feature checking between γ and β, it depends on their selectional properties, theta-grids, and feature specifications. Let us consider the case where no such relation holds. Now, suppose that there is a new element **δ** to be integrated into the phrase marker. By (73), must me merged either as a sister to ***βP*** or as a sister to ***γ***, since these two constituents are the only active nodes at step (75b). Consider, then, the case where the new element **δ** is tucked in as a sister to **γ**, such that **δ** projects, as in (75c). As a consequence, **γ** is no longer a sister of **β** and no longer a daughter of **βP** (although **βP** is still its 'ancestor', dominating it).

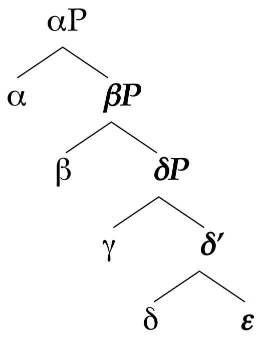
(75) c:



At this point, **γ** is the complement of **δ**. Assuming that **γ** and **δ** have some property to check against each other, this is the moment where such checking/evaluation takes place, since they are now sisters.

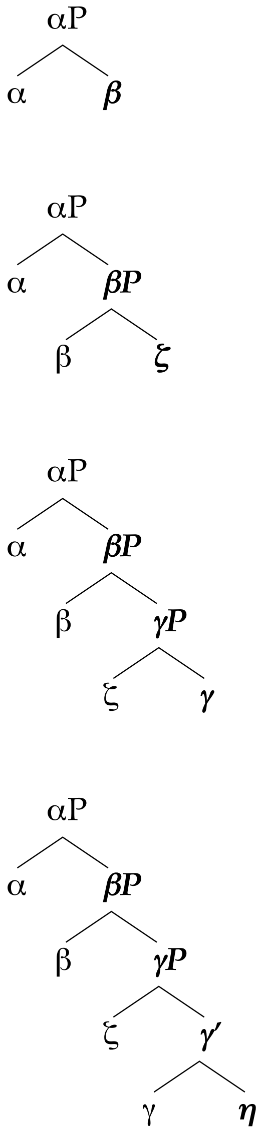
Suppose, further, that another new incoming element **ε** access the derivation. By (73), the candidates for being its sister are ***βP***, ***δP***, and ***δ***, since these are the active nodes in (75c). Assuming that **ε** and ***δ*** are 'compatible', **ε** merges inside ***δP***, as the new complement to **δ**, as in (75d).

(75) d:

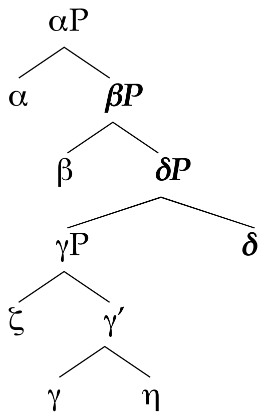


The scenario in (75) is the one of the simplest cases. In principle, specifiers can be arbitrarily complex, as exemplified in (76), which is essentially the same derivation as in (75), except that the specifier of ***δP*** is not an atomic phrase **γ**, but rather a complex phrase **γP** (= $[^{\gamma P}\ \zeta\ [^{\gamma'}\ \gamma\ \eta]]$). Therefore, the analog of step (75b) — i.e. the introduction of the specifier — is broken down into (76b), (76b′) and (76b′′), as shown below.

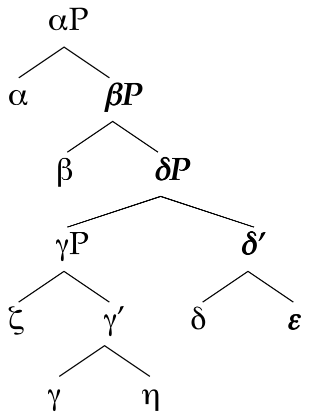
(76) a: b: b′: b′′:



c:

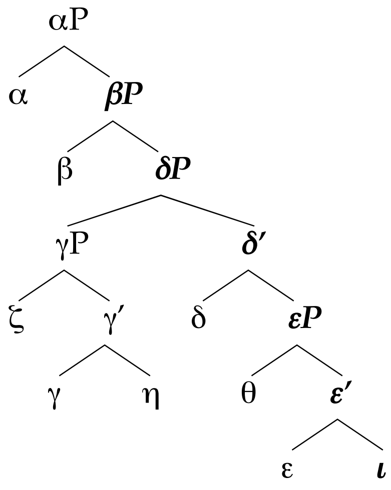


d:



Needless to say, nothing prevents **ε** from being an arbitrarily complex phrase itself, in which case the final representation would be something like (77).

(77)

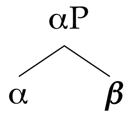


In both (75) and (76), specifiers are systematically introduced before the head of the phrase which they are specifiers of; and complements are systematically introduced right after the head of the phrase which they are complements of. This is not accidental. It follows from (72) and (73).

Let us now consider some alternative derivations for the same target representation(s), to see how they can be ruled out by (72) and (73).

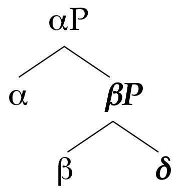
Take (78a) as the starting point.

(78) a:



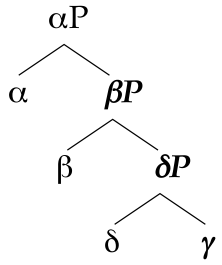
First, the head of **δP** is introduced before its specifier (i.e. **γ**), as in (78b).

(78) b:



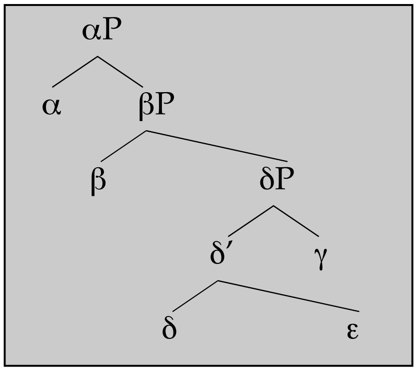
Then, the specifier (i.e. **γ**) is merged as a temporary complement of **δ**, as in (78c).

(78) c:



Up to this point, nothing wrong happened. The problem arises when the actual complement of **δ** (i.e. **ε**) is supposed to be tucked in, as a sister to **δ** in the next step. The structure in (78d) would be the intended representation, but that is impossible, since **δ** was not active in the input structure (i.e. (78c)).

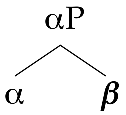
(78) d: *



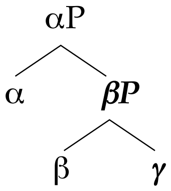
* head-specifier-complement

Another alternative derivation to be considered is the one in (79), where both the specifier and the complement are introduced before the head.

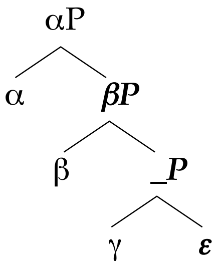
(79) a:



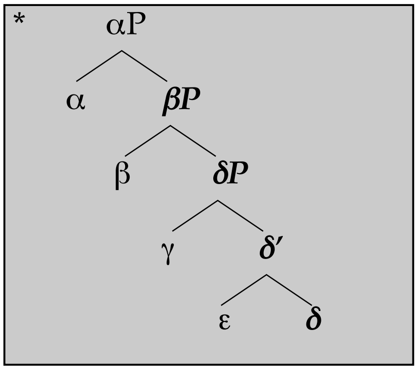
b:



c:



d:

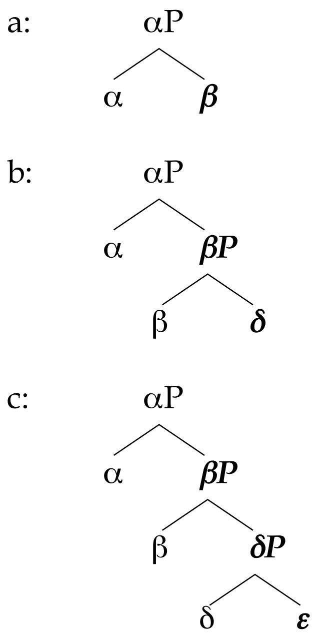


* specifier-complement-head

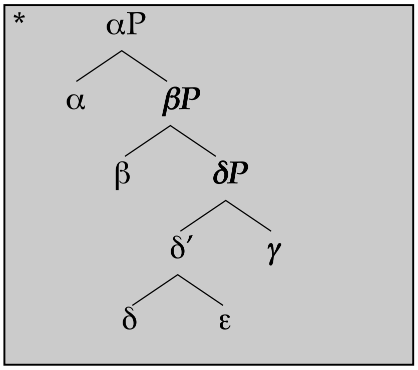
Putting aside the issue of whether constituent labels can be temporarily underspecified, as in (79c), this derivation is problematic because there is no step where the head **δ** and its specifier (i.e. **γ**) are sisters. Thus, by (72), the relevant relation cannot be established.

Yet another derivation to be ruled out is the one in (80), where the specifier (i.e. **γ**) is merged late, after both the head (i.e. **δ**) and the complement (i.e. **ε**) have been introduced.

(80)



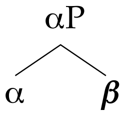
d:



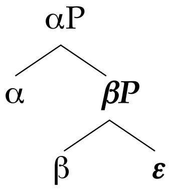
* head-complement-specifier

In (81), we see a variation of (80), where the specifier (i.e. **γ**) is also the last element to be introduced. The difference is that, in (81), the complement (i.e. **ε**) is introduced before the head (i.e. **δ**).

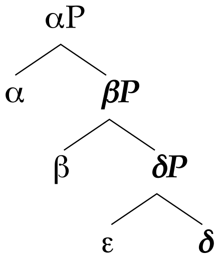
(81) a:



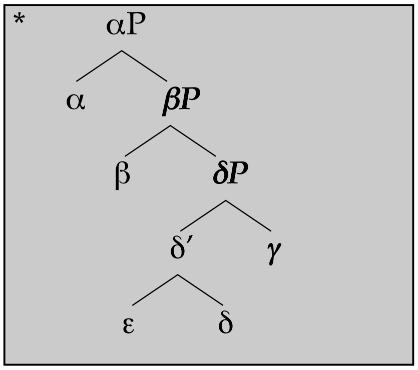
b:



c:



d:



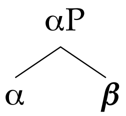
* complement-head-specifier

Notice that both derivations in (80) and (81) violate the sisterhood condition (72), as there is no step where the head **δ** and its specifier (i.e. **γ**) are sisters, making it impossible for the relevant relation to be established.

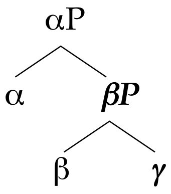
Under the assumption that adjuncts are fundamentally different from arguments, as the later participate in feature-checking and thematic relations but the former do not, it follows that, in principle, the system allows late insertion of adjuncts at the right edge of the phrase.

A sample derivation would be (82), where **σ** is an adjunct to **δP**.[107]

(82) a:

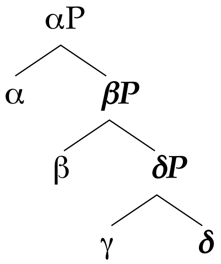


b:

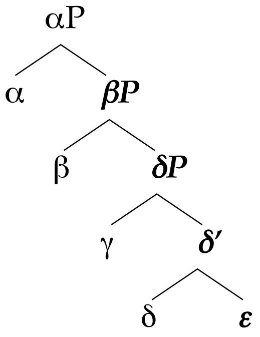


[107] Notice that, by this logic, there is no need to encode any difference between adjuncts and arguments in terms of bar-levels or any category/segment distinction. Also, I will leave it as an exercise to the reader the demonstration that such system predicts that adjuncts to the left can only be merged above the specifier, whereas adjuncts to the right can be the sister of any projection of the head.

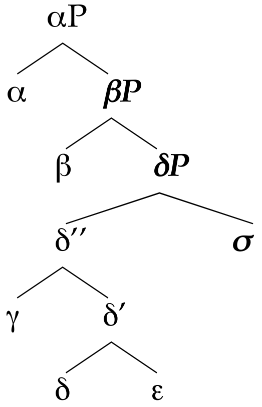
c:



d:



e:



At this point, one may question the role of the LCA in this system. After all, the internal mechanics of Merge itself — i.e. (72) and (73) — appears to be enough to derive the (adjunct)-specifier-head-complement-(adjunct) order.

Although, conceptually, the LCA is not necessary to derive the desired order, it does not introduce any redundancy into the system. This is so because, as defined in (45), the LCA is not a device that linearizes previously unlinearized structures. Rather, it is better understood as an internal device within a 'buffer' between syntax and phonology, which breaks the string of terminals into substrings that are delivered to the phonological component in 'cascades'. The c-command-to-precedence correspondence is the metric that determines the length of each 'cascade'.

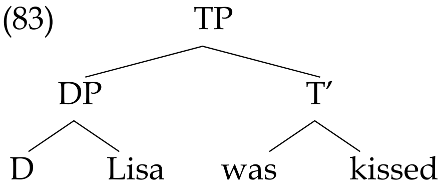
The reason for assuming that the string of terminals reaches the phonological component 'in cascades' is empirical. After the body of work known as *Prosodic Phonology* (cf. Selkirk 1984; Nespor & Vogel 1986; Inkelas & Zec 1990; and subsequent work), it is now a truism that the PF representation of any sentence is much more than a mere string of terminals. To a large extent, syntactic structure shapes prosodic structure, which, at the very least, contains boundaries that separate substrings of a certain kind (and, probably, more than that: like metrical grids, layers of constituents, part-whole relations, etc). The segmental and supra-segmental processes appear to be sensitive to such boundaries. If so, it must be the case that the grammar incorporates some mapping function from syntax to PF, which piggybacks on some structural property of phrase markers in order to determine where the major PF boundaries go. Without such device, and the major PF boundaries would either be absent or be placed according to extra-syntactic criteria (or even at random). That way, the observed (partial) connection between syntax and prosody would be entirely lost.

Relying upon recent work in Minimalism, where (asymmetric) c-command is the main syntactic relation — being pervasive across the whole grammar —, I have proposed in previous research (cf. Guimarães 1997, 1998, 1999a, 1999b), that (asymmetric) c-command is crucial to explain various PF phenomena (cliticization, stress, sandhi, etc.). For the purposes of this dissertation, I have chosen to discuss constraints on intonational phrasing to illustrate the point of why the LCA is necessary, and why it is crucial that it is implemented is a 'generalized tucking-in' system. This is presented in the Appendix at the end of this chapter.

IV.3.3. Movement

Now let us take a look at movement operations more closely. Consider the structure in (83), which would be an intermediate stage in the derivation of a passive sentence.[108]

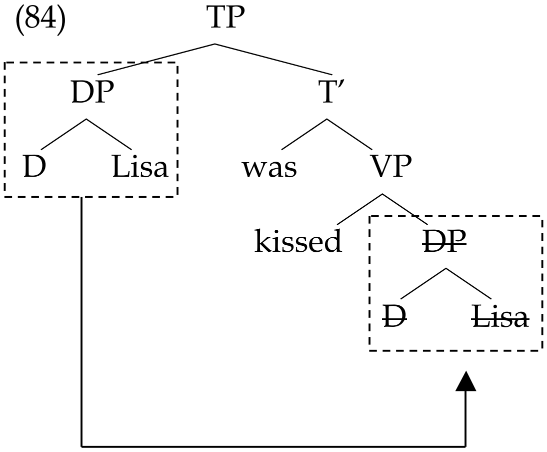
(83)



[108] CP and ΣP have are omitted from the notation for expository reasons, but are assumed to be present in the structure.

The basic intuition is that the dependency established between the subject DP generated in its case position and its theta position obtains through a movement operation, formalized as lowering, as in (84).

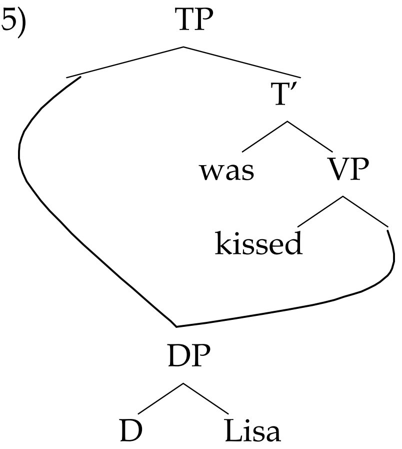
(84)



Under the technical implementation proposed by Phillips (1996, 2003), the effect of upward movement is achieved by making a silent copy (i.e. *copy* plus *PF-deletion*) of a given constituent and merging it at a position in the phrase marker lower than the position of the original copy, as shown in (84). Notice that this is not the traditional concept of lowering, since the moved element gets pronounced in its original/higher position.

In what follows, I assume that movement is nothing but 'remerge'. That is, a phrase may occupy more than one position at the same time, having multiple mothers and multiple sisters (cf. Bobaljik 1995; Drury 1998, 1999; Epstein, Groat, Kitahara & Kawashima 1998; Guimarães 1999, 2002, 2003b/c/d; Abels 2001, 2003; Gärtner 2002; Zhang 2003, *inter alia*). New motherhood and sisterhood relations are established through merge without eliminating the previous one(s), as shown in (85).

(85)



This remerge mechanics has at least two advantages over Phillips' (1996, 2003) approach to movement.

As shown in **§IV.3.1**, new structure can be tucked-in inside a phrase α after α has lowered from its highest position into its 'D-structure position' (so to speak). In some cases, the new structure that gets incorporated into α after the chain is created is actually part of the argument structure of a predicate inside α, which remained unsaturated until was lowered.

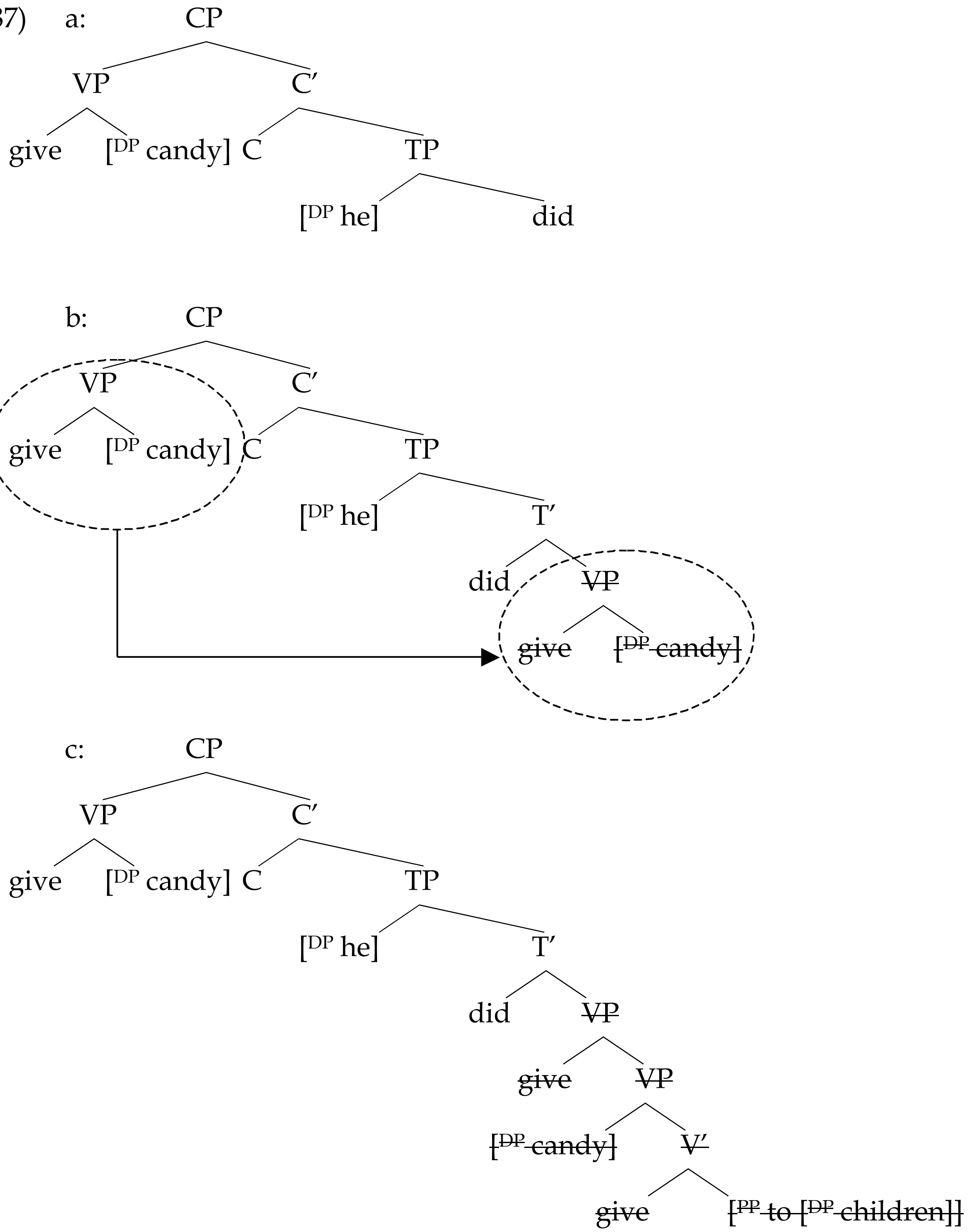
One example of such kind of derivation is the VP-topicalization construction, which was presented by Phillips (1996, 2003) as evidence for dynamic constituency. From that perspective, (86) would be derived as sketched in (87).[109]

[109] As I did in (29) above, I am abstracting away from the VP-internal subject position in (87) for expository reasons.

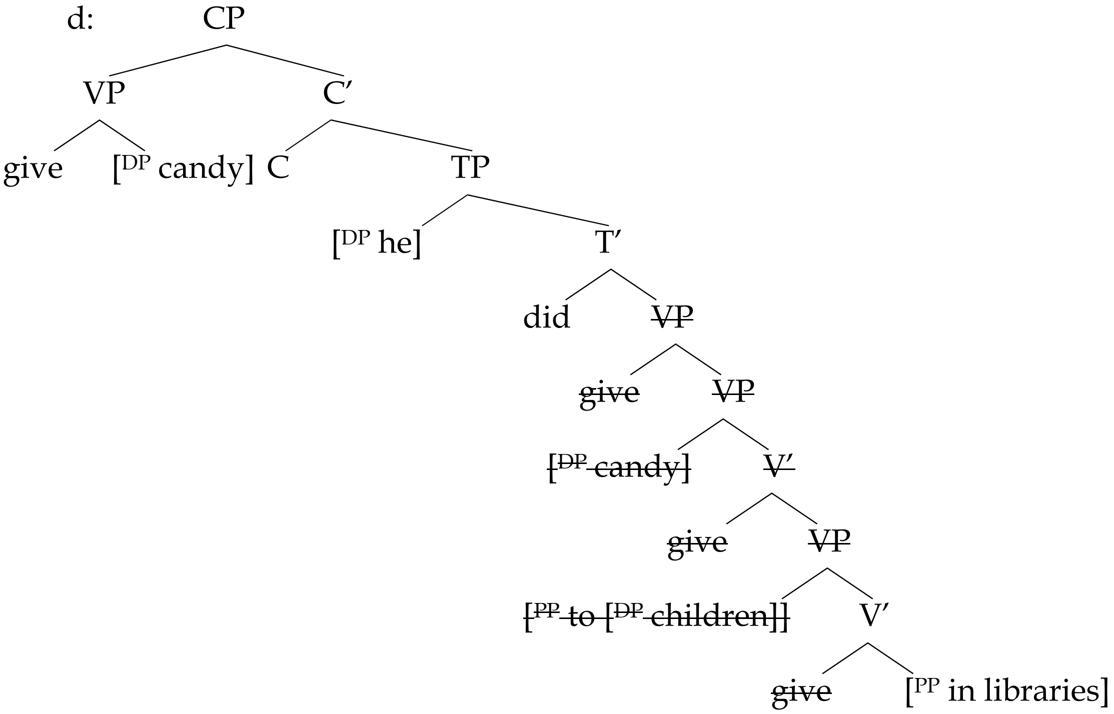
(86) John intended to give candy...

and [give candy]$_1$ he did t_1 to children in libraries on weekends.

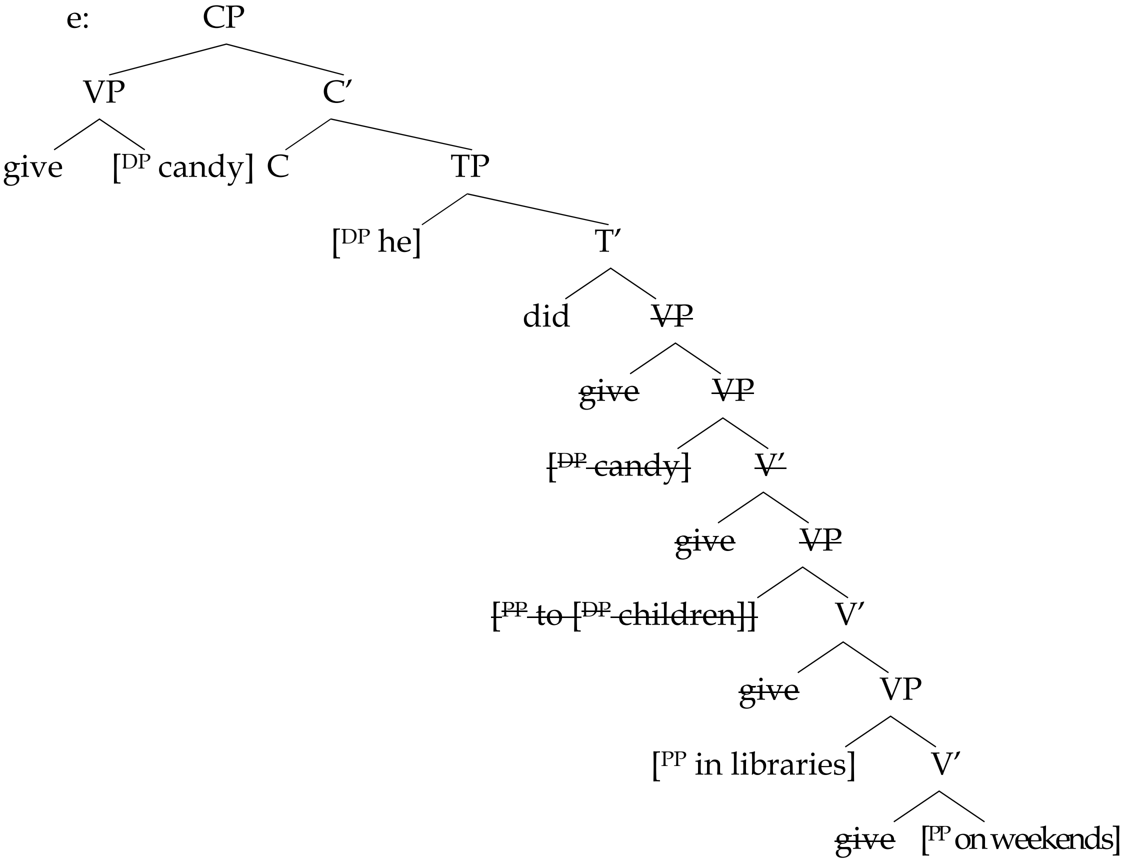
(87)



d:


CP
VP
C′
give
[DP candy]
C
TP
[DP he]
T′
did
~~VP~~
~~give~~
~~VP~~
~~[DP candy]~~
~~V′~~
~~give~~
~~VP~~
~~[PP to [DP children]]~~
V′
~~give~~
[PP in libraries]

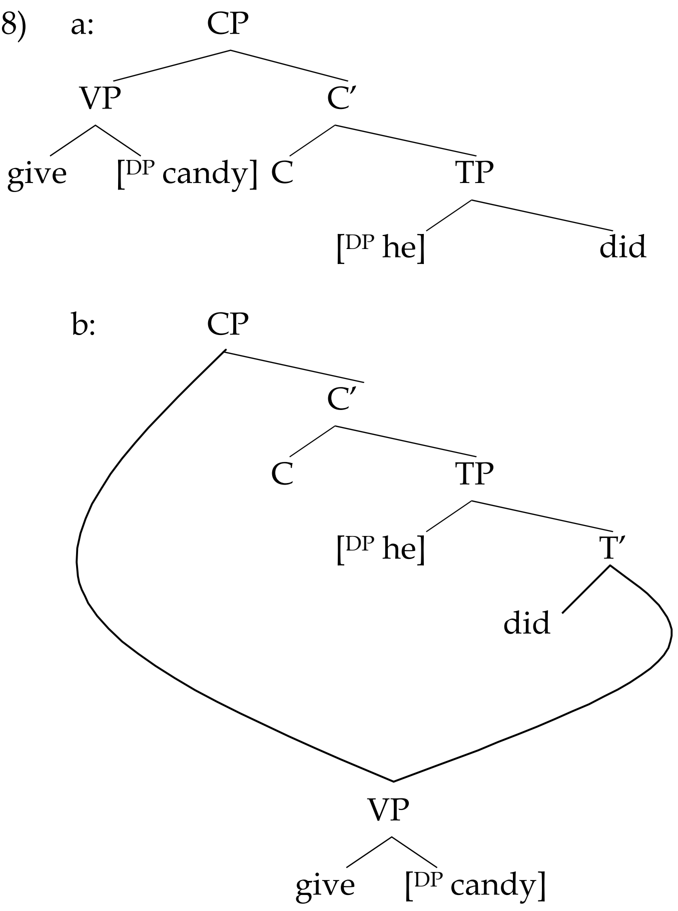
e:


CP
VP
C′
give
[DP candy]
C
TP
[DP he]
T′
did
~~VP~~
~~give~~
~~VP~~
~~[DP candy]~~
~~V′~~
~~give~~
~~VP~~
~~[PP to [DP children]]~~
V′
~~give~~
VP
[PP in libraries]
V′
~~give~~
[PP on weekends]

Notice that, before the topicalized VP is lowered, it contains an unsaturated predicate (i.e. give). As long as that configuration is temporary, there is no problem with that. However, by the end of the derivation, the topicalized VP should, at the very least, satisfy theta-criterion.[110]

This problem does not exist if movement is conceived as remerge. After lowering takes place, the internal structure of the topicalized VP grows, such that its predicate eventually gets saturated, as sketched in (88).[111]

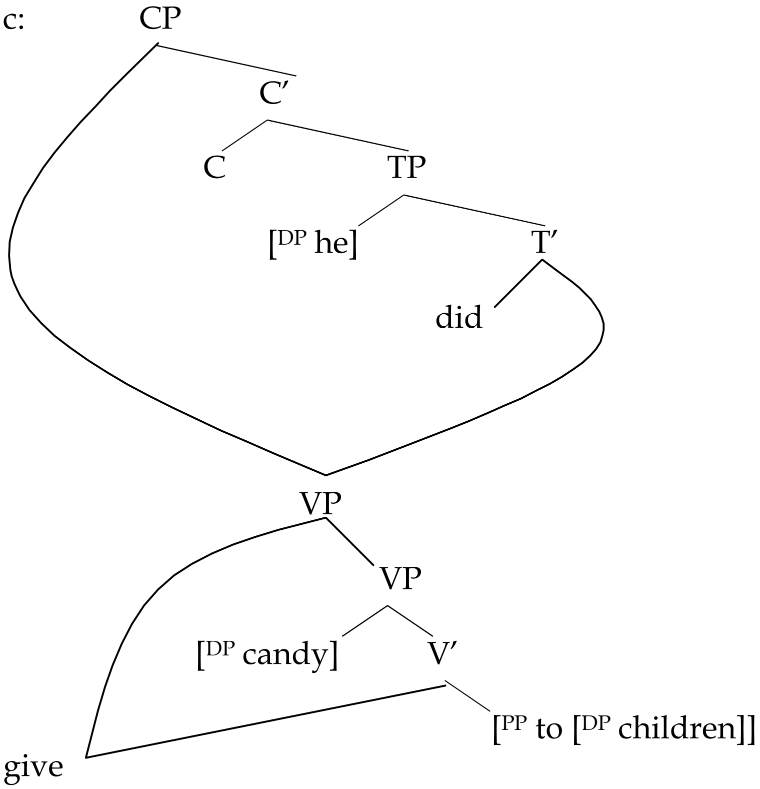
(88)



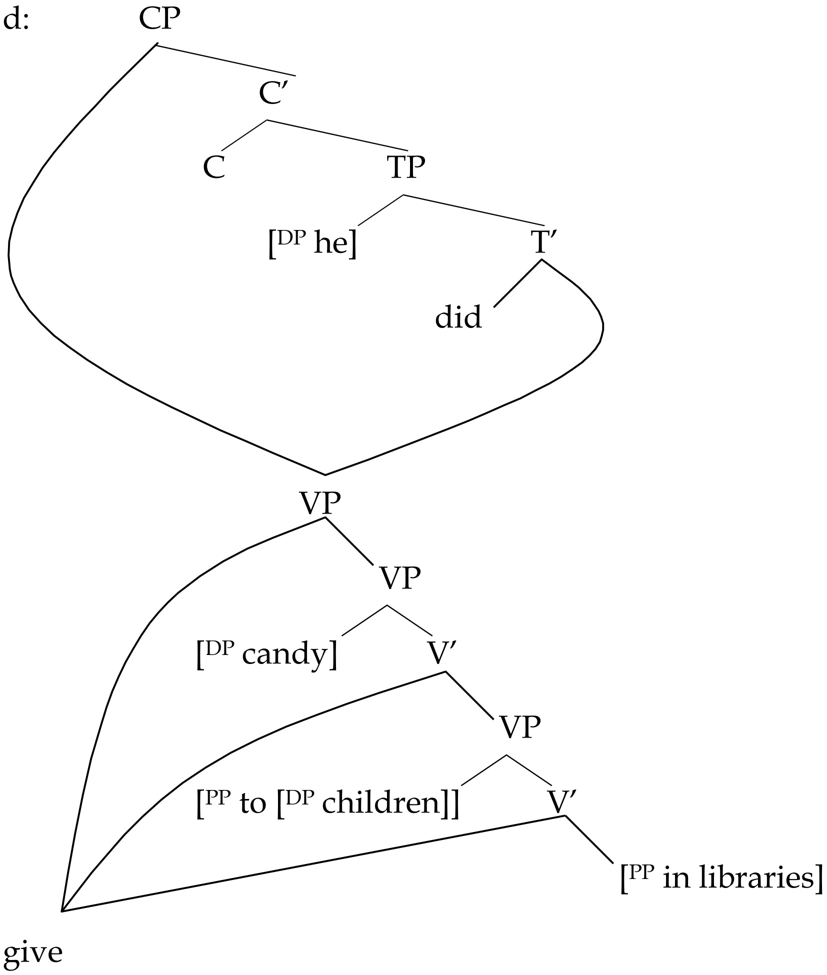
[110] In (87), I have abstracted away from the VP-internal subject position for expository reasons. Strictly speaking, the saturation of the predicate **<u>give</u>** in (87) is not achieved only by the introduction of to children in step (87c). At some point in the derivation, the subject **<u>[^{DP} he]</u>** must lower from spec/TP to spec/VP.

[111] Once again, I am abstracting away from the VP-internal subject position in (88). Also, I am skipping other details of the construction of the VP shell, such as the multiple instances of remerge of **<u>give</u>**. This mechanics will be discussed in detail in chapter **V**.

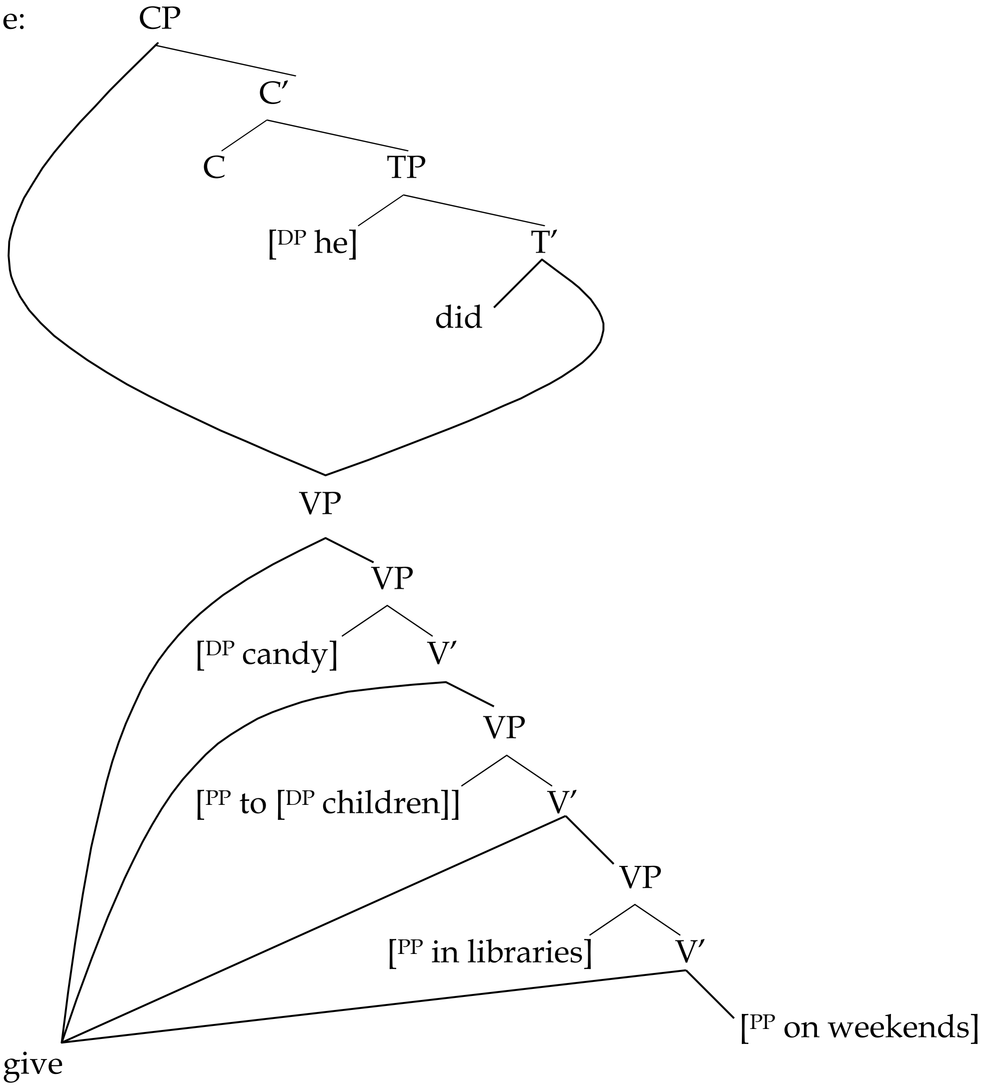
c:



d:


CP
C′
C
TP
[DP he]
T′
did
VP
VP
[DP candy]
V′
VP
[PP to [DP children]]
V′
[PP in libraries]
give

e:


CP
C′
C
TP
[DP he]
T′
did
VP
VP
[DP candy]
V′
VP
[PP to [DP children]]
V′
VP
[PP in libraries]
V′
[PP on weekends]
give

The other advantage of the remerge-based approach over the copy-based approach concerns PF. In Phillips’ (1996, 2003) system, not only is it necessary to assign a theoretical status to the notion of *copy* — which is not trivial in itself (cf. Bobaljik 1995) —, but also it must be stipulated that the lower copy is ‘silent’.

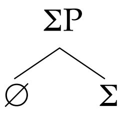
In the remerge-based approach, on the other hand, the fact that the moved element is always pronounced as if it were only in the higher position follows from the ‘upfront linearization’ mechanics coupled with the ‘multiple spell-out’ derivational dynamics. When remerge takes place, the phonological features of the element are no longer in the syntactic component. What is being ‘lowered is an organization of λ-particles whose corresponding π-particles have already left the derivation for good. This is exemplified in (90), which is the derivation corresponding to (89a).

(89) a: Lisa was kissed.

b: * Lisa was kissed Lisa.

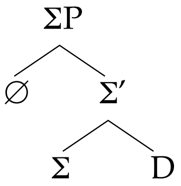
c: * was kissed Lisa.

(90) a:



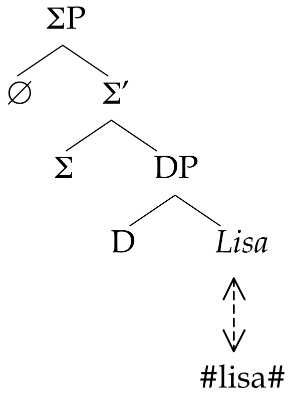
(starting axiom)

b:



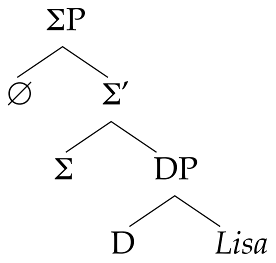
(merge D)

c:



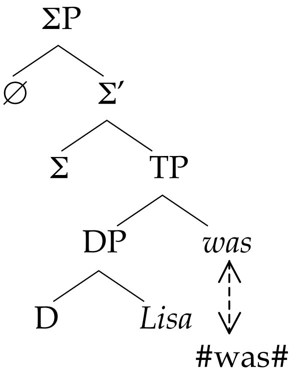
(merge *Lisa*)

d:



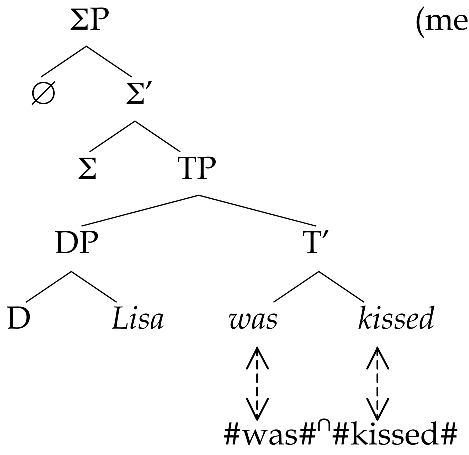
(spell-out)

e:



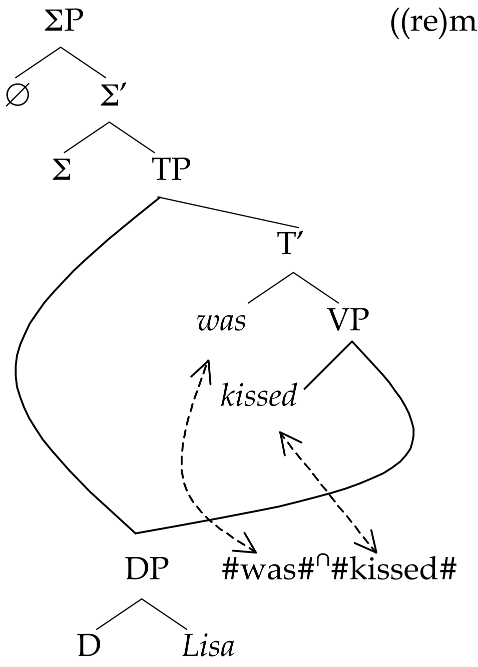
(merge *was*)

f:

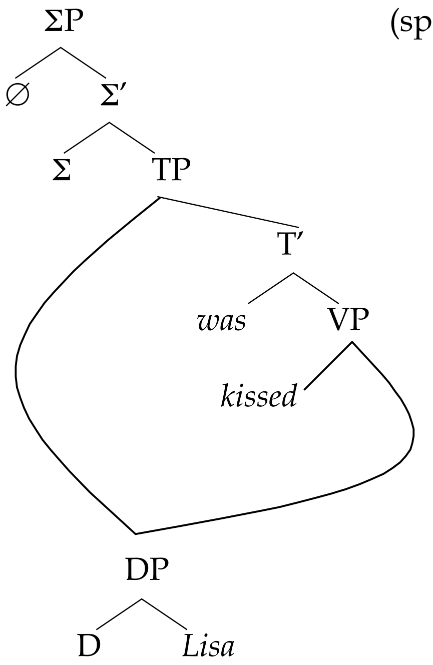


(merge *kissed*)

g: ((re)merge *[^{DP} D Lisa]*)



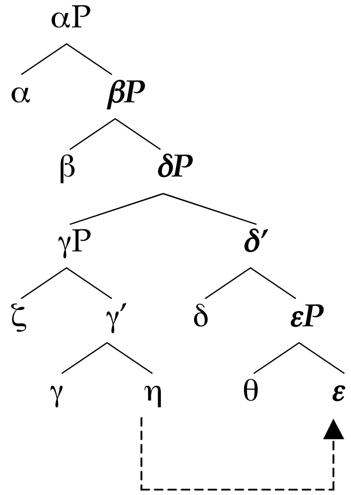
h: (spell-out)



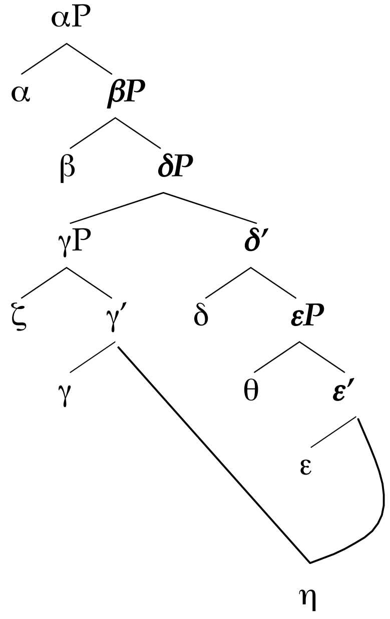
at PF: [#Lisa#]$^\cap$[#was#$^\cap$#kissed#]

It is important to make sure that this remerge mechanism is constrained enough to block the overgeneration of chains whose links do not stand in c-command relation with each other, as in (91), or else the system would go against a well known generalization about movement. Just like movement is always to a c-commanding position in bottom-up systems, top-to-bottom systems should exhibit movement only to a c-commanded position.

(91) input:



output: *



If remerge does not have any independent theoretical status, being just ordinary merge applied to an old constituent, then this c-command condition should be built into the definition of merge itself,[112] along with the 'active node' condition in (73).

This leads us to the final definition of Merge in (92).

(92) *Merge*: (final definition, modified from (34))

input: {A, {*x*, *y*}} & *z*, such that (i) & (ii) hold:

i: *z* c-commands ***y***

ii: ***y*** is active (cf. (73))

output: {A, {*x*, {B, {*y*, *z*}}}}

Notice that this modification does not affect the basic cases where incoming atoms access the derivation via 'first merge'. Consider *z* in (92) is a syntactic atom just taken from the numeration (hence, not connected to anything in the phrase marker yet). Once the standard definition of c-command in (93) is assumed, then *z* would automatically asymmetrically c-command **[**^{A} *x y***]**. This is so because the condition (93-iii) is vacuously satisfied, since there is nothing dominating *z* to begin with.

[112] I do not consider this a definitive solution. More research is needed to derive this c-command requirement on chains from some deeper property of derivations.

(93) *C-Command*:

α c-commands β if and only if (i), (ii) and (iii) hold:

i: α ≠ β;

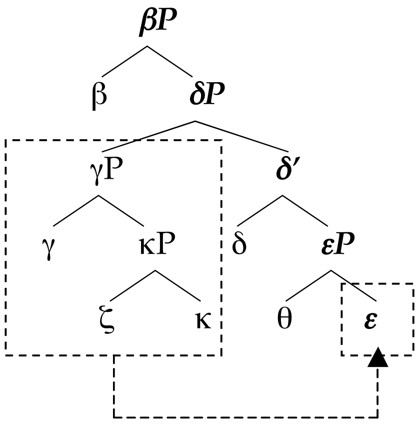
ii: α does not dominate β;

iii: every category that dominates α also dominates β.

Finally, let us further scrutinize the cases where a given phrase β (either atomic or complex) is merged inside an old phrase α after α has lowered from its highest position into its 'D-structure position' (so to speak). Consider the derivation in (94).

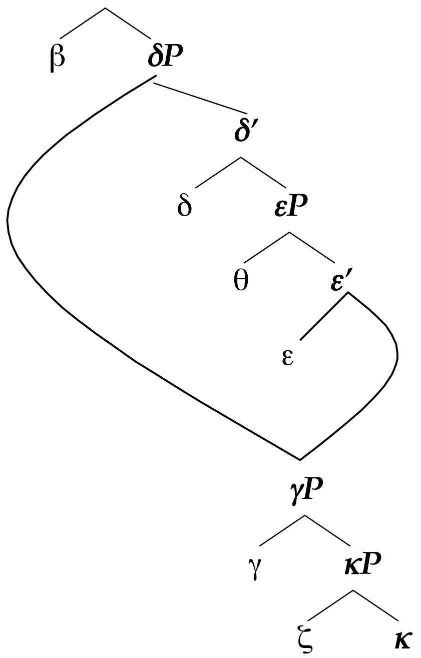
From (94a) to (94b), **γP** is lowered, remerging as a sister of **ε**. This is possible because, in the input, (i) **ε** is a 'new' constituent (hence a potential target for merge), and (ii) **γP** c-commands **ε**.

(94) a:



targeting the future co-sister

b: ***βP***

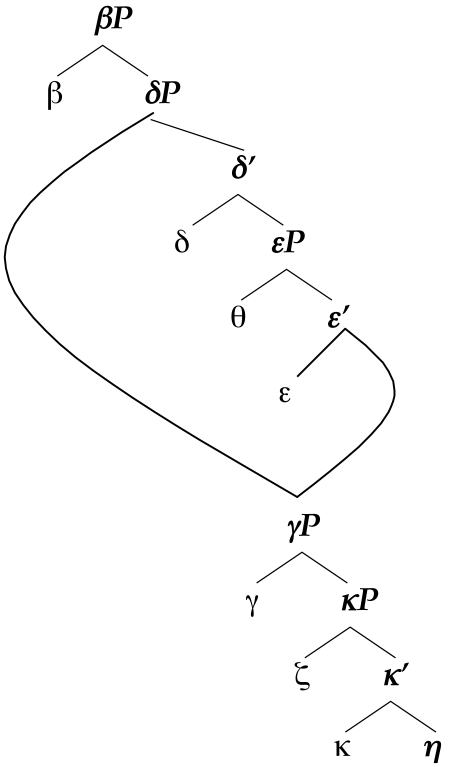


Notice that, immediately after **γP** lowers, it becomes an active node, since it is recognized by the system as the last element tucked into the phrase marker (cf. (73-i)). Therefore, it qualifies as a potential attachment point for future merge operations.

How about the proper subconstituents of **γP**? The desirable outcome (which would be compatible with both Phillips's (1996, 2003) account for conflicting constituency diagnostics and my analysis of syntactic amalgams in **§V**) is that some of the proper subconstituents of the lowered phrase (i.e. **γP**) should have the status of active: namely, the ones in the spine of **γP** (i.e. **κ** and **κP**). In what follows, I will assume that this is the case, although, at this point, I do not have any formalization to offer. The intuition that I will pursue is that, if any given maximal projection XP is lowered, not only XP becomes 'new' at that point, but also all of its subconstituents that were 'the same age' as XP. Those nodes would correspond to the newest nodes inside XP, i.e. the ones that were 'brand new' right before XP was 'pushed out of the spine' (i.e. when it became a specifier). When applied to the derivational step in (94b), this reasoning leads us to conclude that, at that point, **γP**, **κ** and **κP** are active, and therefore count as potential sisters for the element to be tucked into the phrase marker in the next step.[113]

Form that perspective, the derivation in question can proceed as in (94c), where the incoming element **η** is introduced deep inside **γP**, therefore changing its internal structure from $\mathbf{[^{\gamma P}\ \gamma\ [^{\kappa P}\ \zeta\ \kappa]]}$ to $\mathbf{[^{\gamma P}\ \gamma\ [^{\kappa P}\ \zeta\ [^{\kappa'}\ \kappa\ \eta]]]}$.

(94) c:



[113] By (73b), also ***βP***, ***δP***, ***δ′***, ***εP*** and ***ε′*** would count as active nodes at this point, which is not relevant for the point being made now.

As a result, we get a word-order pattern in which $\mathbf{[^{\gamma P}\ \zeta\ [^{\gamma'}\ \gamma\ \eta]]}$ is discontinuously pronounced, since there is no corresponding substring $\mathbf{\#\gamma\#^\frown\#\zeta\#^\frown\#\kappa\#^\frown\#\eta\#}$ at PF. Rather, there is a substring $\mathbf{\#\gamma\#^\frown\#\zeta\#^\frown\#\kappa\#}$ and a substring $\mathbf{\#\eta\#}$ which are not adjacent to each other.

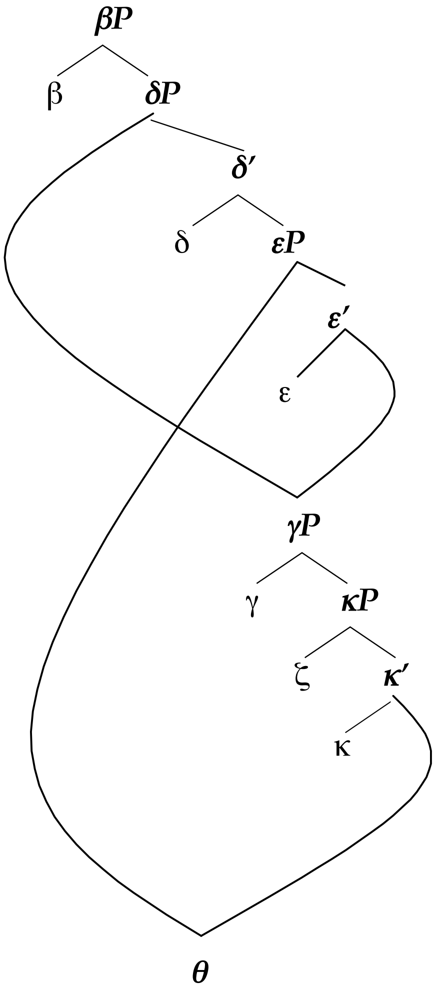
Notice that nothing in the system forces that the element being tucked into the lowered phrase **γP** be an atom just selected from the numeration. In principle, it can be any element already present in the phrase marker (as long as the c-command requirement is met).[114] For instance, instead of going from (94b) to (94c) by introducing **η** into the derivation, the system could have gone from (94b) to (94c′) by lowering **θ** and remerging it as a sister of **κ**.[115]

In nutshell, what happens in (94c′) is that an element already present in the phrase marker is lowered into a lowered phrase. This is exactly how remnant movement (cf. Müller 1998) works in a top-to-bottom system.

[114] As will be shown in §5, this is often the case with syntactic amalgams.

[115] Needless to say, nothing prevents θ in (94c′) from being a complex phrase, rather than an atomic one.

(94) c′:



IV.4. Remerge Without Movement: shared constituency and multiple roots

In the previous section, I argued that movement is best understood in terms of multi-motherhood relations, which obtains through remerge.

In this section, I propose that this same mechanics should be extended to other configurations beyond just chains. This will play a crucial role in the analysis of syntactic amalgamation in chapter **V**.

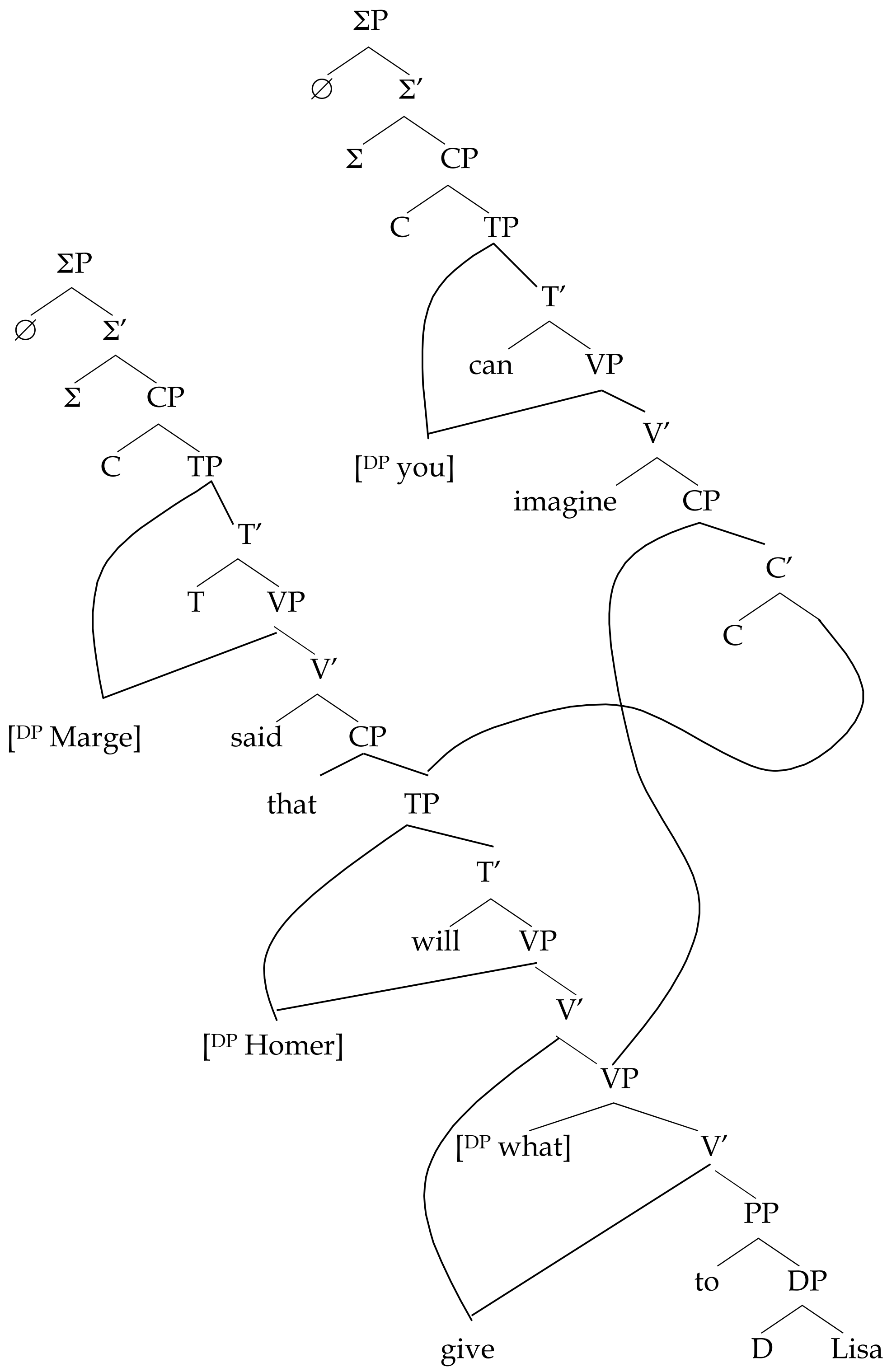
Following van Riemsdijk (2000) and de Vries (2003),[116] I assume that syntactic representations may exhibit multiply-rooted phrase markers with parallel trees that share some constituent(s) somewhere in between the roots and the terminals (*via* multi-motherhood), as shown in (96), which corresponds to the example in (95).[117]

(95) Marge said that Homer will give you can imagine what to Lisa.

[116] The idea of shared constituency and multi-motherhood goes back to McCawley (1982; 1987) and Goodal (1987); and it has recently been explored in some different ways by Muadz (1991), Moltmann (1992), Wilder (1999) and Citko (2000, 2002) among others.

[117] I exemplify multiple-rootness here with a syntactic amalgam for obvious reasons. Riemsdijk (2000) applies the idea mainly to transparent free relatives, whereas de Vries (2003) does so mainly for coordination.

(96)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
can
VP
[DP you]
V′
imagine
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
[DP Marge]
said
CP
that
TP
T′
will
VP
[DP Homer]
V′
VP
[DP what]
V′
PP
to
DP
give
D
Lisa

Notice that the ‘Siamese Trees’ in (96) can be factored out into two phrase markers, as in (97).[118] Basically, these two phrase markers are *quasi*-independent parallel structures that share one constituent (i.e. the embedded TP: **Homer will give t$_1$ to Lisa**).

(97) a: [^{CP} C [^{TP} Marge$_4$ [$^{T'}$ T [^{VP} t$_4$ [$^{V'}$ said [^{CP} that [^{TP} Homer$_2$ [$^{T'}$ will [^{VP} t$_2$ [$^{V'}$ give t$_1$ [^{PP} to Lisa]]]]]]]]]]]

b: [^{CP} C [^{TP} you$_3$ [$^{T'}$ can [^{VP} t$_3$ [$^{V'}$ imagine [^{CP} what$_1$ [$^{C'}$ C [^{TP} Homer$_2$ [$^{T'}$ will [^{VP} t$_2$ [$^{V'}$ give t$_1$ [^{PP} to Lisa]]]]]]]]]]]]

The basic idea is that structure sharing —which formally corresponds to multi-motherhood, obtained via remerging a given constituent (in this case, the embedded TP) — is what gives rise to paratactic relations.

Thus, as put by de Vries (2003: 205-207), aside from dominance (and its derivative: c-command), which relates syntactic nodes hypotactically, there is also ‘behindance’, which relates syntactic nodes paratactically.

> (...) [A] third dimension could be a useful addition to syntax in principle. In general we can say this: paratactic material interferes with the linear order of the matrix, but it backs out of the dominance relations. Therefore I will assume that two nodes in a syntactic structure can be related not only by dominance, but also by ‘behindance’. (...) [N]ext to dominance and precedence we have a third relation called *behindance*. We can then say that syntactic relations are defined in terms of dominance, whereas behindance encodes paratactic relations, and precedence is related directly to word order. Independent relations are mathematically orthogonal to each other. Since we have three degrees of freedom here, we may envisage the syntactical space as a cube. The x-axis encodes precedence, the y-axis dominance and the z-axis behindance.

[118] The terminology ‘Siamese Treee’ is taken from Riemsdijk (2000).

For instance, in (96), the VP headed by **<u>imagine</u>** is *behind* the VP headed by **<u>said</u>**. Notice that, interpretation-wise, neither the event of saying scopes over the event of imagining nor vice versa. However, there is one asymmetry at the level of informational structure: namely, the event of saying is interpreted as the 'main message', whereas the event of imagining is interpreted as a secondary thought. Interestingly, both scope over the very same event of giving.

I suggest that, in a system like the one proposed here, where derivational time plays a crucial role, the C-I interface piggybacks on the order of introduction of terminals (which ends up being reflected as PF-precedence, as discussed above) to encode 'salience' of the corresponding non-terminal nodes at the level of informational structure.

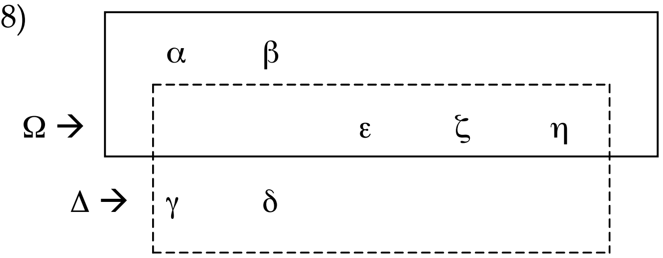
In a nutshell, given any two non-terminal nodes that are not under the same root, whichever of them gets to be built first will be seen by the C-I interface as figuring 'at the front' at the level of information structure, which is a notion established 'on the fly', as the derivation proceeds, which is in accordance with Phillips' (1996: chapter 5) idea that the grammar and the parser are the same structure building engine.

In (96), there are two matrix clauses (i.e. (97a) and (97b)). The fact that (97a) is 'at the front' makes it the '*master* clause' of the whole paratactic construction, whereas (97b), being 'on the back', becomes *subservient* to (97a). basically, whichever of the parallel matrix clauses gets to be built first automatically gets assigned the status of master matrix clause, whereas all the subsequent others become subservient matrix clauses.

That said, let me introduce the basic tools of the derivational mechanics that yields behindance relations in ‘Siamese Trees’.

As already said in **§IV.1**, inputs to syntactic derivations can be made up of multiple intersecting numerations, as in (98).

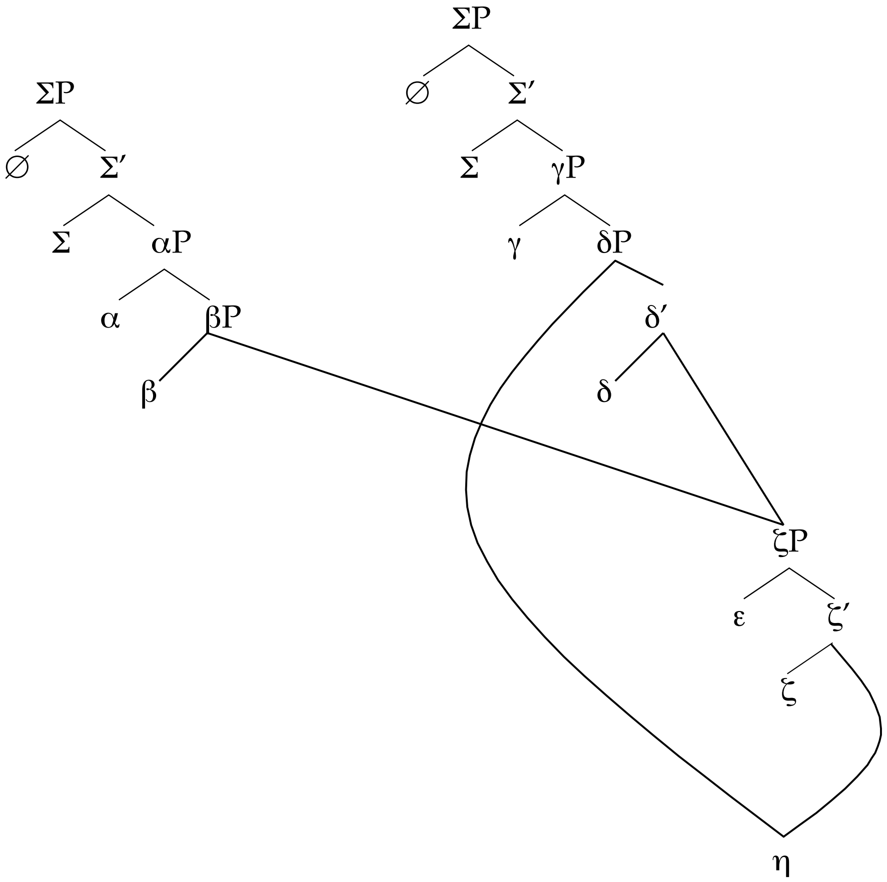
(98)



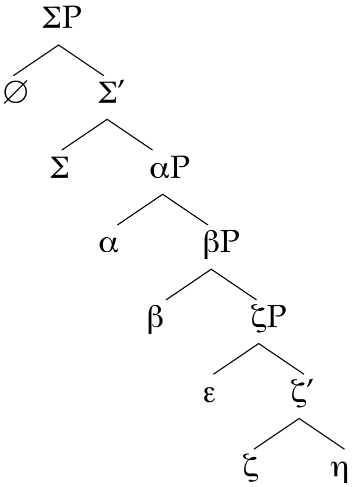
Once an input like (98) above is established, two (sub)computations will run, one for each numeration, and the intersection allows for these (sub)computations to interfere with each other to some extent.

Consider the target global structure to be (99), which breaks down into (100a) and (100b).

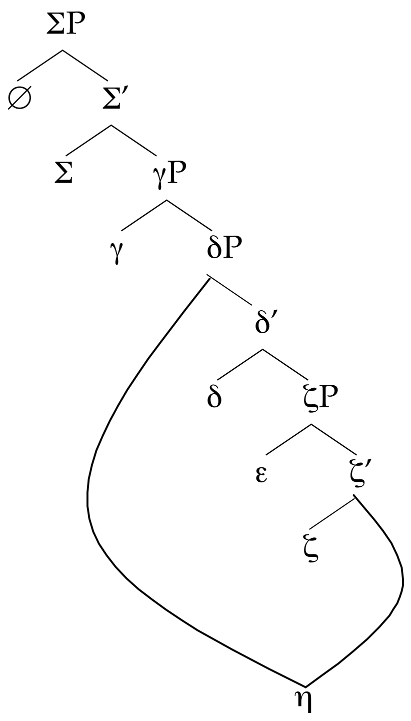
(99)



(100) a:



b:

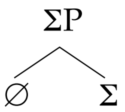


Since no asymmetry between the multiple numerations is encoded in the input, the choice of which numeration to start from is a random one. Whichever one gets picked first will correspond to the ‘master clause’.

Let is consider the case where Ω is randomly chosen as the starting point. By the assumptions made in **§IV.3** above, the derivation of (99) will then be as in (111).

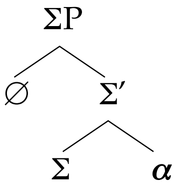
First, the system ‘zooms into’ Ω and proceeds tucking in the lexical tokens in the usual fashion, as in (111a) through (111e).

(111) a:



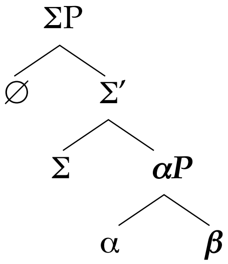
(starting axiom)

b:



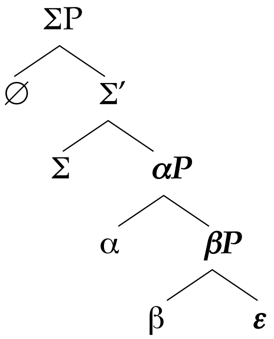
(merge α)

c:



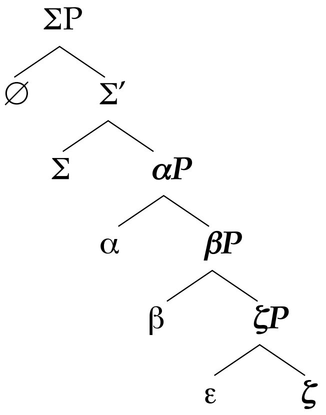
(merge β)

d:



(merge ε)

e:



(merge ζ)

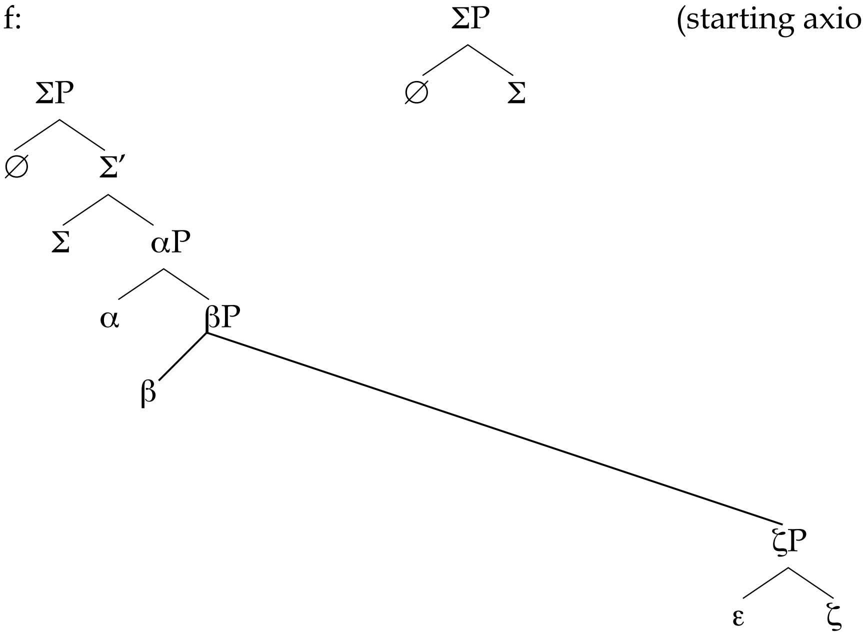
Notice that, at step (111e), the master clause is not complete yet. Given the target structure in (100a), an extra element is supposed to take place, where **η** mergesas the complement of **ζ**.

However, as it will become clear soon, step (111e) is as far as the system can go without crashing. This is so because, if **η** is introduced in the first derivational flow, it will not be able to remerge in the appropriate position in the subservient clause in the next derivational flow. The mere fact that **η** is left out at the end in the first derivational flow does not immediately make it impossible for the master clause to be eventually completed, since **η** is also present in numeration Δ, and the next subcomputation can, in principle, 'take care of it', as it will be made clear in chapter **V**, with concrete examples of syntactic amalgamation.

Thus, at this interruption point, the phrase marker corresponding to the incomplete *master clause* is spelled-out, and the string $\boldsymbol{\#\alpha\#^{\frown}\#\beta\#^{\frown}\#\varepsilon\#^{\frown}\#\zeta\#}$ is delivered to PF. The application of spell-out is mandatory at this point, otherwise the LCA would be massively violated in subsequent steps, as new terminals are about to be introduced 'behind' the ones already in the derivational workspace, therefore violating the asymmetric c-command requirement. After spell-out, all relevant π-particles of previous cascades are no longer in the derivational workspace, and consequently the LCA gets trivially satisfied.

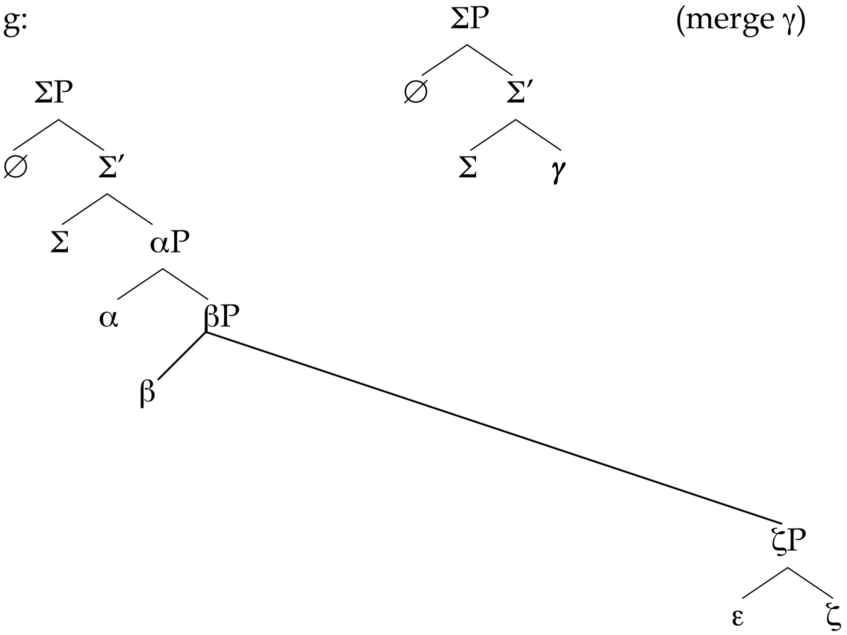
Then, the computational systems shifts its attention to numeration Δ, and the construction of the subservient clause proceeds with the lexical tokens being tucked-in in the usual fashion, as in (111f) through (111k).

(111) f:



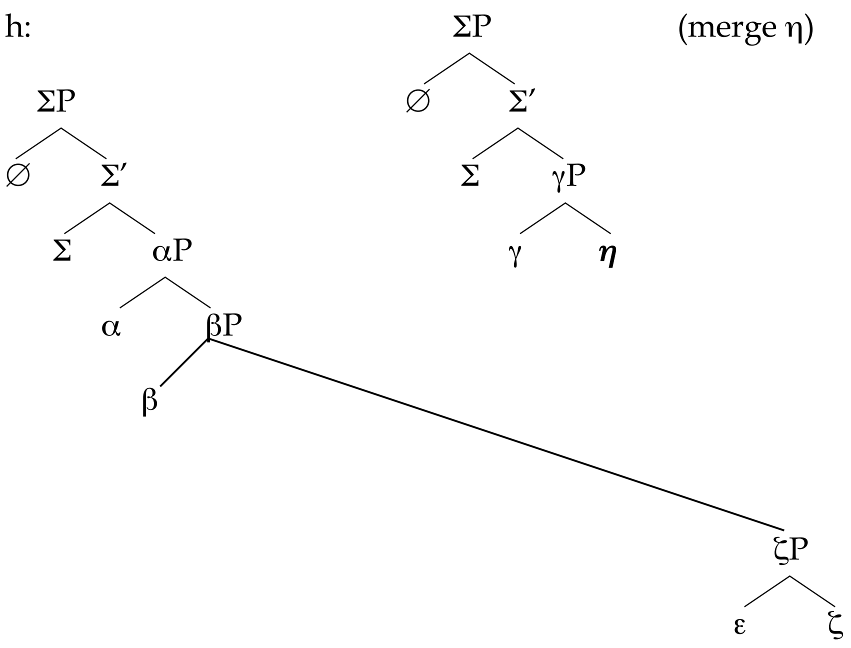
(starting axiom)

g:



(merge γ)

h:

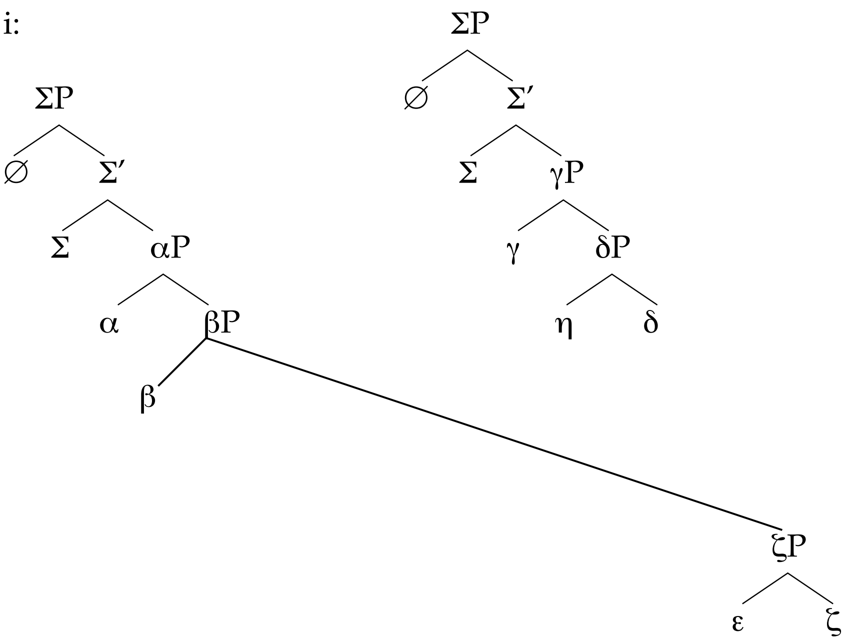


(merge η)

This derivational stage deserves further comment. If **η** had been previously tucked-in within **ζP** in the previous derivational flow, it would not be able to be remerged as a (temporary) sister of **γ** in step (111h). This is so because **ζP** is visible to both computations (given that its terminals belong in the intersection of numerations). Therefore, **η** would fail to c-command the target of merge (i.e. **γ**). This is why the introduction of **η** must be delayed in those cases. As it will be discussed in detail in chapter **V**, with concrete data, this 'late merge' mechanics is crucial in the derivation of syntactic amalgams.

The next steps are as follows.

(111) i:



At this point, there are two separate structures which do not share constituents (yet).

In the following step, the whole constituent **ζP** 'travels' from one subderivation to the other and is remerged as the complement of **δ**, yielding (99j).[119]

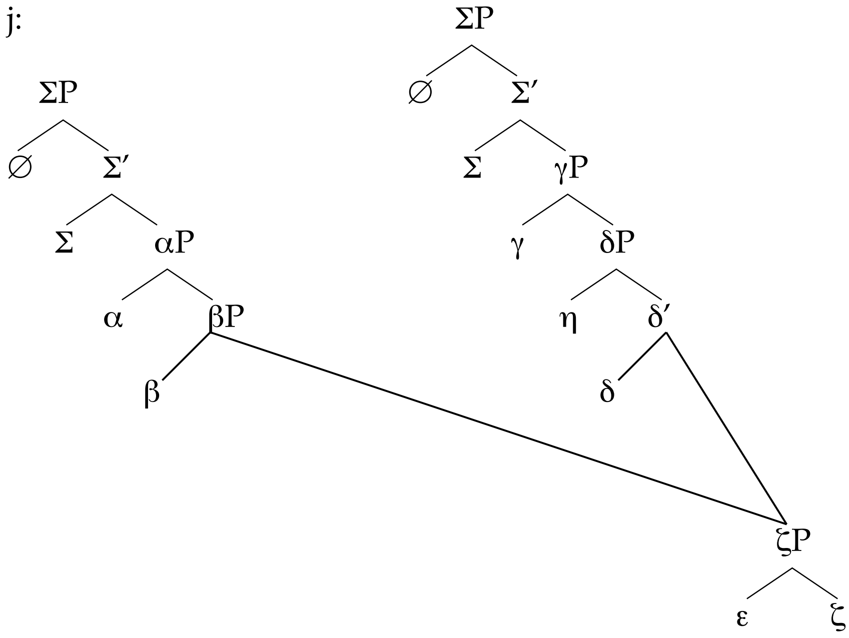
[119] *Mutatis mutandis*, this mechanism of phrases travelling from one subderivation to another is also found in Chomsky's (2000, 2001) original notion of factoring out computations into subderivations and subnumerations. The derivation of the sentence in (i) would start from the numeration in (ii). First, the items from the subnumeration {Mary, loves, him} are combined to form (iii). Then, in the next round, the output of that subderivation is taken by the other subderivation and embedded inside the larger structure in (iv).

i: Paul knows Mary loves him.
ii: {{Paul, knows}, {Mary, loves, him}} (functional heads omitted for expository reasons)
iii: [^{CP} [^{IP} [^{DP} Mary] [^{VP} loves [^{DP} him]]]]
iv: [^{CP} [^{IP} [^{DP} Paul] [^{VP} knows [^{CP} [^{IP} [^{DP} Mary] [^{VP} loves [^{DP} him]]]]]]]

There are three ways in which this formalism differs from what I am proposing. First of all, in (i-iv) above, it is the root node generated by one subderivation that travels to the other subderivation, while in my account of multiple amalgams it is a non-root constituent that travels across subderivations. Moreover, in Chomsky's (2000, 2001a) system, there is no intersection between two (or more) subnumerations, as opposed to what happens in my system. Finally, Chomsky's (2000, 2001a) formalism involves *merge* instead of *remerge*. As for the first issue, it

The basic intuition is that the shared tokens (with the exception of **η**, which is not yet inside **ζP** at the relevant step) are selected all at once. Since they are already part of a syntactic structure being built in the derivational workspace, the system takes that whole constituent and remerges it into the targeted position, as in (111j). This is the most economical strategy, because a single application of (re)merge generates the intended structure.

(111) j:



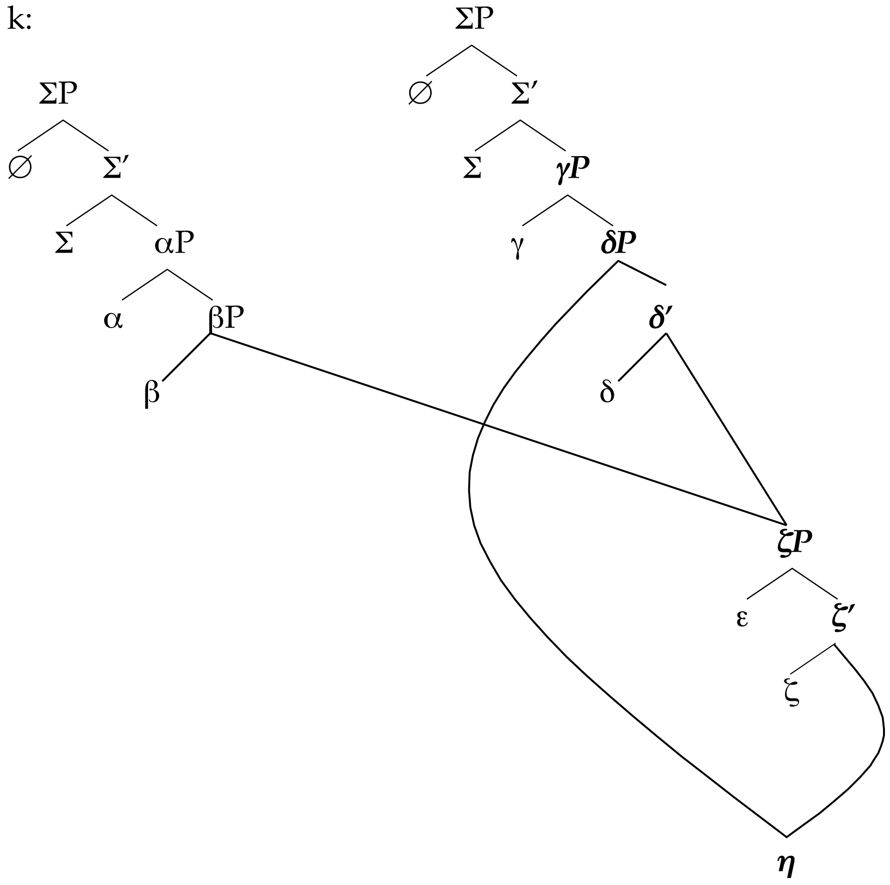
should be kept in mind that in order to restrict merge across subderivations to root nodes, one has to make a further assumption (i.e. adding a constraint), therefore complicating the theory. So, we should not do it unless we have to; and, as far as syntactic amalgams go, such further assumption would just prevent us from getting the desired results. As for the second issue, I think that the absence of intersection between subnumerations in Chomsky's (2000, 2001a) system conflicts with his idea that each subnumeration corresponds to a local computation that is blind to what goes on outside it. Once two (or more) numerations share some lexical tokens, then a link between parallel subderivations is established, while keeping the computations local. As for the third issue, the question does not arise to the extent that the difference between merge and remerge has no theoretical status whatsoever in my system.

At first blush, the remerge of **ζP** in derivational step (111j) appears to violate the c-command condition on (re)merge, since **ζP** does not seem to c-command **δ** in the input structure.

However, the material that dominates the shared constituent (i.e. **ζP**) without dominating the target of the lowering operation (i.e. **δ**) is all built up from lexical tokens that are not part of the same numeration from which the subservient clause (= 100b) is being built (i.e. Δ). So, in relatively to the step when the shared constituent **ζP** is about to be inserted inside **δP** as the new sister of **δ**, the computational system cannot detect anything that dominates **ζP** in the other parallel derivation previously, given that only syntactic material built from lexical tokens of the relevant numeration is visible. Consequently, for all intents and purposes, **ζP** does c-command **δ** in (111i), through vacuous satisfaction, as discussed in **§IV.4**.

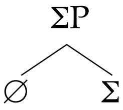
Finally, **η** is remerged in its 'D-structure position' (so to speak) inside the shared constituent **ζP**, as shown in (111k). After that, spell-out applies, delivering the string **#γ#$^\frown$#η#$^\frown$#δ#** to PF.

(111) k:

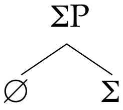


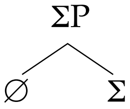
That said, we must address the issue of whether (111) is the only possible derivation for the target representation (99) from the same input. One of the many logical possibilities that must be taken into consideration is the derivation in (112), which is an extreme case of a derivation where the computational system keeps switching back-and-fourth from one numeration to the other (instead of focusing on one numeration, going as far as it can go there, and then shifting to the next numeration for good, as in (111)).

(112) a:

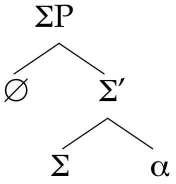


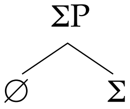
b:



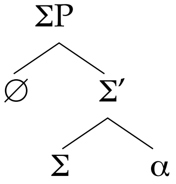


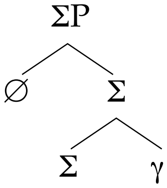
c:



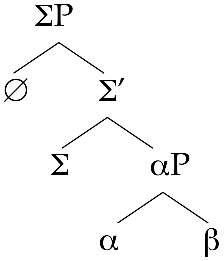


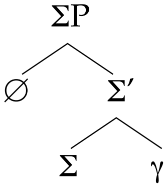
d:



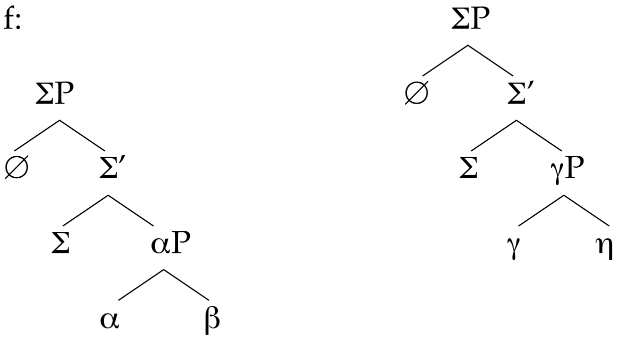


e:

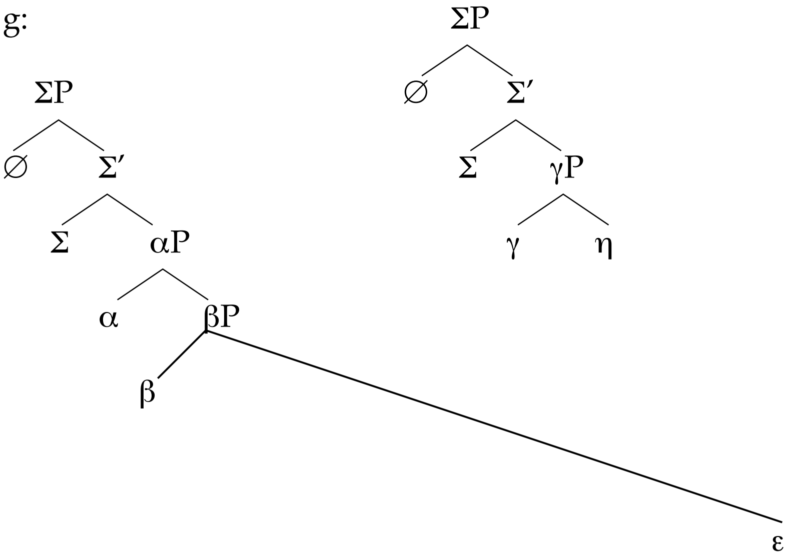




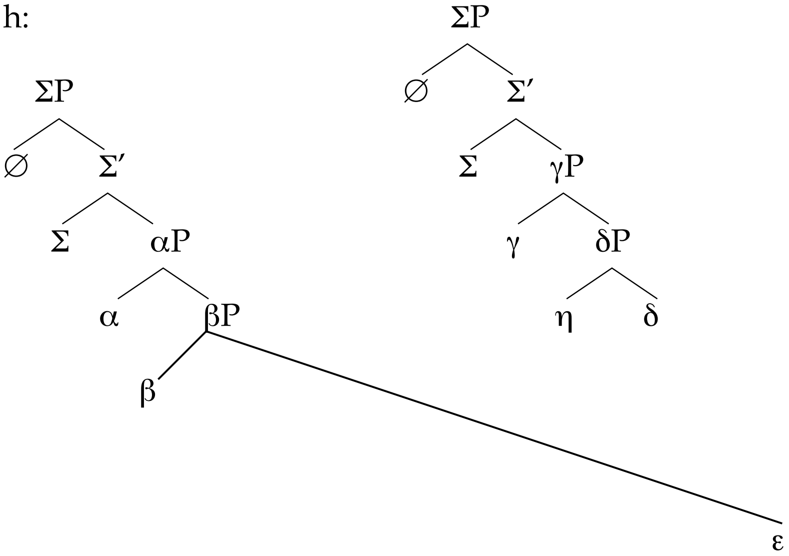
f:


ΣP
∅
Σ′
Σ
αP
α
β
ΣP
∅
Σ′
Σ
γP
γ
η

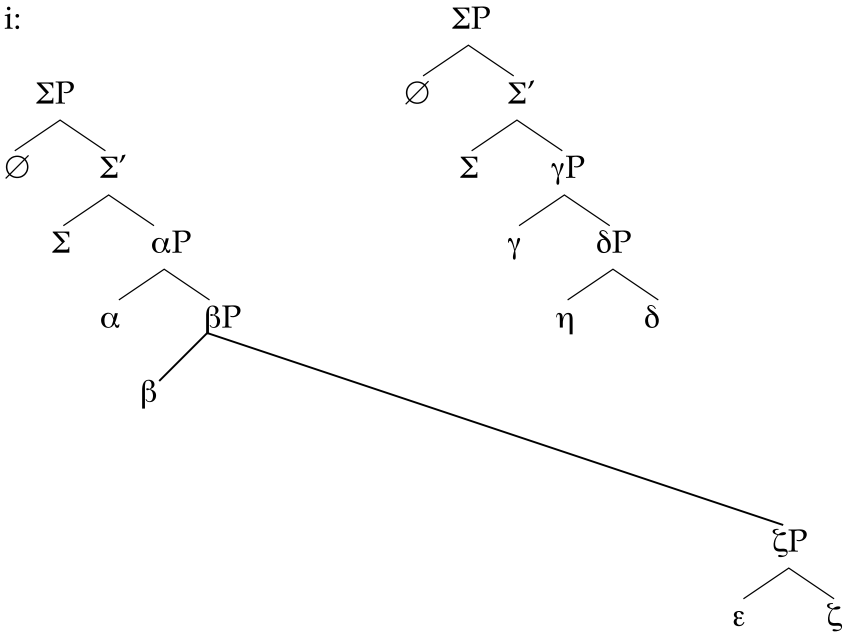
g:


ΣP
∅
Σ′
Σ
αP
α
βP
β
ε
ΣP
∅
Σ′
Σ
γP
γ
η

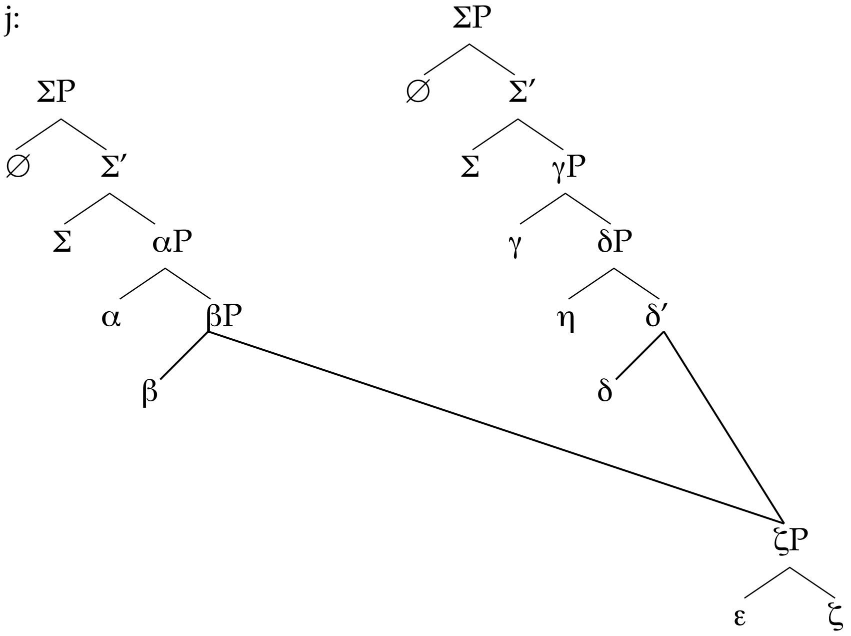
h:


ΣP
∅
Σ′
Σ
αP
α
βP
β
ε
ΣP
∅
Σ′
Σ
γP
γ
δP
η
δ

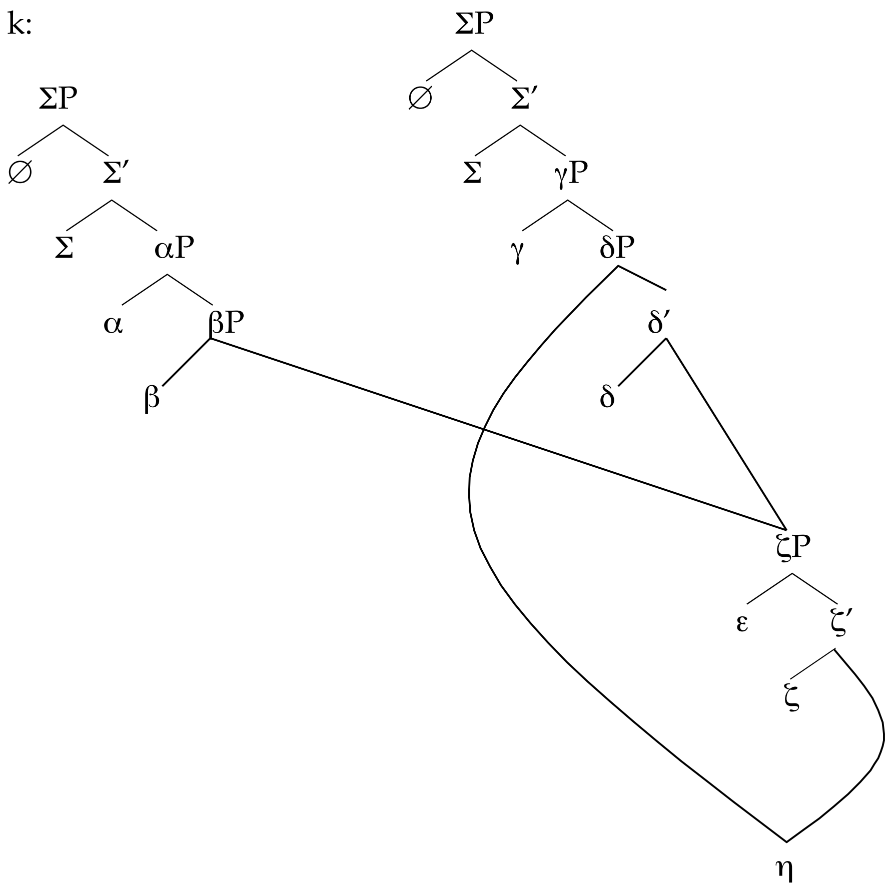
i:


ΣP
∅
Σ′
Σ
αP
α
βP
β
ΣP
∅
Σ′
Σ
γP
γ
δP
η
δ
ζP
ε
ζ

j:


ΣP
∅
Σ′
Σ
αP
α
βP
β
ΣP
∅
Σ′
Σ
γP
γ
δP
η
δ′
δ
ζP
ε
ζ

k:



Given the assumptions about word order presented in **§IV.3**, which determine that terminals be pronounced in the same order they access the derivation, the corresponding PF string for (111) would be as in (113a), whereas the corresponding PF string for (112) would be as in (113b). In a nutshell, both derivations produce the same final LF representation, but each one produces a distinct PF-string.

(113) a: α β ε ζ γ η δ

b: α γ β η ε δ ζ

This becomes very problematic when we start dealing with concrete cases of amalgamation, since this freedom of being able to switch back-and-fourth from numerations ultimately leads to the wrong prediction that, for any given multiply-rooted phrase marker, there are many possible corresponding word order patterns.

For instance, if (96) above were to be generated along the lines of (112), the expected PF-string would be as in (114b), rather than as in (114a (=95)).

(114) a: Marge said that Homer will give you can imagine what to Lisa.

b: * Marge you said can that imagine what Homer will give to Lisa.

Thus, in a nutshell, there must be some sort of constraint in the system, preventing derivations from going back-and-fourth across numerations.

That can be taken to be a consequence of some deeper principle of reduction of computational complexity, since breaking down the global computation into separate derivational flows restricted each one to a numeration is a straightforward way of limiting the search space of computations.

Appendix to Chapter IV

1. Top-to-Bottom Derivations and the Syntax-Phonology Interface

As stated in **§IV.3.1**, one of the main motivations for adopting a derivational top-to-bottom approach to syntax is methodological. By exploring the limits of grammatical theorizing in that way, we can shed some light on the representationalism–*versus*–derivationalism debate, pointing out some facts that can help us to tease apart these two approaches that appear to be fully inter-translatable at first blush. As I said before, once the directionality of derivation is reversed, we make predictions that, *ceteris paribus*, no representational approach can make. Moreover, these predictions seem to be consistent with the facts.

In this Appendix, I am concerned with a syntax-phonology interface phenomenon that constitutes some evidence that syntactic structure should be built in a top-to-bottom fashion.

Although the position of prosodic and syntactic boundaries with respect to each other reveals that the syntax-phonology interface involves no absolute isomorphism, the mismatching at the surface level is better understood in derivational terms as a 'relativized isomorphism'. By that I mean that the core prosodic units that define the domains of prosodic phrasing always correspond to syntactic constituents at the relevant derivational point, namely: when Spell-

Out applies, delivering the phonological material of syntactic phrases to the relevant interface.

The interaction between economy and legibility principles forces syntax to interface phonology in cascades (cashing out chunks of structure), rather than in a single step at the end of the derivation (cashing out the whole syntactic structure at once), or after every merge (cashing out each terminal in isolation). After each phonological cascade falls, its isomorphic syntactic counterpart does not leave the derivation. Rather, it continues being processed by the syntactic component, and may have parts of its constituency relations modified in such a way that it ends up being no longer isomorphic to its phonological counterpart.

I argue that intonational phrases can be taken as an accurate diagnostic for figuring out the exact shape of these PF-chunks that emerge from phonological cascades, and that constitute fossils of extinct syntactic phrases.

2. The Facts

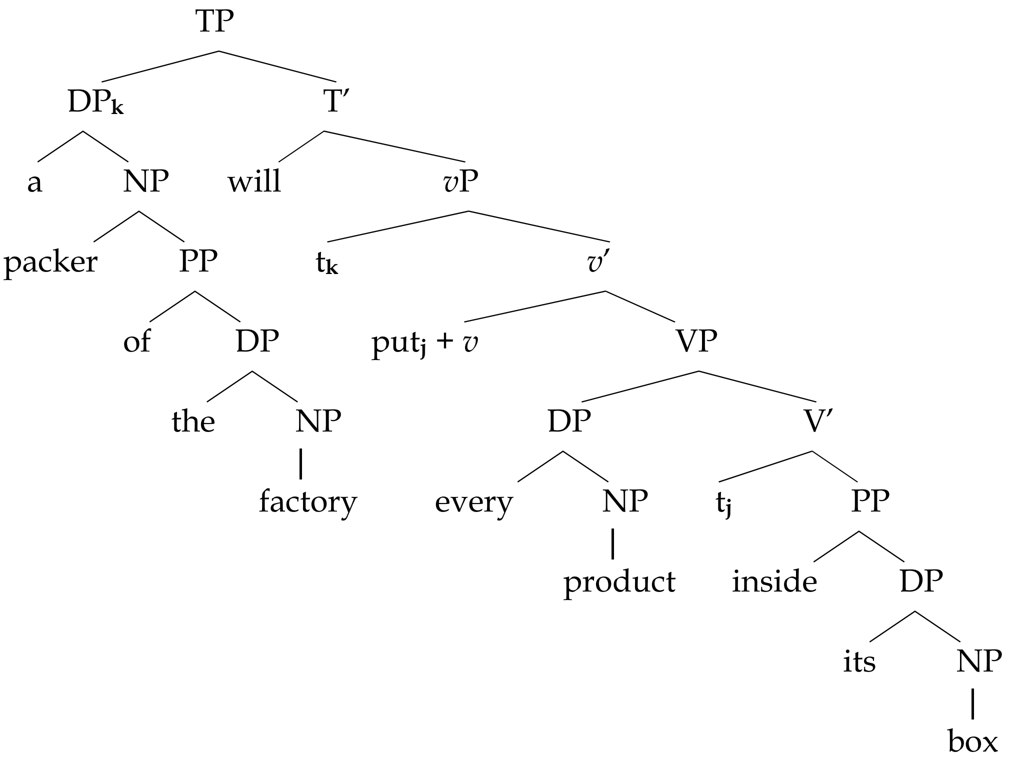
It is a robust fact about human languages that (the phonological component of) UG allows a certain flexibility on the shape of intonational phrases[120]. For example, the strings of words of the sentence in (01) – whose syntactic structure is assumed to be (02) – can either be prosodically parsed in a single

[120] A precise definition of intonational phrase is not necessary here. Roughly speaking, the phonetic correlates of the intonational phrase are (i) lengthening of its last syllable(s), and/or (ii) tendency towards the occurrence of pauses in its initial and final boundaries, and/or (iii) the existence of a complete melodic contour circumscribed in its limits, and/or (iv) maintenance of constant patterns of rate of speech and tessitura in its domain.

intonational phrase, as in (03), or be partitioned into a sequence of intonational phrases in many different ways. Some of them are shown in (04) [121].

(01) A packer of the factory will put every product inside its box.

(02)



(03) ❙ a packer of the factory will put every product inside its box ❙

(04) a: ❙ a packer of the factory ❙ will put every product inside its box ❙

b: ❙ a packer of the factory will put every product ❙ inside its box ❙

[121] Of course, some possible partitioning strategies are (or tend to be) associated with particular readings, contrasting with each other with respect to informational structure, although sharing the same propositional structure (see Steedman 1991a, 1991b, 1999, 2001, on the matter). I will abstract away from this complication here.

c: ▌a packer of the factory ▌will put every product ▌inside its box ▌

d: ▌a packer Ì of the factory ▌will put Ì every product ▌inside its box ▌

Nonetheless, this flexibility is not unrestricted. Some intonational phrasing strategies are ungrammatical (no matter what the intended reading is), like the ones in (05).

(05) a: * ▌a packer of the factory will ▌put every product Ì inside its box ▌

b: * ▌a packer of the factory Ì will put ▌every product inside Ì its box ▌

c: * ▌a packer Ì of the factory will put ▌every product Ì inside its box ▌

d: * ▌a packer of the factory Ì will put ▌every product inside its box ▌

The offending intonational phrase in (05-a) is ▌**a packer of the factory will** ▌. The ungrammaticality of (05-b) is due to ▌**every product inside** ▌. In (05-c), the problem is in ▌**of the factory will put** ▌. Finally, (05-d) is ruled out because of ▌**every product inside its box** ▌.

At first sight, it seems that this restriction can be straightforwardly accounted for in simple syntactic terms. That is, the mapping from syntactic phrase-markers to prosodic constituents requires some kind of isomorphism. If we look at (05-a), (05-b) and (05-c), we see that their respective offending intonational phrases do not correspond to any syntactic constituent in (02). Things are much more complex, however. A closer look at the format of licit and illicit intonational phrases reveals that the mapping from syntactic phrases to intonational phrases does not involve strict isomorphism under any representational view of syntactic constituency. On one hand, **❙ will put every product ❙** in (04-c) is a well-formed intonational phrase regardless it not being isomorphic to any syntactic constituent in (02). On the other hand, **❙ <u>every product inside its box</u> ❙** in (05-d) is an ill-formed intonational phrase even though it is isomorphic to a syntactic constituent in (02): namely, the lower VP of the VP-shell, including a trace of the verb, and both objects.

3. Phonology-Semantics Interface?

This absence of isomorphism had lead Selkirk (1984: 286-296) to postulate the existence of the *Sense Unit Condition*, defined below.

(06) **<u>The Sense Unit Condition on Intonational Phrasing</u>**: (Selkirk 1984)
The immediate constituents of an intonational phrase must together form a sense unit.

(07) An *immediate constituent of an intonational phrase* IntP$_i$ is a syntactic constituent contained entirely within ("dominated" exclusively by) IntP$_i$ and not dominated by any other syntactic constituent contained entirely within IntP$_i$.

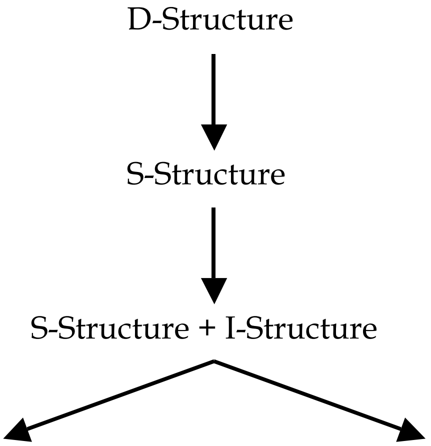
(08) Two constituents C_i, C_j form a sense unit if (a) or (b) is true of the semantic interpretation of the sentence:

a: C_i modifies C_j (a head)

b: C_i is an argument of C_j (a head)

This principle is based on prosodic, syntactic, and semantic notions at the same time, and it is not clear how the relevant relations are computed. For example, what does it mean for a syntactic category to be dominated by a prosodic category? Moreover, if we adopt something like the *Sense Unit Condition*, we clearly need to assume an alternative architecture for UG, like (09) or (10), and dump many minimalist assumptions, like the inclusiveness condition, the absence of S-Structure, and the lack of interaction between interface levels responsible for sound and meaning. Although this might be true, certainly it is not the null hypothesis, and we should try another way of handling the effects of the *Sense Unit Condition* if we can.

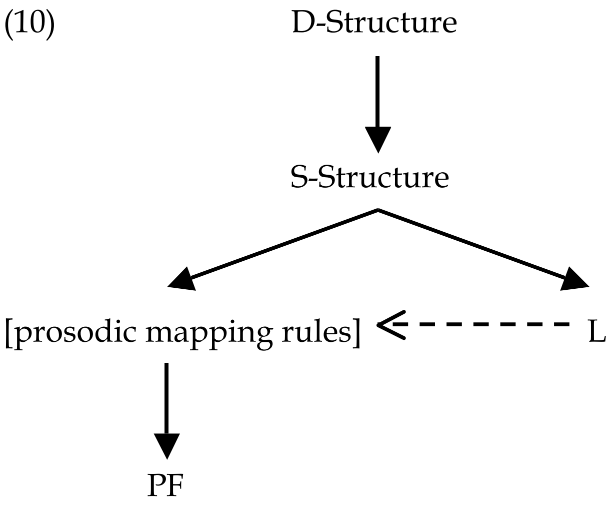
(09)



(Selkirk 1984)

PF LF

(10) (Vogel & Kenesei 1987)



4. The Input to Prosodic Phrasing as a Super-String

4.1. The Factored LCA Hypothesis (Guimarães 1998)

In face of that, I proposed in Guimarães (1998) that the effects of the *Sense Unit Condition* can be captured in a straightforward way if we assume a version of the Minimalist Program in which the input to prosodic phrasing (= output from linearization) is not a flat string of words, like (11), but a super-string (*i.e.* a string of strings of words), like (12), in which overt terminals are linearized with respect to each other, forming kernel strings (= phonological clauses) on the basis of c-command relations among them; and these kernel strings are linearized with respect to each other on the basis of c-command relations involving the non-terminals that dominate the terminals represented in the kernel strings.

Roughly speaking, for any two overt terminals x & y, such that x is pronounced immediately before y, they belong to the same kernel string (= phonological clause) if and only if x asymmetrically c-commands y in the syntax.

(11) <a, packer, of, the, factory, will, put, every, product, inside, its, box>

(12) <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>

In order for this system to account for the facts, the only further assumption that we have to make is the one in (13), which is far way more trivial than Selkirk's *Sense Unit Condition*.[122]

(13) **Constraint on the Shape of Intonational Phrases**:[123] (naïve definition)[124]

[122] Of course, the use of the expression *"far away more trivial"* is appropriate only if we find a way of naturalizing the notion of super-string, arguing that it follows from an independent property of the grammar, which is precisely my goal here.

[123] The constraint in (13) is intended to be just a general mapping function, not a specific algorithm that executes it. In principle, this can be formalized either as an alignment constraint (McCarthy & Prince 1993) in the OT framework (Prince & Smolensky 1993), or as a procedural mechanism of generating prosodic constituents from syntactic outputs.

[124] See Guimarães (1998: chapter IV) for a formal definition of (13), taking into consideration the whole prosodic hierarchy (including prosodic words, phonological phrases, metrical grid, *etc.*), so that the ungrammaticality of *** ❙ a packer ❙ of the factory ❙ will put every ❙ product ❙ inside its box ❙** is explained in terms of a bracketing paradox involving another level of the prosodic hierarchy.

> There must be no bracketing paradox involving phonological clause boundaries and intonational phrase boundaries.

If this is correct, then all the facts shown in the previous section follow straightforwardly, as we can see in (14), (15) and (16) below.

(14) <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>
❙ a packer of the factory will put every product inside its box ❙

(15) a: <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>
❙ a packer of the factory ❙ will put every product inside its box ❙

b: <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>
❙ a packer of the factory will put every product Ì inside its box ❙

c: <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>
❙ a packer of the factory ❙ will put every product ❙ inside its box ❙

d: <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>
❙ a packer Ì of the factory ❙ will put Ì every product ❙ inside its box ❙

(16) a: <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>
* ❙ a packer of the factory will ❙ put every product inside its box ❙

b: <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>
* ❙ a packer of the factory ❙ will put ❙ every product inside Ì its box ❙

c: <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>

* ❙ a packer ❙ of the factory will put ❙ every product ❙ inside its box ❙

d: <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>

* ❙ a packer of the factory ❙ will put ❙ every product inside its box ❙

But how do we get the super-strings like the one in (12)? If we adopt mainstream Minimalism (*i.e.* the bottom-up derivational approach of Chomsky (1995, 2000, 2001a, 2001b)), we need to add extra technology to the system in order to get the super-strings.

One way of doing this is redefining the LCA in such a way that it generates super-strings with the desired shape, instead of flat strings. This is what I did in Guimarães (1998: chapter IV). The basic idea is to factor out the base and the induction steps of the post-kaynean version of the LCA in (17) into two distinct algorithms (18) and (19), which are logically ordered.

(17) **Linear Correspondence Axiom:**

For every terminal elements ***x*** & ***y***, ***x*** precedes ***y*** if and only if (i) or (ii):

i: ***x*** asymmetrically c-commands[125] ***y***;

ii: ∃ Z | Z dominates[126] ***x*** & Z asymmetrically c-commands ***y***.

(18) **Algorithm of Linearization of Terminals**: (ALT)

Linearize all overt terminals of the input phrase marker in one or more strings, such that ***x*** can precede ***y*** if and only if ***x*** asymmetrically c-commands ***y***.

(19) **Algorithm of Linearization of Strings**: (ALS)

Given a set K of strings of overt terminals generated by the ALT, generate a super-string linearizing all members of K with respect to each other, such that, for every L and M which are members of K, L precedes M if and only if ∃ *w*, ∃ *z* | [[[*w* is a symbol of L] & [*z* is a symbol of M]] & [∃ Q | [Q dominates *w*] & [Q asymmetrically c-commands *z*]]]

The first algorithm (*i.e.* 18) generates a set of strings of words, as in (20)[127]; then the second one (*i.e.* 19) linearizes those strings with respect to each other, generating a super-string, as in (21).

(20) { [a⁀packer⁀of⁀the⁀factory], [will⁀put⁀every⁀product], [inside⁀its⁀box] }

[125] *C-Command*: Given two maximal and/or minimal projections α & β, α c-commands β if and only if (i) α ≠ β & (ii) no segment of α dominates β & (iii) every category that dominates α also dominates β.

[126] *Dominance*: Given a syntactic object K = {γ, {α, β}}, K dominates a syntactic object α if and only if either (i) ∃ L | α ∈ L & L ∈ K or (ii) ∃ M | K dominates M & M dominates α.

[127] As I discuss in Guimarães (1998: 162-171), there is always more than one legitimate output from the ALT (for example, in the case under discussion, it could be { [a⁀packer⁀factory], [⁀of⁀the], [will⁀put], [every⁀product], [inside⁀its⁀box] } instead of (20)). However, it is always the case that, among all potential outputs from the ALT, only one constitutes a legitimate input to the ALS. If the wrong choice is made, the derivation is cancelled.

(21) [a$^\frown$packer$^\frown$of$^\frown$the$^\frown$factory]$^\frown$[will$^\frown$put$^\frown$every$^\frown$product]$^\frown$[inside$^\frown$its$^\frown$box]

Despite being so cumbersome, this mechanics makes the right predictions, by and large (see Guimarães (1998) for details). The problem is that it does not follow from anything.

4.2. An Alternative Approach within Mainstream Minimalism

Another way of getting the same result is assuming the standard version of the LCA in (17) and positing an additional mapping algorithm that applies after the linearization procedure, converting flat strings like (11) – repeated below as (22) – into super-strings with the desired shape, like (12) – repeated below as (23) – on the basis of syntactic information encoded in the phrase-marker: the same object that serves as the input to linearization. The basic idea remains the same: for any two overt terminals ***x*** & ***y***, such that ***x*** is pronounced immediately before ***y***, they belong to the same kernel string (= phonological clause) if and only if ***x*** asymmetrically c-commands ***y*** in the syntax.

(22) <a, packer, of, the, factory, will, put, every, product, inside, its, box>

(23) <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>

Of course, we can formulate such mapping algorithm that gets us from (22) to (23)[128]. But this is not going to be less inelegant than (18-19), and, again, it does not follow from anything in the system. Moreover, it sounds counterintuitive to have two distinct mapping algorithms based on the very same syntactic information. That is, if two adjacent words are linearized with respect to each other through the base step of the LCA, there is no phonological clause boundary between them. If they are linearized with respect to each other through the induction step of the LCA, there is a phonological clause boundary between them[129].

This might be right or wrong, but, certainly, it is not the null hypothesis. From the minimalist viewpoint, we better collapse these two algorithms into a single one if we can, as I did in Guimarães (1998).

4.3. Inadequacy of Bottom-up Multiple Spell-Out

In face of that, one may think that the desired result can come for free if we assume Uriagereka's (1999a) original version of the Multiple Spell-Out model. After all, chunks of structure defined on the basis of c-command is what it is all about[130]. However, it is easy to see that the boundaries of the chunks created by Spell-Out in a bottom-up system will not be any helpful. Notice that the offending intonational phrases in (27-a), (27-b) and (27-c) are no different from the well-formed intonational phrase **▌ will put every product ▌** in (26-c) with respect to the principle in (13) above.

[128] One possibility would be that there is an algorithm that starts from a flat string without boundary symbols (understood as substantive entities, dummy prosodic formatives, a la Chomsky & Halle 1968), and inserts them in between two adjacent terminals if and only if their copies left in the phrase marker are not in asymmetric c-command relation.

[129] Keep this in mind: technically speaking, at the relevant level of abstraction, after the super-string (23) is generated from (22), *factory* does not precede *will* anymore, even though the former happens to be pronounced immediately before the latter, by virtue of them being the last and the first symbols the strings A (=a⁀packer⁀of⁀the⁀factory) & B (=will⁀put⁀every⁀product) respectively, such that A immediately precedes B.

[130] Here I am assuming that my audience is completely familiar with Uriagereka's (1999a) work.

(24) <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

(25) <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

Ì a packer of the factory will put every product inside its box Ì

(26) a: <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

❚ a packer of the factory ❚ will put every product inside its box ❚

b: <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

❚ a packer of the factory will put every product ❚ inside its box ❚

c: <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

❚ a packer of the factory ❚ will put every product ❚ inside its box ❚

d: <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

❚ a packer ❚ of the factory ❚ will put ❚ every product ❚ inside its box ❚

(27) a: <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

* ❚ a packer of the factory will ❚ put every product inside its box ❚

b: <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

* ❚ a packer of the factory ❚ will put ❚ every product inside ❚ its box ❚

c: <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

* ❙ a packer ❙ of the factory will put ❙ every product ❙ inside its box ❙

d: <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

* ❙ a packer of the factory ❙ will put ❙ every product inside its box ❙

If we assume Uriagereka's original version of the Multiple Spell-Out model, we need an additional mapping algorithm that generates (29) from (28), in order to get the facts through a trivial 'avoid-bracketing-paradox' assumption. Again, this mapping function, besides being formally very complex, does not seem to follow from anything.

(28) <<a, packer, of, the, factory>, will, put, <every, product>, inside, its, box>

(29) <<a, packer, of, the, factory>,<will, put, every, product>,<inside, its, box>>

4.4. The 'Back-and-Fourth Derivation' Hypothesis

A third way of getting the desired super-strings is redefining the LCA as a mapping algorithm that builds strings and super-strings step by step, removing (demerging) the terminals from the phrase-marker in a top-to-bottom fashion, and organizing them into strings (see Fukui & Takano 1998, for an approach along these lines). Roughly speaking, this procedure would work as follows.

First of all, the least embedded overt terminal A is targeted, demerged, and turned into the first symbol of a PF string. Second of all, the next overt terminal in the c-command path of A (say, B) is targeted, demerged and concatenated to the right of A. Then, the next overt terminal in the c-command path of B (say, C), is targeted, demerged and concatenated to the right of B... and so on. This is a trivial markovian process that terminates every time that the last demerged element has no overt terminal in its c-command path. If the set of overt terminals of the phrase marker is not exhausted, this markovian process is required to apply again, now starting from the least-embedded overt terminal among the ones that remain merged. It seems plausible to assume that, every time the markovian procedure (re)starts, a dummy phonological formative # (a la Chomsky & Halle 1968) is inserted as the first symbol of the string, and then the first demerged element is concatenated to the right of # (and perhaps, another instance of # is inserted right after the last demerged element of each round). If this is so, then we get a flat string with the relevant properties of the super-string proposed by Guimarães (1998), and all the rest follows.

(30) #$^\frown$a$^\frown$packer$^\frown$of$^\frown$the$^\frown$factory$^\frown$#$^\frown$#$^\frown$will$^\frown$put$^\frown$every$^\frown$product$^\frown$#$^\frown$#$^\frown$inside$^\frown$its$^\frown$box$^\frown$#

In this system, syntax works in a bottom-up fashion, whereas the syntax-to-phonology mapping proceeds the other way around, from the top downwards. From now on, I will refer to this as the 'Back-and-Forth Derivational Hypothesis'. Notice that the mapping algorithm just described follows from nothing, and is not in any sense more natural than the ones required in the systems mentioned above.

4.5. Top-to-Bottom Derivations, Dynamic Constituency and Relativized Isomorphism

There seems to be robust evidence that PF-structure-building mechanism really works in this incremental, left-to-right/top-to-bottom fashion. Thus, what if the syntax also builds structure this way? Then, the way the phonological component works would follow from the way the syntactic component works, and we can kill two birds with one stone.

Once we assume a 'generalized tucking-in' system like the one proposed in this dissertation, where constituency is dynamic, we start to make correct predictions with regards to the boundaries of PF-strings.

Having said that, let us run a sample derivation from the very beginning till the very end (abstracting away from movement), and see how the system works. I will take the same sentence used to illustrate the generalizations about intonational phrasing above, and show how the desired super-strings emerge from the very nature of the derivation.

(31) A packer of the factory will put every product inside its box.

(32a) derivational stage #1

syntax	$[^{\Sigma P}\ \varnothing\ \Sigma]$
phonology	

(32b) derivational stage #2

syntax	$[^{\Sigma P}\ \varnothing\ [^{\Sigma'}\ \Sigma\ \textit{a}]]$
	#a#
phonology	

(32c) derivational stage #3

syntax	$[^{\Sigma P}\ \varnothing\ [^{\Sigma'}\ \Sigma\ [^{DP}\ \textit{a packer}]]]$
	#a#$^{\frown}$#packer#
phonology	

(32d) derivational stage #4

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{DP} *a* [^{NP} *packer of*]]]]
	#a#$^{\cap}$#packer#$^{\cap}$#of#
phonology	

(32e) derivational stage #5

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{DP} *a* [^{NP} *packer* [^{PP} *of the*]]]]]
	#a#$^{\cap}$#packer#$^{\cap}$#of#$^{\cap}$#the#
phonology	

(32f) derivational stage #6

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]]]]
	#a#$^{\cap}$#packer#$^{\cap}$#of#$^{\cap}$#the#$^{\cap}$#factory#
phonology	

(32g) derivational stage #7

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]]]]
	SPELL-OUT
phonology	#a$^{\cap}$packer$^{\cap}$of$^{\cap}$the$^{\cap}$factory#

Spell-Out is forced to apply at stage #7 because otherwise there would be no way of satisfying the LCA from the next stage onwards[131]. After all π-particles have been removed from the derivational workspace, the next item (*i.e.* the λ-particle of **<u>will</u>**) can access the derivation at stage #8 without causing any linearization problem. Although its corresponding π-particle (*i.e.* #will#) can not be linearized with respect to any other π-particle (by virtue of the older sister of its corresponding λ-particle being a complex phrase), no violation of the

[131] An alternative would be merging *will* to *factory*. Although this strategy would create no linearization problem, it is ruled out on semantic grounds.

Linearity Principle arises, since #will# is the only π-particle in the derivational workspace. That is, the LCA is vacuously satisfied.

(32h) derivational stage #8

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] *will*]]]
	#will#
phonology	#a⁀packer⁀of⁀the⁀factory#

(32i) derivational stage #9

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] [$^{T'}$*will put*]]]]
	#will#⁀#put#
phonology	#a⁀packer⁀of⁀the⁀factory#

(32j) derivational stage #10

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] [$^{T'}$*will* [^{VP}*put every*]]]]]
	/will/⁀/put/⁀/every/
phonology	#a⁀packer⁀of⁀the⁀factory#

(32k) derivational stage #11

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] [$^{T'}$*will* [^{VP}*put* [^{DP} *every* [^{NP} *product*]]]]]]]
	/will/⁀/put/⁀/every/⁀/product/
phonology	#a⁀packer⁀of⁀the⁀factory#

(32l) derivational stage #12

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] [$^{T'}$*will* [^{VP}*put* [^{DP} *every* [^{NP} *product*]]]]]]]
	SPELL-OUT
phonology	<#a⁀packer⁀of⁀the⁀factory#, #will⁀put⁀every⁀product#>

(32m) derivational stage #13

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] [$^{T'}$*will* [^{VP}*put* [^{PP} [^{DP} *every* [^{NP} *product*]] *inside*]]]]]]
	#inside#
phonology	<#a∩packer∩of∩the∩factory#, #will∩put∩every∩product#>

(32n) derivational stage #14

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] [$^{T'}$*will* [^{VP}*put* [^{PP} [^{DP} *every* [^{NP} *product*]] [$^{P'}$ *inside its*]]]]]]]
	#inside#∩#its#
phonology	<#a∩packer∩of∩the∩factory#, #will∩put∩every∩product#>

(32o) derivational stage #15

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] [$^{T'}$*will* [^{VP}*put* [^{PP} [^{DP} *every* [^{NP} *product*]] [$^{P'}$ *inside* [^{DP} *its* [^{NP} *box*]]]]]]]]]
	#inside#∩#its#∩#box#
phonology	<#a∩packer∩of∩the∩factory#, #will∩put∩every∩product#>

(32p) derivational stage #16

syntax	[$^{\Sigma P}$ ∅ [$^{\Sigma'}$ Σ [^{TP} [^{DP} *a* [^{NP} *packer* [^{PP} *of* [^{DP} *the* [^{NP} *factory*]]]]] [$^{T'}$*will* [^{VP}*put* [^{PP} [^{DP} *every* [^{NP} *product*]] [$^{P'}$ *inside* [^{DP} *its* [^{NP} *box*]]]]]]]]]
	SPELL-OUT
phonology	<#a∩packer∩of∩the∩factory#, #will∩put∩every∩product#, #inside∩its∩box#>

5. Concluding Remarks

5.1. Prosodic Hierarchy is built over a super-string, that partially determines the shape of prosodic constituents[132]. That is, prosodic phrasing must respect the boundaries of PF-chunks.

5.2. The format of the PF chunks follows of the very nature of the derivation. PF chunks correspond to syntactic constituents at some point of the derivation (specifically, at the turning point between two phonological cascades), even though they do not correspond to final LF-chunks. So, in a sense, we can say that there is isomorphism between PF chunks and syntactic constituents. We can call it 'relativized isomorphism'. This can be taken as an evidence for the strong derivational hypothesis, since it is very hard to get the same results in strict representational terms.

5.3. We can entirely dispense with the *Sense Unit Condition* and its consequences for the architecture of the grammar.

[132] Here I am concerned with intonational phrases only, but see Guimarães (1998, 1999a) for evidence that the super-string is relevant to other levels of prosodic hierarchy too.

V

The Emergence of Parataxis as 'Syntax Pushed to the Limit'

After having described the phenomenon of amalgamation in **§II**, pointed out the problems with the (neo)conservative analyses in **§III**, and then, in **§IV**, presented the theoretical framework for my proposal, this chapter is dedicated to analyzing the range of facts shown in **§II** in an explanatorily adequate fashion.

V.1. Deriving a Simple Syntactic Amalgam

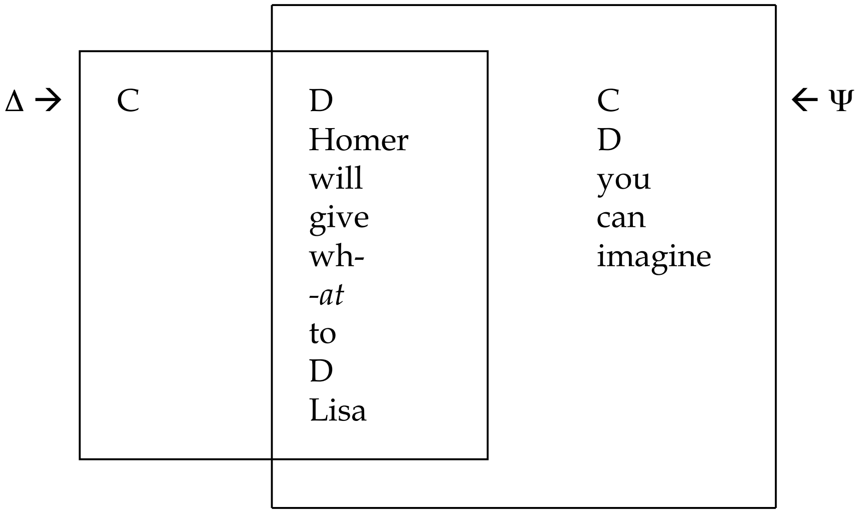
(01) Homer will give you can imagine what to Lisa.

According to the assumptions presented in chapter **IV**, the generation of the syntactic amalgam in (01) involves the intersecting numerations in (02) as the input to the computational system.[133]

[133] Following Bennett (1977: 282), I assume that anything that appears to be a bare WH word at the morpho-phonological level actually corresponds to a much more complex structure at the syntactic and semantic levels, as shown in (i).

(i) a: *who* = [$^{\text{DP}}$ [$^{\text{D}}$ *which*] [$^{\text{N(P)}}$ *person*]]
b: *where* = [$^{\text{DP}}$ [$^{\text{D}}$ *which*] [$^{\text{N(P)}}$ *place*]]
c: *when* = [$^{\text{DP}}$ [$^{\text{D}}$ *which*] [$^{\text{N(P)}}$ *time*]]
d: *why* = [$^{\text{DP}}$ [$^{\text{D}}$ *which*] [$^{\text{N(P)}}$ *reason*]]
e: *how* = [$^{\text{DP}}$ [$^{\text{D}}$ *which*] [$^{\text{N(P)}}$ *manner*]]

(02)



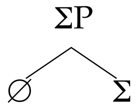
The computational system starts by randomly zooming into numeration Δ to begin the structure-building process. Given the assumptions (tacitly) assumed in chapter **IV** about the nature of the grammar and the parser (heavily inspired by Phillips' (1996, 1998) work), the choice of Δ automatically makes the sentence built from Δ the master clause, with the one built from Ψ being subservient to it. The PF representation of the whole Siamese-Tree structure is built incrementally, as the derivation proceeds, with smaller chunks of each subcomputation getting successively spelled-out and being pronounced with respect to each other in an order that directly reflects the order in which syntax delivers them to the A-P system.

f: *what* = [^{DP} [^{D} *which*] [$^{N(P)}$ *thing*]]

The motivations for taking this approach have to do with the semantic interpretation of syntactic amalgams, which I investigated in Guimarães (2003c).

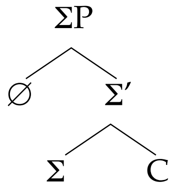
Within the confines of that subcomputation defined by Δ, the first derivational step is the introduction of the initial phrase by the starting axiom, as in (03a).

(03) a:



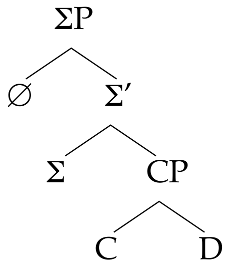
The next step is the introduction of C, which gets tucked in inside ΣP, as the older sister of Σ, as in (03b).

(03) b:

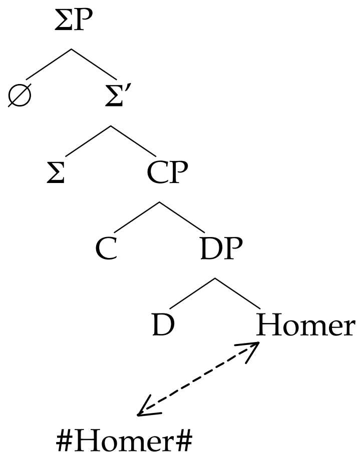


Then, the lexical tokens of Δ are introduced, one by one, and tucked in at the bottom of the phrase marker, each one becoming a (provisory) older sister of the lexical token introduced in the immediately preceding step, as in (03c-d) below.

(03) c:

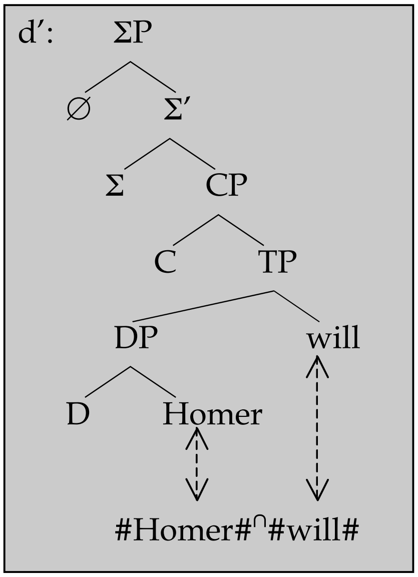


d:



At this point, the natural way to take the next step would be to introduce T as the sister to the DP **<u>Homer</u>**, as in (03d′).

(03) d′:

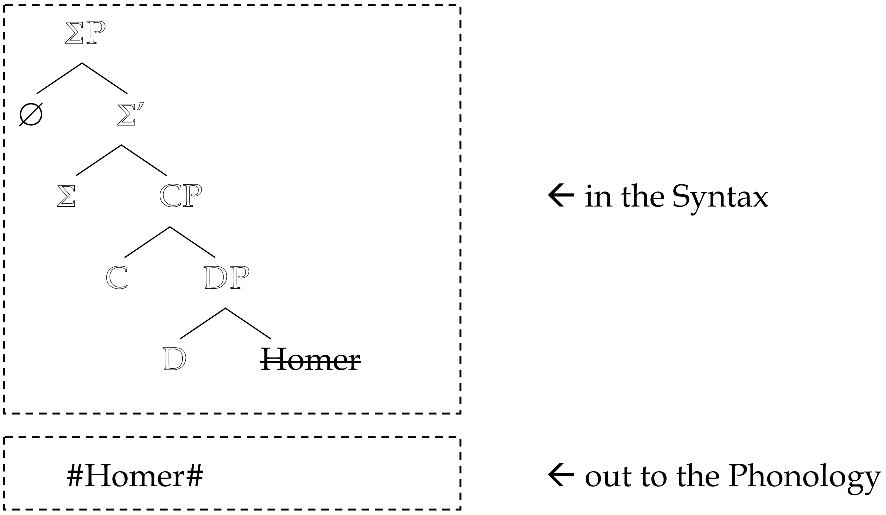


However, that would lead to a violation of the LCA, as defined in **IV.3** Notice that the π-particle of **<u>Homer</u>** (i.e. #Homer#) precedes π-particle of **<u>will</u>** (i.e. #will#) even though **<u>Homer</u>** does not asymmetrically c-command **<u>will</u>**, as it should. But, since selectional properties of the lexical tokens in Δ require that

will be eventually merged as the (temporary) sister of the DP **Homer**, the system needs to ‘prepare’ the derivational workspace first, shipping the current PF-string to the phonological component, leaving the phrase marker ‘naked’, with no π-particle linked to any of its terminals, so that the introduction of **will**, when it happens, will not constitute a violation of the LCA.

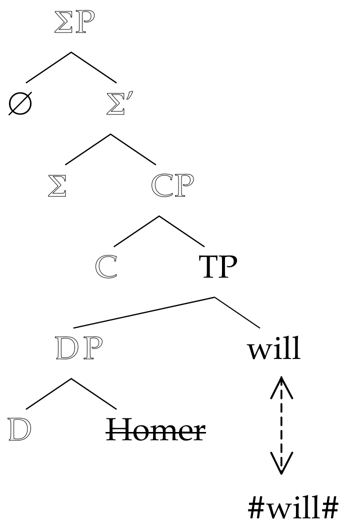
Therefore, for convergence reasons, the step immediately after the one in (03d) is not (03d’). Rather, it is the one in (03e), where the current structure undergoes spell-out

(03) e:



The next step, then, is the introduction of **will** as the (temporary) sister to the DP **Homer**, as in (03f). Notice that, now, #will# is the only π-particle in the derivational workspace. Therefore, the LCA is trivially satisfied in this step.

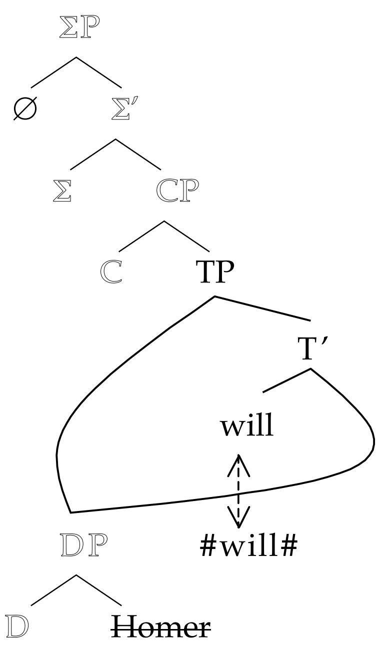
(03) f:



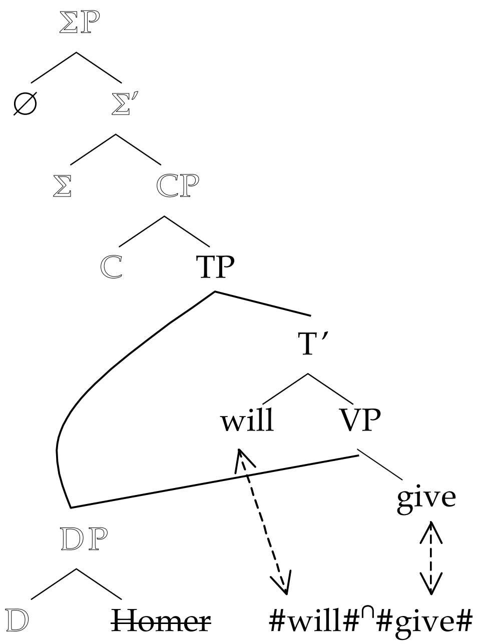
Then, the DP **Homer** (which no longer has phonological material) is remerged as a sister to **will** in (03g),[134] so that the subject can find itself in its theta-position in a subsequent step, after the introduction of **give**, as in (03h).

[134] This may seem counter-intuitive at first sight, since, in (03f), **Homer** is already a sister to **will**. (although, in the previous step, it was **will** which merged to **Homer**). However, this step (which makes **Homer** become simultaneously the complement and the specifier to **will**) is crucial to make it possible for **Homer** to become the specifier of VP in a subsequent step.

(03) g:

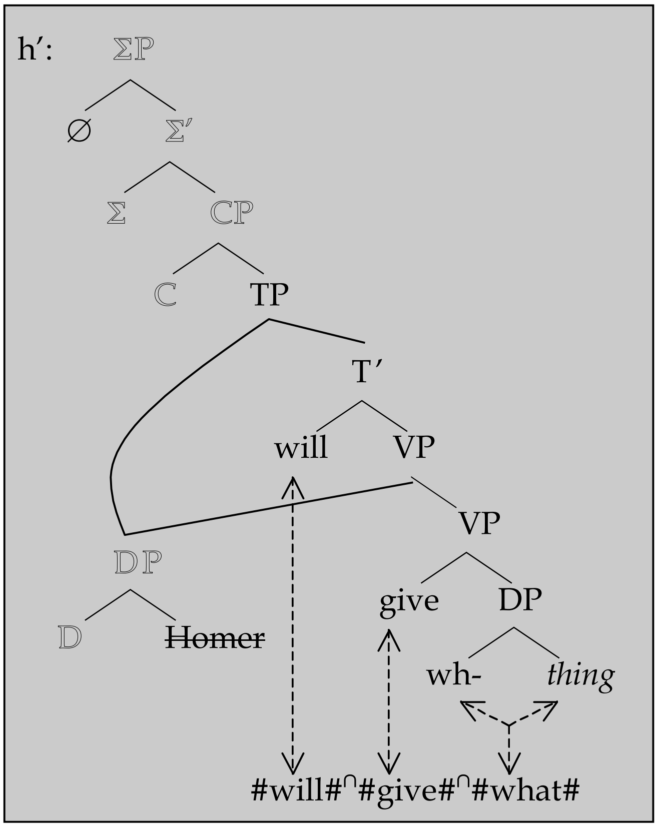


h:



At this point, the natural continuation towards building the sentence corresponding to Δ would be to build the WH-phrase **what**, by tucking in its terminals, one at a time, at the bottom of the spine of the tree, as in (03h′).

(03)

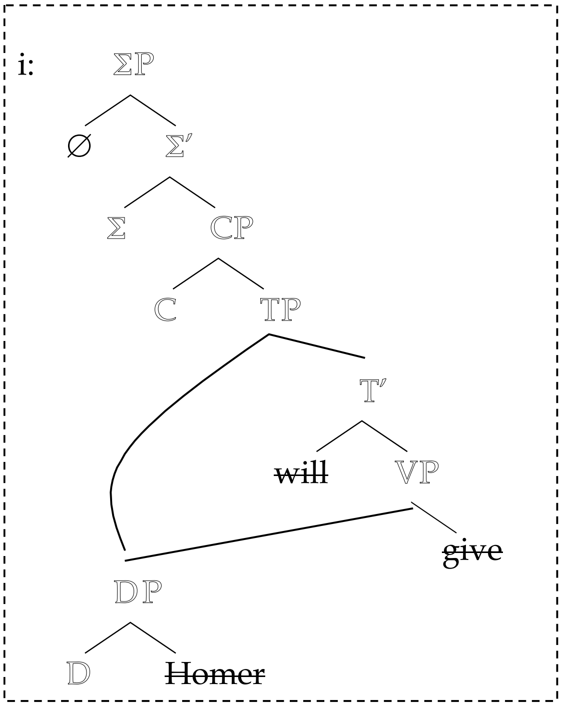


However, if that happens, the WH-phrase what will not be able to be later merged in the lower spec/CP of the subservient clause, since it will fail to c-command that [+WH] complementizer. I will return to this matter shortly (cf. (03r′) below).

The only alternative that could lead to convergence, then, is the termination of the first derivational round, leaving the phrase marker corresponding to Δ as an incomplete structure. This is possible because all relevant lexical tokens necessary to build this chunk of the structure are shared by the numeration in Ψ. That way, the subcomputation performed in the second derivational round can finish the job of building that chunk of structure left incomplete by in the first derivational round.

Therefore, for convergence reasons, the step immediately after the one in (03h) is not (03h′). Rather, it is the one in (03i), where the current structure undergoes spell-out.

(03) i:



← in the Syntax

[#Homer#]$^{\cap}$[#will#$^{\cap}$#give#]

← out to Phonology

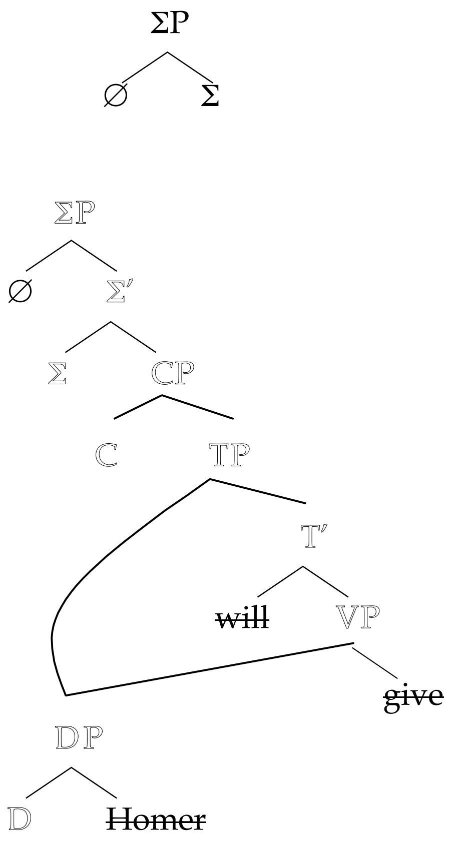
Notice that, in the phonological component, the incoming string (i.e. #will#$^{\cap}$#give#) gets concatenated to the final edge of the previous one, rather than to its initial edge. As discussed in **§IV.3.2**, this deterministic linearization of strings that happens within the Phonological component follows from the deeper derivational time equals real time (Phillips 1996) across components in the grammar. In a nutshell, under the assumption that the 'parser is the grammar', the string **[#Homer#]** is pronounced before the string **[#will#$^\cap$#give#]** simply because arrived at the Phonological component first, getting the first timing slot.

The computational system then shifts its attention to numeration Ψ to continue the structure-building process. As discussed in **§IV.3.4**, The fact that the matrix clause built from Δ is done after the one built from Ψ automatically makes the later ***subservient*** to the former.

The higher portion of the subservient clause is built in the usual fashion, by integrating the relevant lexical tokens in the non-shared part of Ψ, into a parallel phrase marker built from the top downwards, via successive applications of tucking-in, as shown in (03j) though (03z').

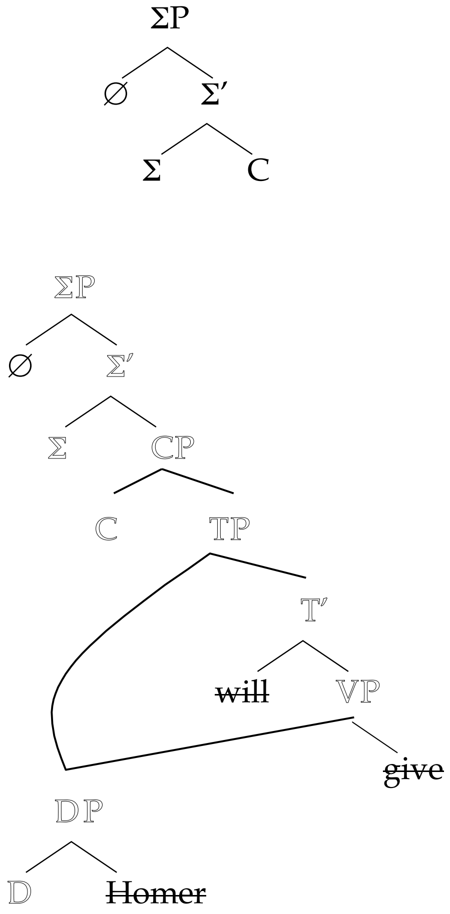
First, the initial phrase is introduced by the starting axiom, as in (03j).

(03) j:



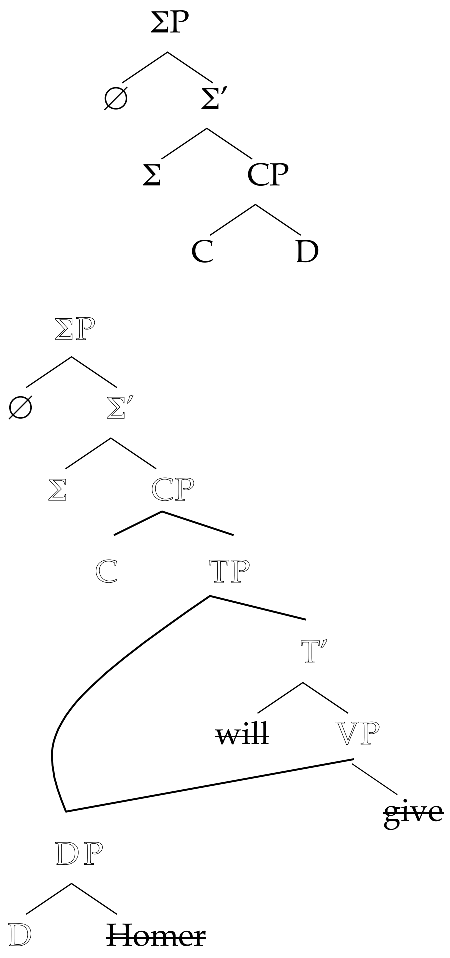
Then the highest complementizer is tucked in, becoming the (temporary) sister of Σ, as in (03k).

(03) k:

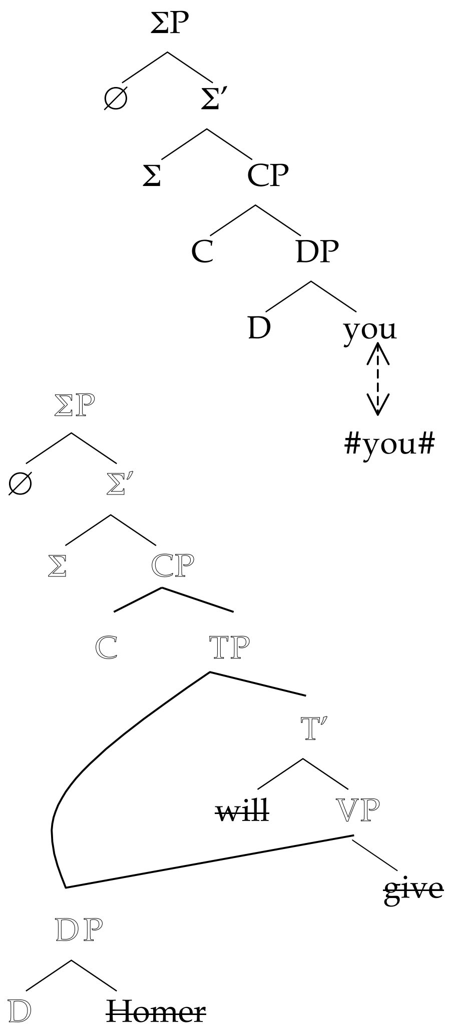


Then the subject is built as a (temporary) sister to C, by first merging D to C, and then you to D, as shown in (03 l-m).

(03) l:

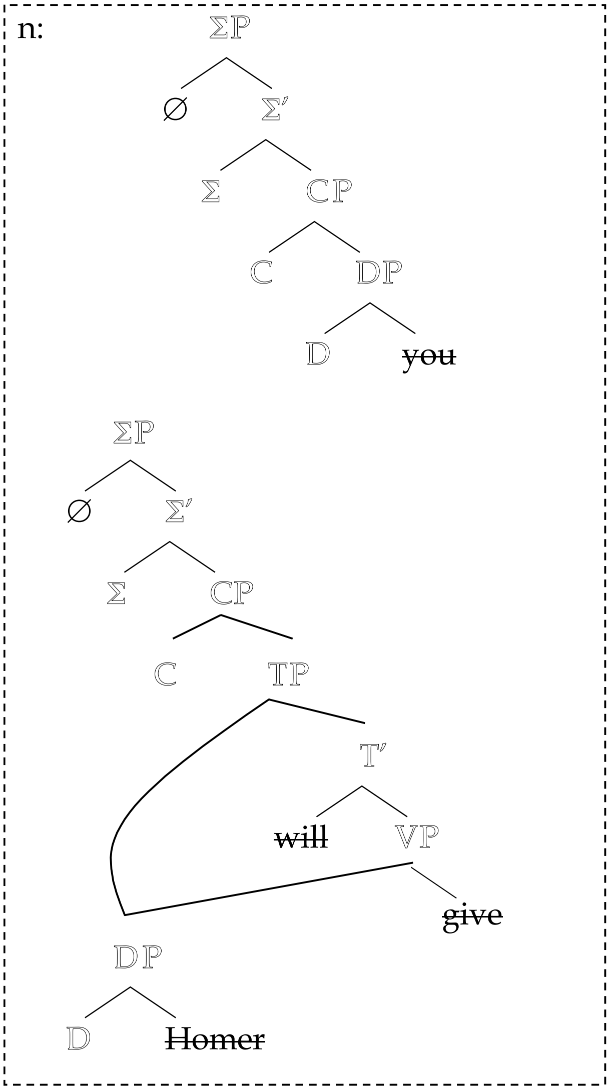


(03) m:



At this point, the DP **you** is about to become a complex specifier, as soon as the head of T is integrated to the phrase marker, making it impossible for the pronounceable terminal inside that subject DP (i.e. #you#) to c-command any other terminal in the rest of the rest of the structure. That would lead to a violation of the LCA. In order to avoid that, the system then needs to apply Spell-Out to the phrase marker under construction, removing the current PF-string from the derivational workspace, as shown in (03n).

(03)



← in the Syntax

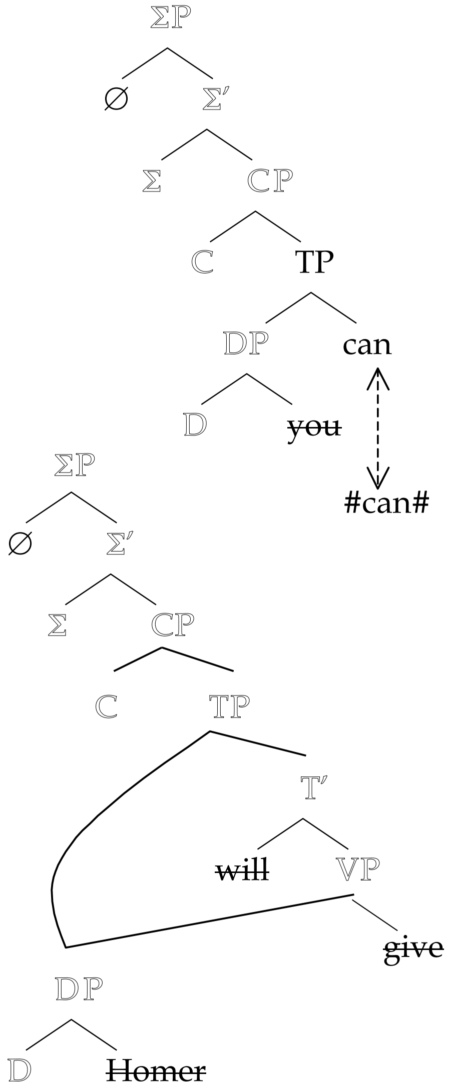
[#Homer#]$^\cap$[#will#$^\cap$#give#]$^\cap$[#you#]

← out to Phonology

In the phonological component, the incoming string (i.e. **[#you#]**) gets concatenated to the final edge of the existing super-string (i.e. **[#Homer#]$^\cap$[#will#$^\cap$#give#]**), rather than to its initial edge, as determined by the 'first come first serve basis mapping algorithm' discussed in **§IV.3.2.**

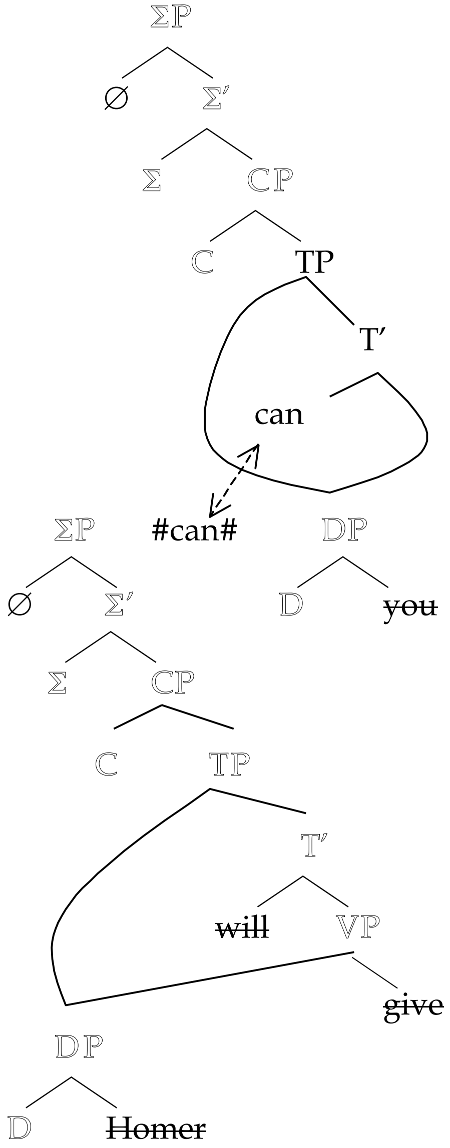
The next step, then, is the introduction of **can** as the (temporary) sister to the DP **you**, as in (03o). Notice that, now, #can# is the only π-particle in the derivational workspace. Therefore, the LCA is trivially satisfied in this step.

(03) o:



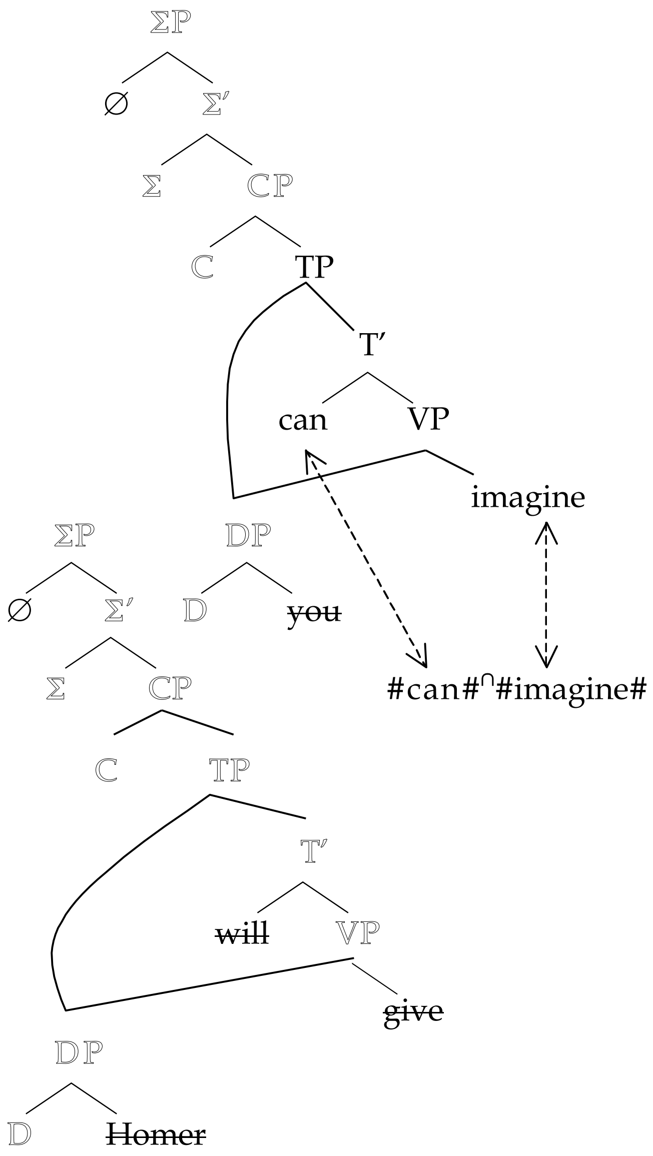
Then, the DP **<u>you</u>** (which no longer has phonological material) is remerged as a sister to **<u>can</u>** in (03p),[135] so that the subject can find itself in its theta-position in a subsequent step, after the introduction of **<u>imagine</u>**, as in (03q).

(03) p:



[135] This may seem counter-intuitive at first sight, since, in (03f), **<u>Homer</u>** is already a sister to **<u>will</u>**. (although, in the previous step, it was **<u>will</u>** which merged to **<u>Homer</u>**). However, this step (which makes **<u>Homer</u>** become simultaneously the complement and the specifier to **<u>will</u>**) is crucial to make it possible for **<u>Homer</u>** to become the specifier of VP in a subsequent step.

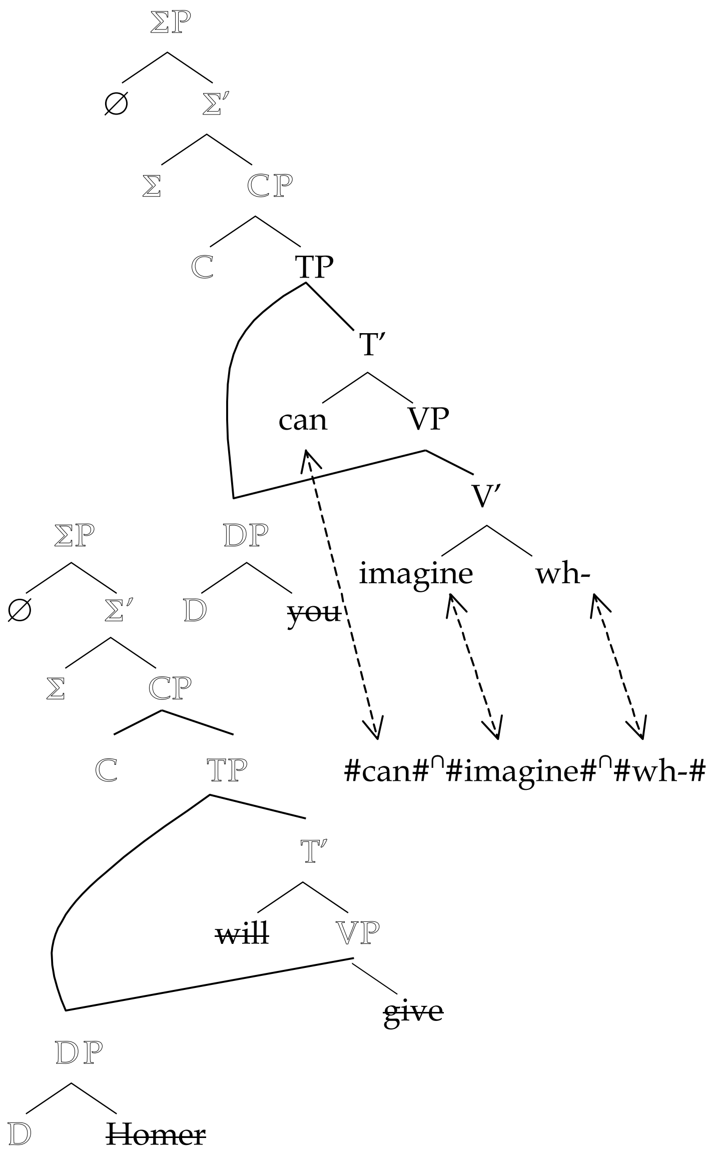
(03) q:



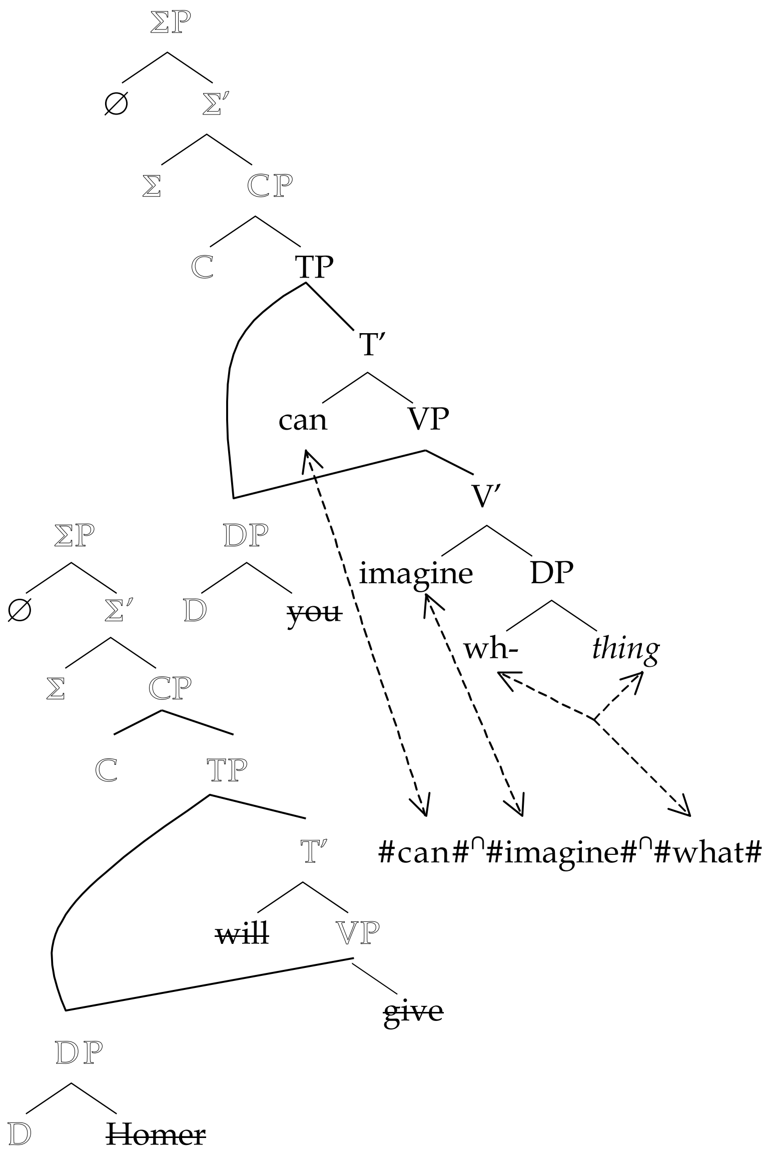
The verb imagine at the bottom of the phrase marker under construction selects for a CP with a [+WH] head. Since that head requires a WH-phrase in its specifier, the introduction of the complementizer must be delayed until the WH-phrase is built as a temporary sister to imagine, as shown in (03r-s).[136]

[136] I am assuming that the morphologization of [^{DP} wh- *thing*] as #what# happens in the phonological component, after the string of π-particles leaves the syntactic derivational workspace. However, I will keep using the notation in (03s) for expository reasons.

(03) r:

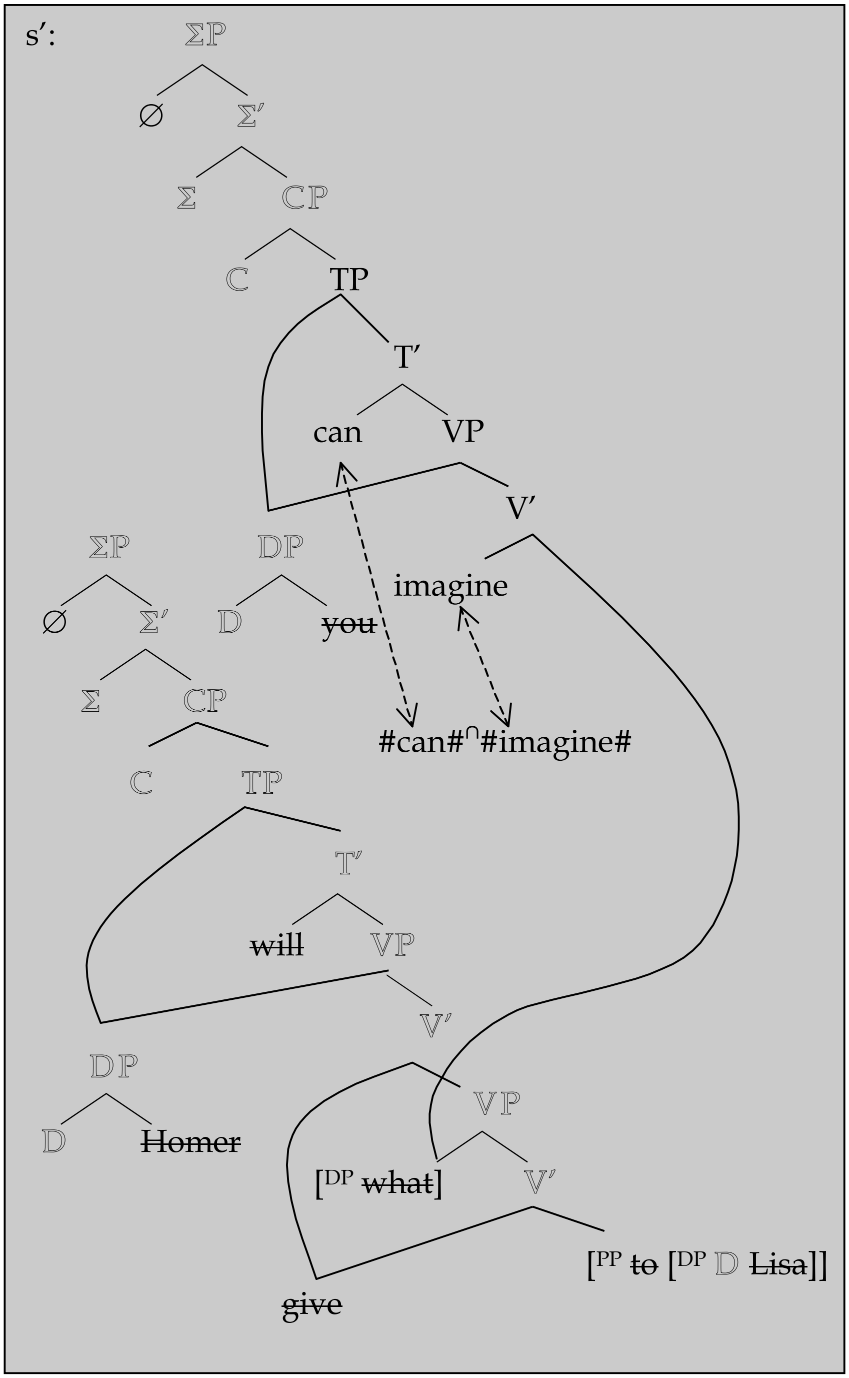

ΣP
∅
Σ′
Σ
CP
C
TP
T′
can
VP
V′
imagine
wh-
ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
~~give~~
DP
D
~~Homer~~
DP
D
~~you~~
#can#∩#imagine#∩#wh-#

(03) s:



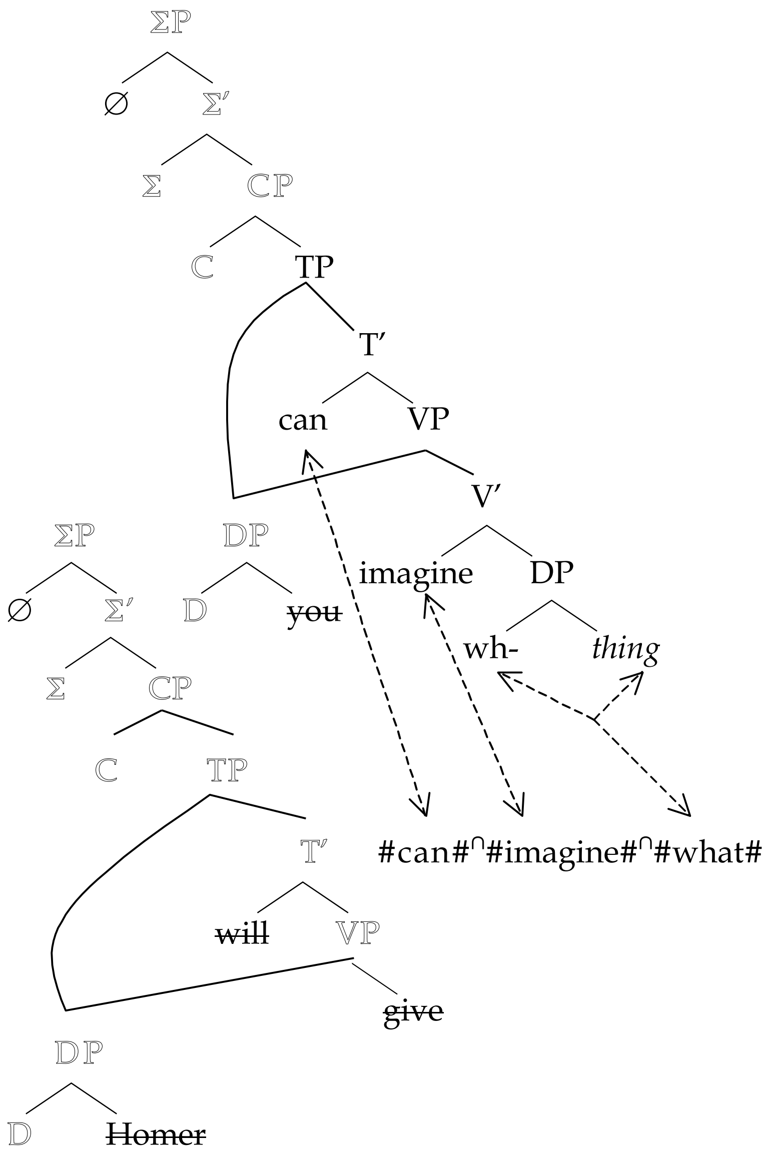
Notice that this WH-phrase could, in principle, have been built in the previous derivational round (cf. (03h’) above), and then remerged a (temporary) sister to imagine, as in shown (03s’).

(03)



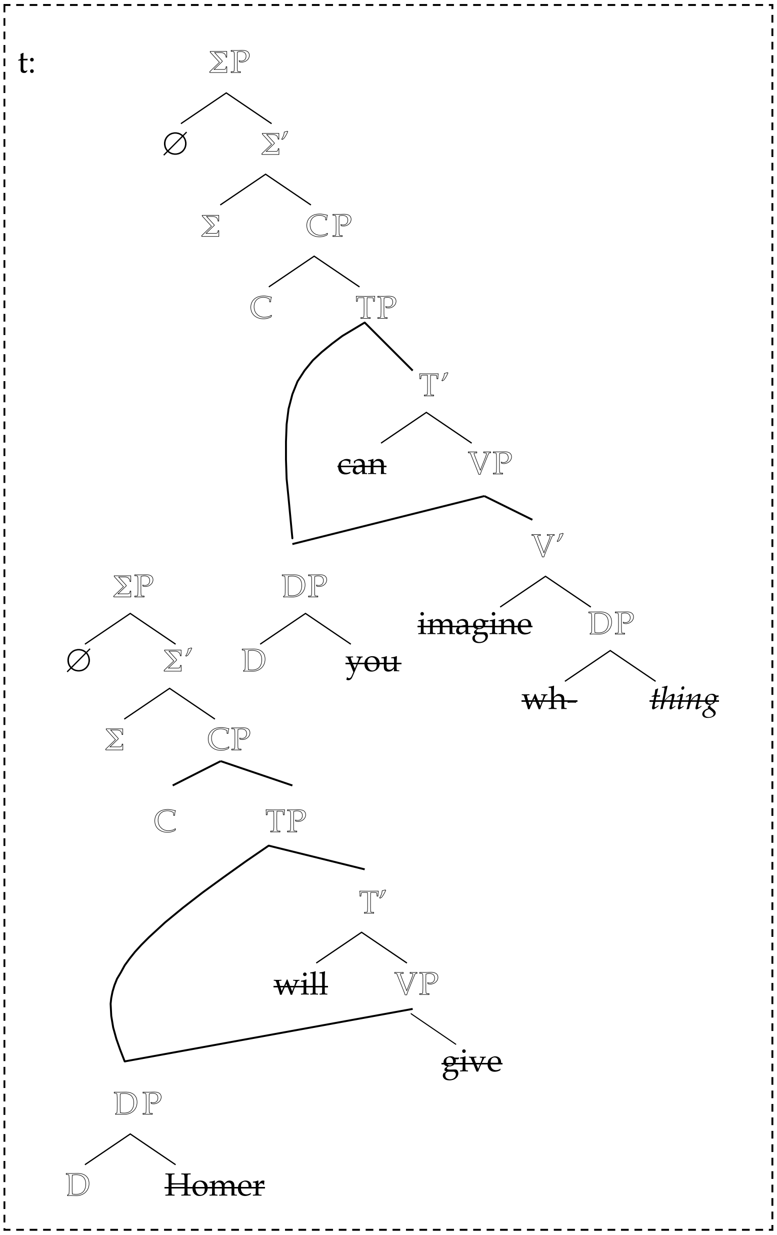
Notice, however, that such instance of (re)merge would have violated the c-command condition on merge. This is so because, right before the DP **<u>what</u>** gets shared, it is dominated by projections of **<u>will</u>** and **<u>give</u>**, which are lexical tokens that are present in numeration Ψ, therefore visible for calculating c-command relations in the derivational round that builds the subservient clause. Notice that none of those projections of **will** and **give** happen to dominate **imagine**. As a result, the DP **what** would fail to c-command **imagine**, which makes the step in (03s') illegitimate. This is the reason why the first derivational round was forced to terminate early, leaving an incomplete structure. That said, let us go back to step (03s), repeated below.

(03s)



At this point, the DP **what** is about to become a complex specifier, as soon as the new incoming [+WH] complementizer is selected from numeration Ψ and integrated to the phrase marker as a (temporary) sister to **what**. That would make it impossible for the pronounceable material inside that subject DP (i.e. #what#) to c-command any other terminal in the rest of the rest of the structure about to be formed, which would lead to an irreparable violation of the LCA in the subsequent steps. In order to avoid that, the system then needs to apply Spell-Out to the phrase marker under construction, removing the current PF-string from the derivational workspace, as shown in (03t).

(03)



← in the Syntax

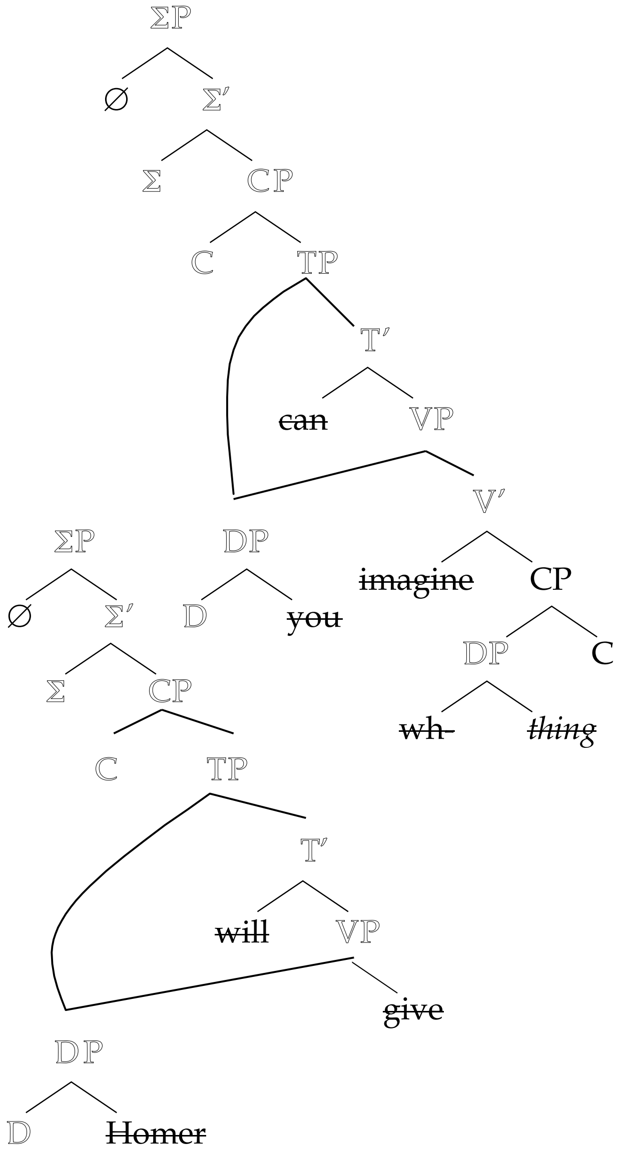
out to Phonology ↓

[#Homer#]$^\cap$[#will#$^\cap$#give#]$^\cap$[#you#]$^\cap$[#can#$^\cap$#imagine#$^\cap$#what#]

In the phonological component, the incoming string (i.e. **[#can#$^\cap$#imagine#$^\cap$#what#]**) is concatenated to the final edge of the existing super-string (i.e **[#Homer#]$^\cap$[#will#$^\cap$#give#]$^\cap$[#you#]**).

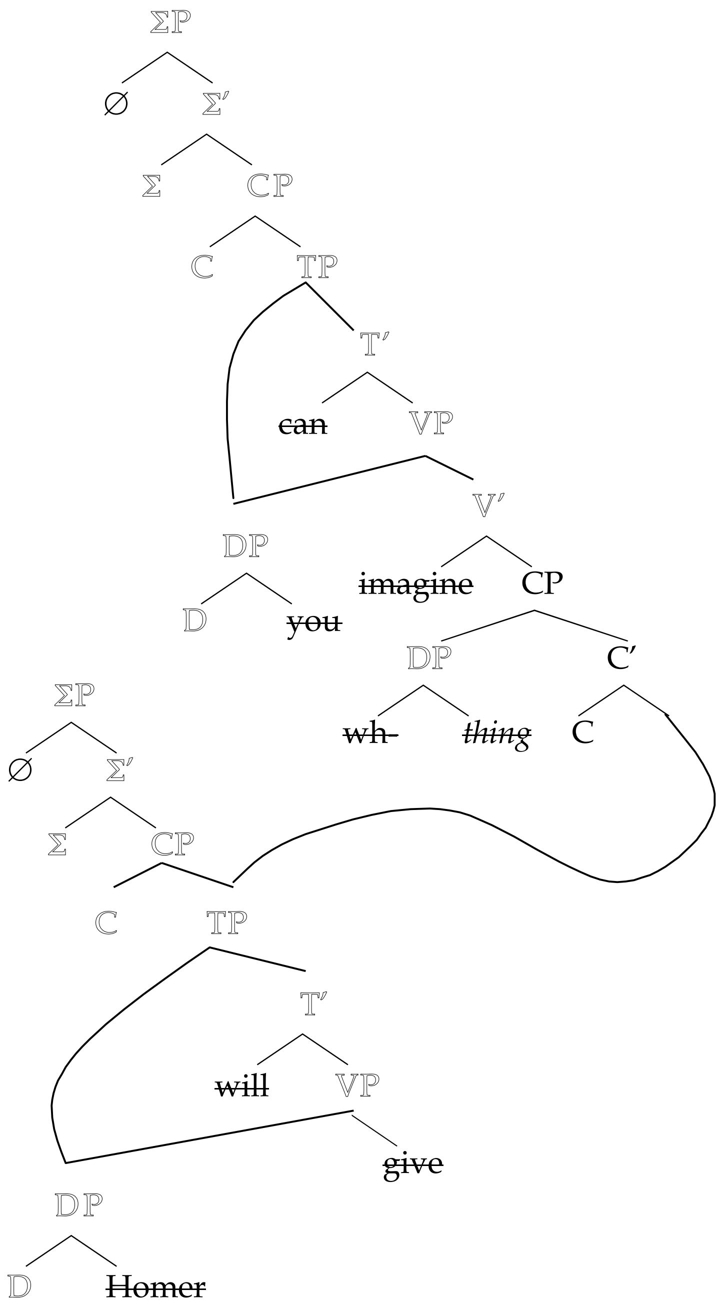
Then, the [+WH] complementizer in Ψ is finally introduced as a (temporary) sister to the DP **what**, as in (03u).

(03) u:



Now comes the crucial step. Given the intersection of numerations in the input, the system can take a TP that is already partially built as and remerge it as the complement of the lowest C in the subservient clause, as in (03v). Notice that this instance of remerge is possible because, in (03u), the TP vacuously c-commands what is about to become its new sister (i.e. $C_{[+WH]}$), as none of the nodes that dominate that TP is visible in the second derivational round (i.e. none of those dominating nodes is a projection of a lexical tokens in Ψ).

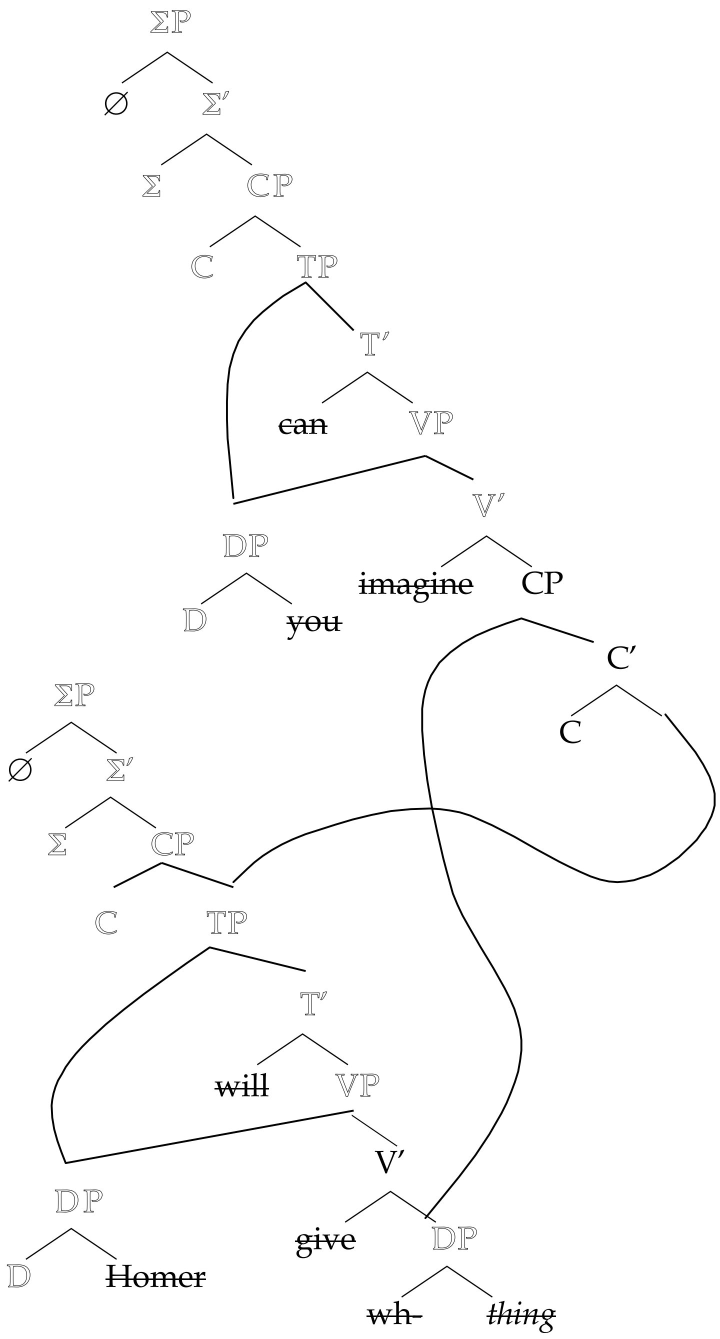
(03) v:



Next, the WH-phrase **<u>what</u>** is lowers into its theta position, being remerged inside the embedded VP, as a sister to **<u>give</u>**, as in (03w).[137]

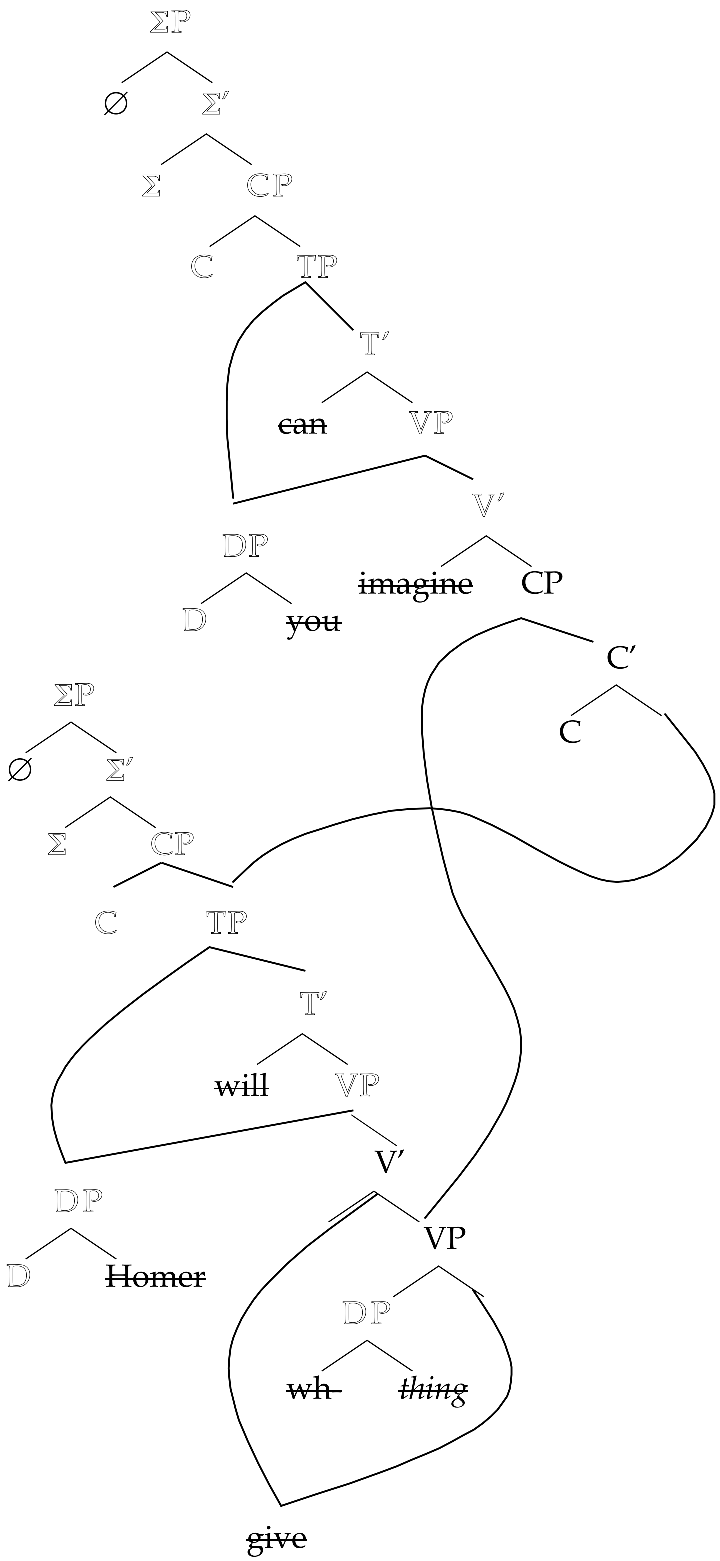
[137] I am abstract away from how what checks accusative case.

(03) w:

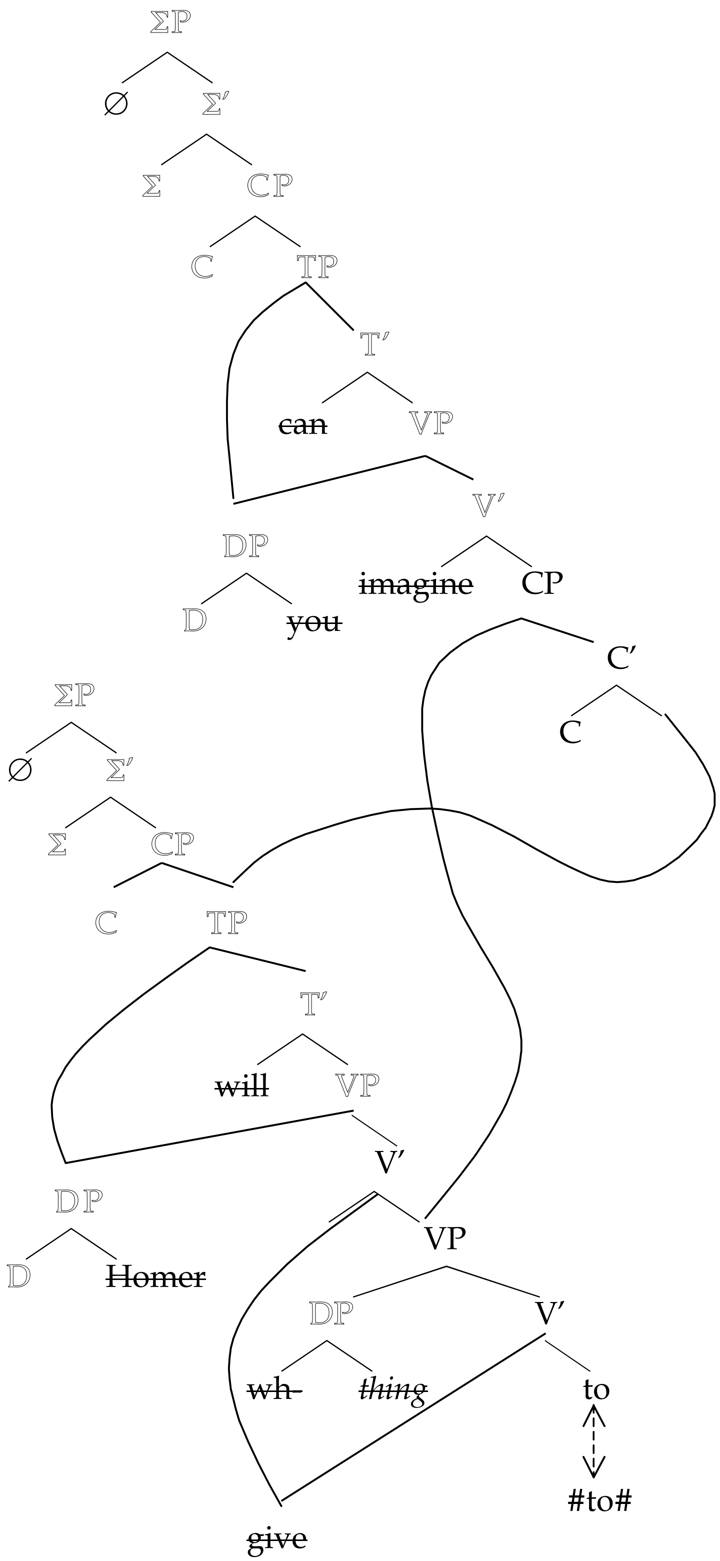


Then, the lower layer of the VP shell is built through the lowering of **give**, which remerges with **what**, as in (03x), so that the indirect object can be introduced as a sister to the verb in subsequent steps.

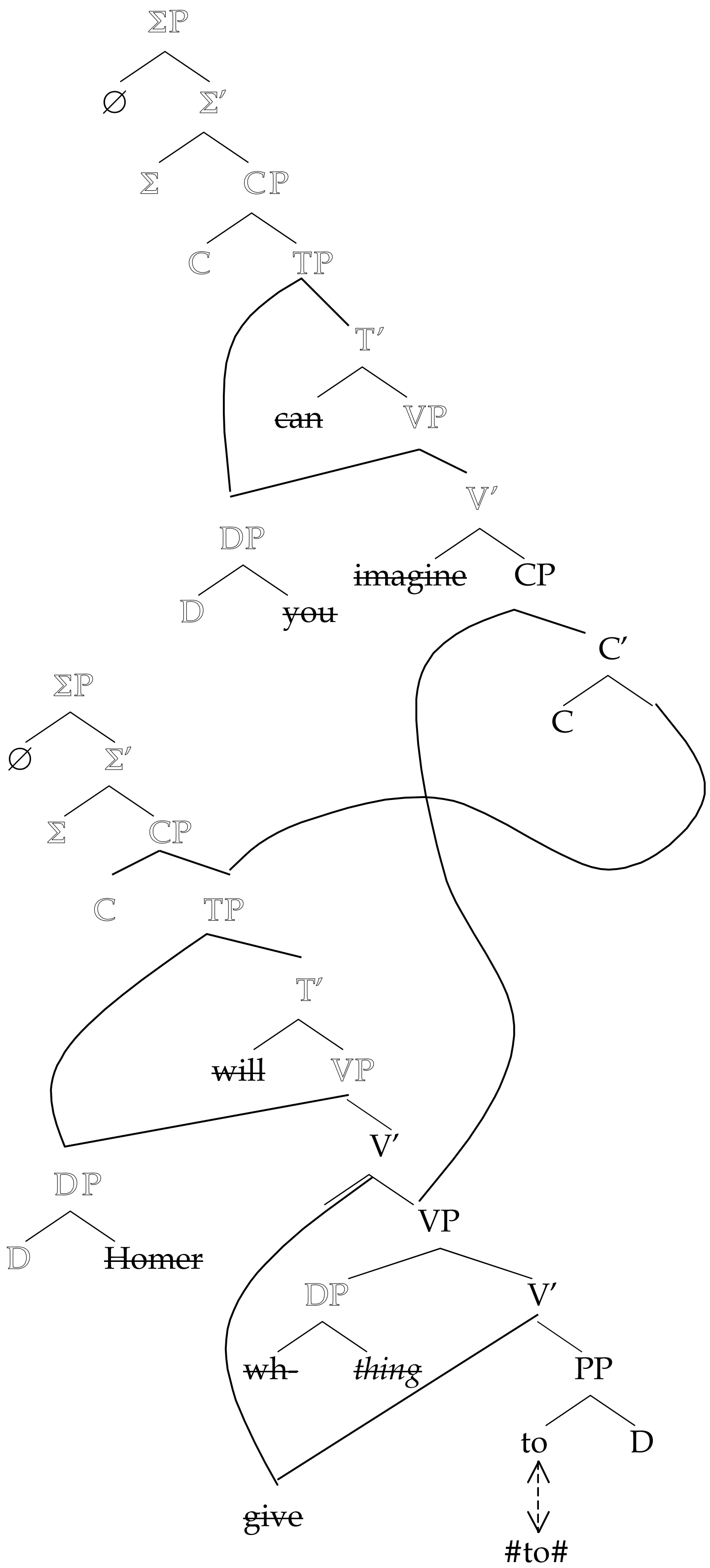
(03) x:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~can~~
VP
V′
DP
D
~~you~~
~~imagine~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~Homer~~
VP
DP
~~wh-~~
~~*thing*~~
~~give~~

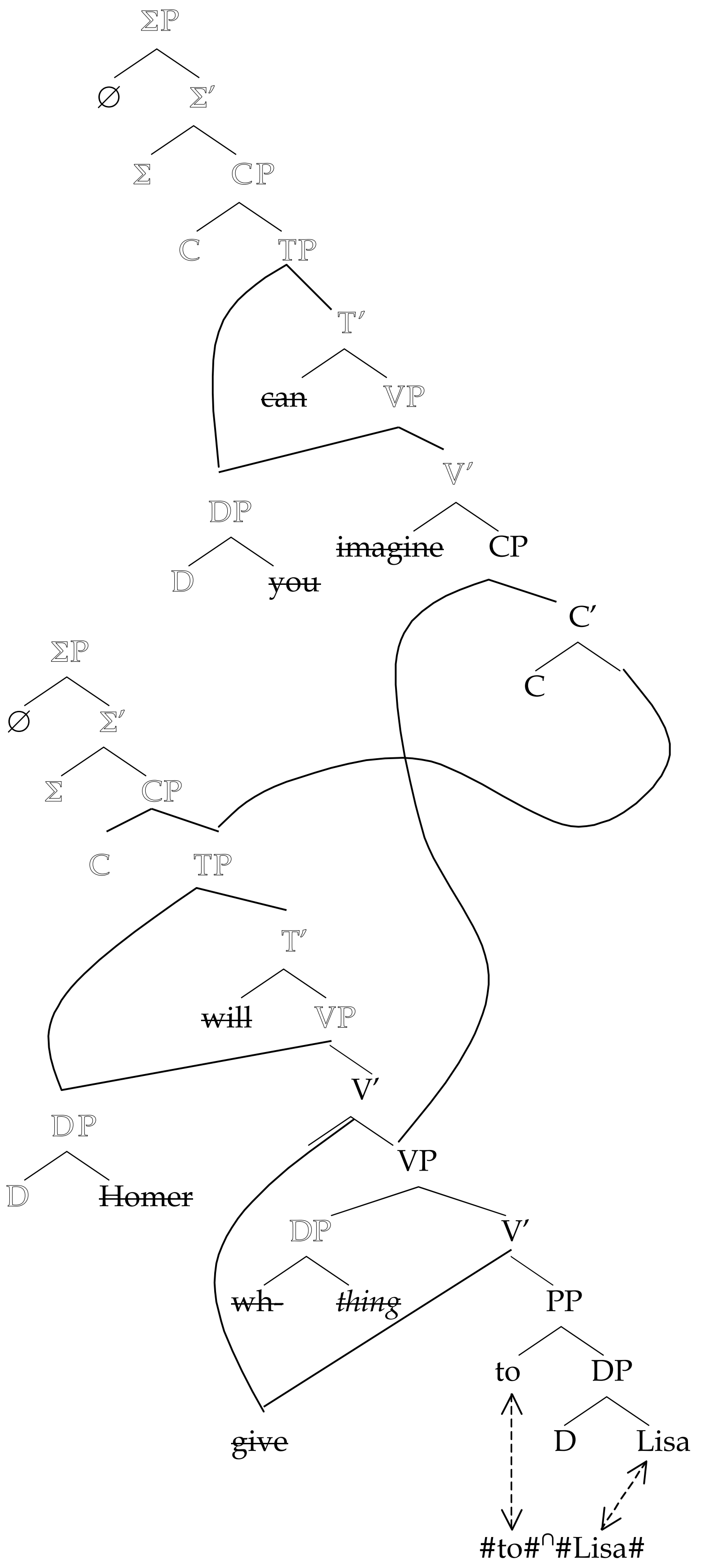
(03) y:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~can~~
VP
V′
DP
D
~~you~~
~~imagine~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~Homer~~
VP
DP
V′
~~wh-~~
~~*thing*~~
to
#to#
~~give~~

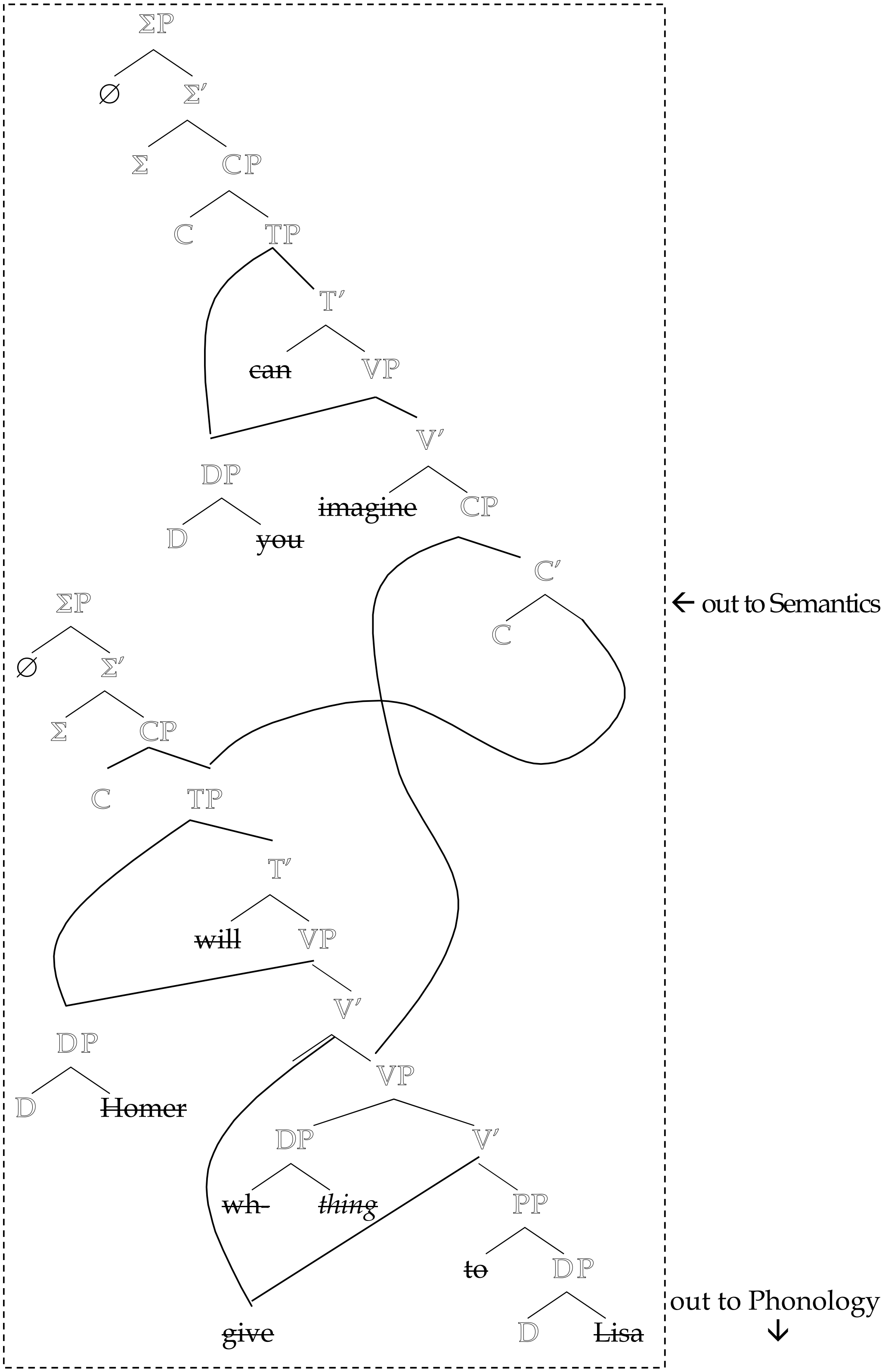
(03) z:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~can~~
VP
V′
DP
D
~~you~~
~~imagine~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~Homer~~
VP
DP
~~wh-~~
~~*thing*~~
V′
PP
to
D
~~give~~
#to#

(03) z′:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~can~~
VP
V′
DP
D
~~you~~
~~imagine~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~Homer~~
VP
DP
~~wh-~~
~~*thing*~~
V′
PP
to
DP
D
Lisa
~~give~~
#to#∩#Lisa#

(04)



[#Homer#]$^\cap$[#will#$^\cap$#give#]$^\cap$[#you#]$^\cap$[#can#$^\cap$#imagine#$^\cap$#what#]$^\cap$[#to#$^\cap$#Lisa#]

V.2. Multiple Matrix Clauses: parallelism and 'behindness'

Now, let us take a quick look at another case of amalgamation.

(05) I will find out if Homer gave you can imagine what to Lisa.

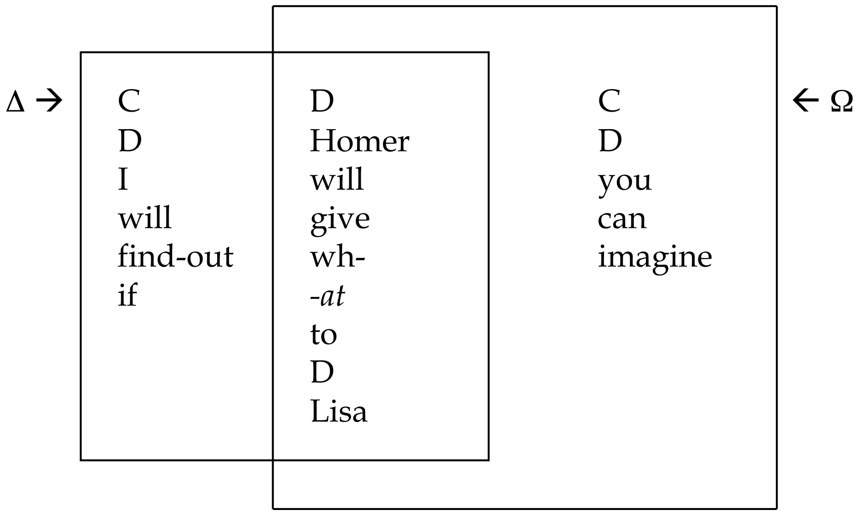
This example is very similar to the one analyzed above. The only difference is that its master clause is more complex. That is, in the previous example, the shared TP is a matrix TP in the master clause and an embedded TP in the subservient clause. The only portion of the master clause that is not shared is the CP.

In (05), on the other hand, the shared TP is an embedded TP in both the master clause and the subservient clauses, as we are going to see below.

The derivation of (05) would be pretty much like the derivation of (01), except that there is more material in the master clause.

The input would be the intersecting numerations in (06).

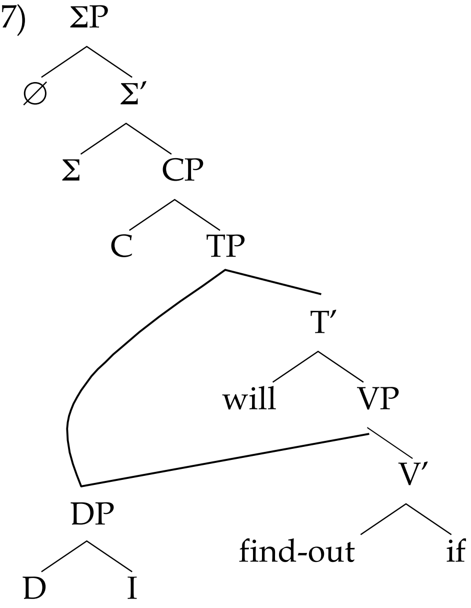
(06)



Abstracting away from the multiple application of spell-out along the derivation, the generation of (05) can be summarized in four stages.

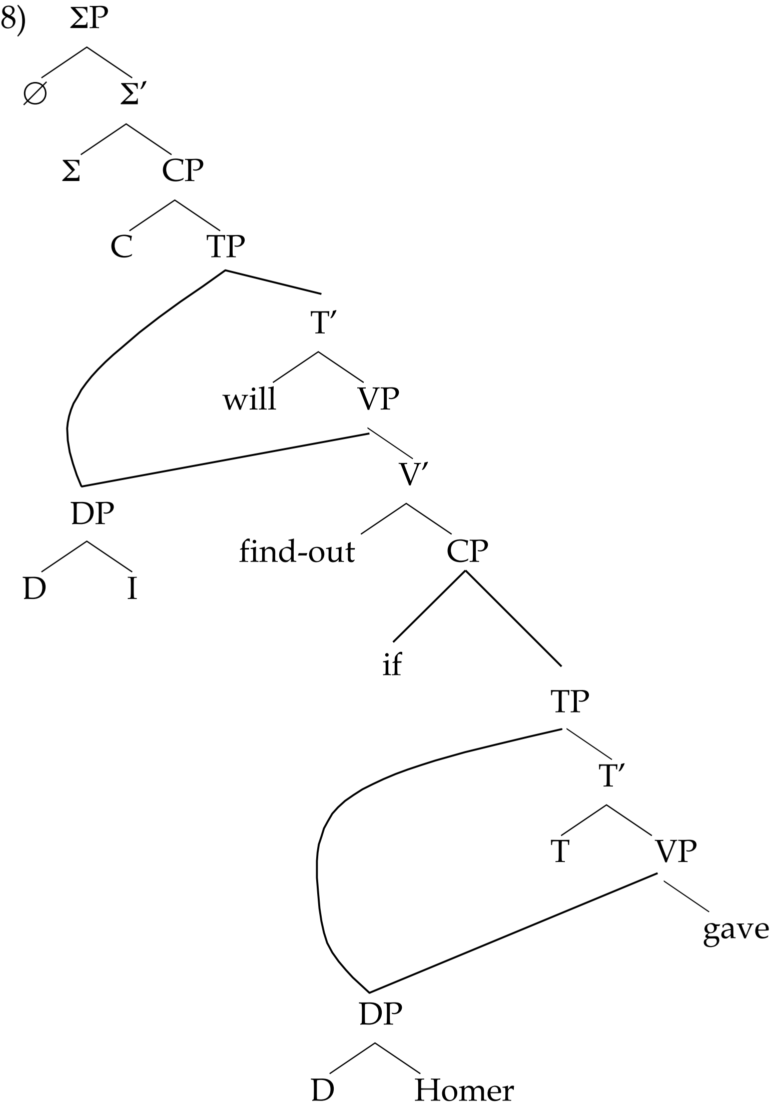
First, the computational system targets Δ, and starts to combine its lexical tokens, from the top downwards, up to the point where the higher portion of the master clause is built, as in (07).

(07)



Then that subcomputation proceeds, and the items in the intersection between Δ and Ψ start being integrated to the phrase marker, up to the point where (08) obtains.

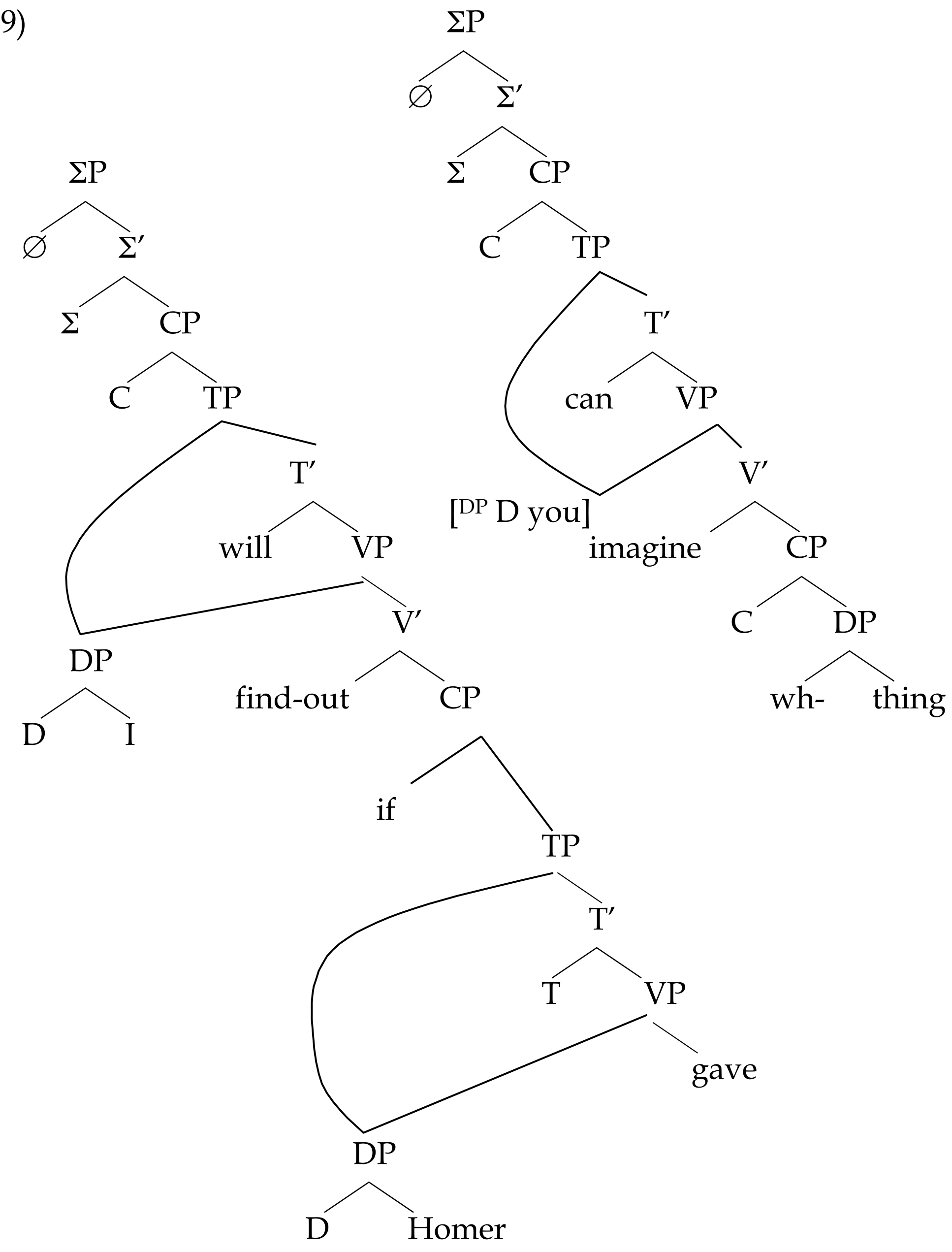
(08)



For the same reasons discussed above with regards to (03h′) and (03r′), the system is forced to terminate the first derivational round at this point, leaving the current phrase marker incomplete, to be finished by the end of the second derivational round.

The second derivational round begins. The lexical tokens in the non-shared portion of Ψ start to be combined, and the higher portion of the subservient clause is built, as in (09).

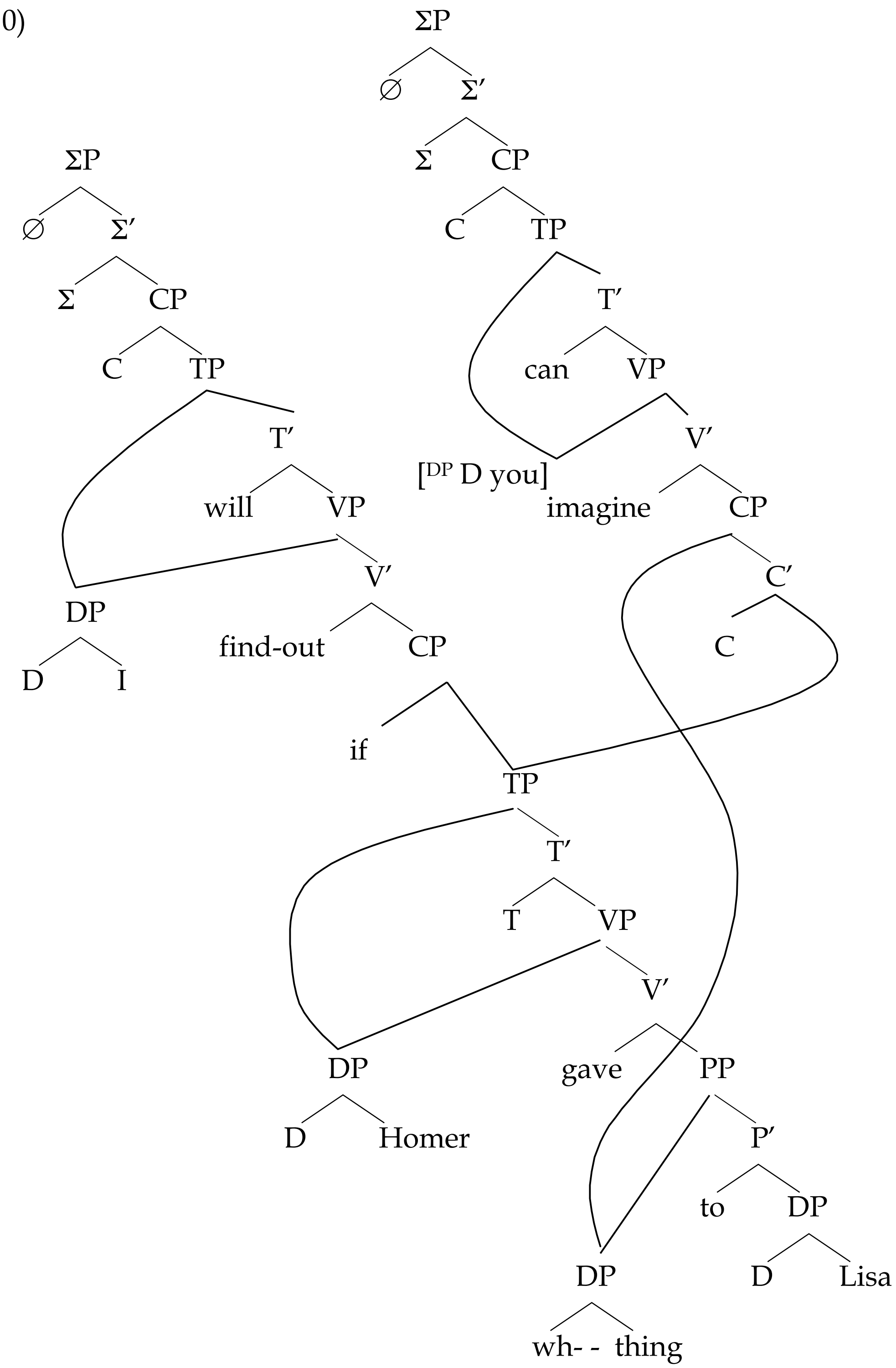
(09)



Eventually, the lower TP of the master clause is remerged as the sister to the lower C of the subservient clause. The WH-phrase what is remerged at its theta position inside the shared TP, and the indirect object **to Lisa** is eventually built, thus finishing the construction of that structure left incomplete in the previous derivational round.

The final representation for (05) is as in (10).

(10)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
will
VP
V′
DP
D
I
find-out
CP
ΣP
∅
Σ′
Σ
CP
C
TP
T′
can
VP
[DP D you]
V′
imagine
CP
C′
C
if
TP
T′
T
VP
V′
DP
D
Homer
gave
PP
P′
to
DP
D
Lisa
DP
wh- -
thing

Examples of this kind, with more complexity in the master clause, are more transparent in terms of exhibiting the properties of 'parallel messages' and 'multiple layers of information', as discussed in **§II.4**.

Notice that, in (10), the substructure corresponding to the giving event is under the scope of both (i) the substructure corresponding to the imagining event, and (ii) the substructure corresponding to the figuring-out event. Thus, both the master clause is about finding out something about a certain giving event, whereas the subservient clause is about the ability to imagine something about that very same giving event. Crucially, there is no scope relation between the substructure corresponding to the figuring-out event and the substructure corresponding to the imagining event. Syntactically, these two substructures are fully parallel, but one (i.e. the higher portion of the subservient clause) is ***behind*** the other (i.e. the higher portion of the master clause). In the framework developed here, this 'behindness' is a function of which portion of the Siamese-Tree gets to be built first, which has a direct impact on informational structure and on the PF-string.

In a nutshell, strictly speaking, a syntactic amalgam is not a sentence. It is a organization of two (or more) sentences that share some subpart. The representation in (10) above can be factored out into (11) and (12) below.

(11) [^{CP} C [^{TP} I$_2$ will [^{VP} t$_2$ find-out [^{CP} if [^{TP} Homer$_1$ T [^{VP} t$_1$ gave what [^{PP} to Lisa]]]]]]]

(12) [^{CP} C [^{TP} you$_3$ can [^{VP} t$_3$ imagine [^{CP} what$_3$ [^{TP} Homer$_1$ T [^{VP} t$_1$ gave t$_3$ [^{PP} to Lisa]]]]]]]

While (12) is an independently available well-formed structure when in isolation, the structure in (11) is not, as it violates whatever principle demands that WH-movement be overt (in English). Somehow, the instance of WH in situ in (11) gets licensed by virtue of (11) being amalgamated with (12).[138]

Descriptively speaking, (11) and (12) somehow collapse at the paratactic level, yielding (10), whose only WH-phrase is in a chain configuration. Interestingly, this paratactic effect obtains from the application of syntactic operations alone. As a result, the WH-phrase in the master clause (cf. (11)) behaves as an indefinite for all intents and purposes, which makes the structure in (10) equivalent to the paratactic construction in (13).[139]

(13) I will find out if Homer gave a certain thing to Lisa, and you can imagine what is that certain thing that Homer gave to Lisa.

Another welcome consequence of analyzing syntactic amalgams in terms of Siamese-Trees representations such as (04) and (10) is that the matrix-clause

[138] Actually, there is nothing new about the idea of licensing a structure that would otherwise be ungrammatical. Examples (i-a) and (ii-a) are ungrammatical in isolation, but are legitimate parts of larger syntactic structures, as in (i-b) and (ii-b). The question, then, is why and how each particular otherwise ungrammatical structure gets licensed when combined with extra structure of a certain kind.

(i) a: * [^{IP} [^{DP} John]$_1$ to be [^{AP} t$_1$ happy]]
b: [^{IP} [^{DP} Mary]$_1$ believes [^{IP} [^{DP} John]$_1$ to be [^{AP} t$_1$ happy]]]
(ii) a: * [^{CP} [^{DP} who] [^{IP} John will invite t$_1$ to his party]]?
b: [^{IP} I don't know [^{CP} [^{DP} who] [^{IP} John will invite t$_1$ to his party]]]

[139] The intuition behind this analysis is that the WH-feature of **what** is licensed/checked against a [+WH] complementizer in the subservient clause, leaving only the 'pronominal' part of it to be seen by the [-WH] complementizer of the master clause. For a detailed and lengthy discussion on the formal details of how WH-phrases may semantically behave as indefinites in syntactic amalgams, see Guimarães (2003c).

behavior exhibited by all invasive substrings is straightforwardly accounted for. As discussed in **§II.10**, those quasi-parenthetic chunks may exhibit syntactic patterns found only in matrix clauses, like auxiliary-inversion for questions, or imperative mood, as shown in (14) and (15).

(14) Bob told me that Amy danced with [*do you know who?*] at the party

(15) Bob told me that Amy danced with [*guess who!*] at the party

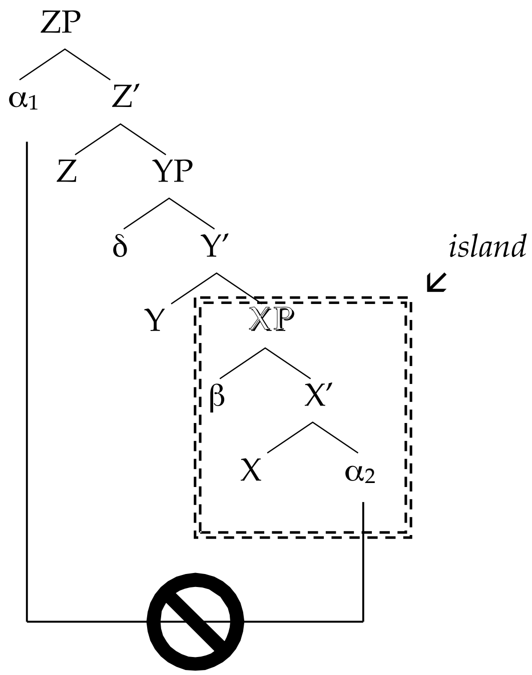
Under the approach developed here, this result is not surprising, as any of those 'invasive chunks' is indeed another matrix clause built in parallel, which just happens to be 'behind' the one chosen as the ' main message'.

V.3. Multiple Roots and Relativized Islandhood

It is outside the scope of this dissertation to investigate (i) what makes a given syntactic constituent an island for extraction, or (ii) why islandhood only prevents movement transformations and not other long-distance dependencies (i.e. binding, agreement), or (iii) whether there is a unified account for all types of island.

My starting point is the well-known generalization that certain kinds of constituent, for some reason, are island for extraction. Whatever the ultimate explanation for that turns out to be, the generalization itself can be equally described in at least three different meta-languages, as summarized in (16).

(16)



From a representational point of view, we can say that, if a given constituent XP is an island, there must not be a chain formed with only link inside XP and another link outside XP.

From a derivational perspective, assuming a bottom-up directionality, we can say that, if a given constituent XP is an island, no element α can move from a position inside XP to another position outside XP.

From a derivational perspective, assuming a top-to-bottom directionality (as proposed in his dissertation), we can say that, if a given constituent XP is an island, no element α can move from a position outside XP to another position inside XP.

As shown in **§II.5**, amalgamation is insensitive to islands. That is, the invasive clause can interrupt the invaded clause at a position of the substring that is exhaustively included inside a constituent of the kind that defines an island for extraction, as exemplified in (17) and (18) below.

(17) *Complex NP/DP Island*

a: Susan dismissed the claim that her husband dated *I can't remember who* before they got married.

b: Susan dismissed the claim that her husband dated Sarah before they got married.

c: * I can't remember **[who]$_1$** Susan dismissed **{** the claim that her husband dated **t$_1$** before they got married **}**.

(18) *Adjunct Clause Island*

a: John invited all his friends to a big party after he got *you can imagine which job*.

b: John invited all his friends to a big party after he got *the job of head coach of the Chicago Bulls* .

c: * You can imagine **[which job]**$_1$ [John invited all his friends to a big party **{** after he got $\mathbf{t}_1$ **}**].

As argued in **§III.3.5.2**, this is a serious problem for any neo-conservative approach to amalgamation based on remnant-movement, as that would require that WH-extraction out of an island at some point in the derivation, as sketched in (19) and (20).

(19) a: [^{TP} Susan dismissed [^{DP} the claim [^{CP} that [^{TP} her husband dated who [^{PP} before they got married]]]]

b: ☹ [^{TP} I can't remember [^{CP} **who**$_1$ [^{TP} Susan dismissed [^{DP} the claim [^{CP} that [^{TP} her husband dated $\mathbf{t}_1$ [^{PP} before they got married]]]]]]]

c: [^{CP} [^{TP} I can't remember [^{CP} **who**$_1$ [^{TP} Susan dismissed [^{DP} the claim [^{CP} that [^{TP} her husband dated $\mathbf{t}_1$ $\mathbf{t}_2$]]]]]] [^{PP} before they got married]$_2$]

d: [^{CP} [^{TP} Susan dismissed [^{DP} the claim [^{CP} that [^{TP} her husband dated $\mathbf{t}_1$ $\mathbf{t}_2$]]]]$_3$ [^{CP} [^{TP} I can't remember [^{CP} **who**$_1$ $\mathbf{t}_3$]] [^{PP} before they got married]$_2$]]

(20) a: [^{TP} John invited all his friends to a big party [^{PP} after [^{TP} he got [which job]]]]

b: ☹ [^{TP} you can imagine [^{CP} **[which job]**$_1$ [^{TP} John invited all his friends to a big party [^{PP} after [^{TP} he got $\mathbf{t}_1$]]]]]

c: [^{CP} [^{TP} John invited all his friends to a big party [^{PP} after [^{TP} he got **t$_1$**]]] [^{TP} you can imagine [^{CP} **[which job]$_1$ t$_2$**]]]

Under the system proposed in this dissertation, however, the facts follow straightforwardly. Once we analyze syntactic amalgams in terms of Siamese-Tree configurations, where one embedded TP simultaneously belongs inside more than one matrix clause (with these matrix clauses standing 'behind' each other), the facts follow straightforwardly. Speaking in 'bottom up' terms for the sake of exposition, what happens with the structures under discussion is that the shared TP, out of which a WH is extracted,[140] can be inside an island relatively to one embedding domain, but outside an island relatively to the domain to which the WH moves.

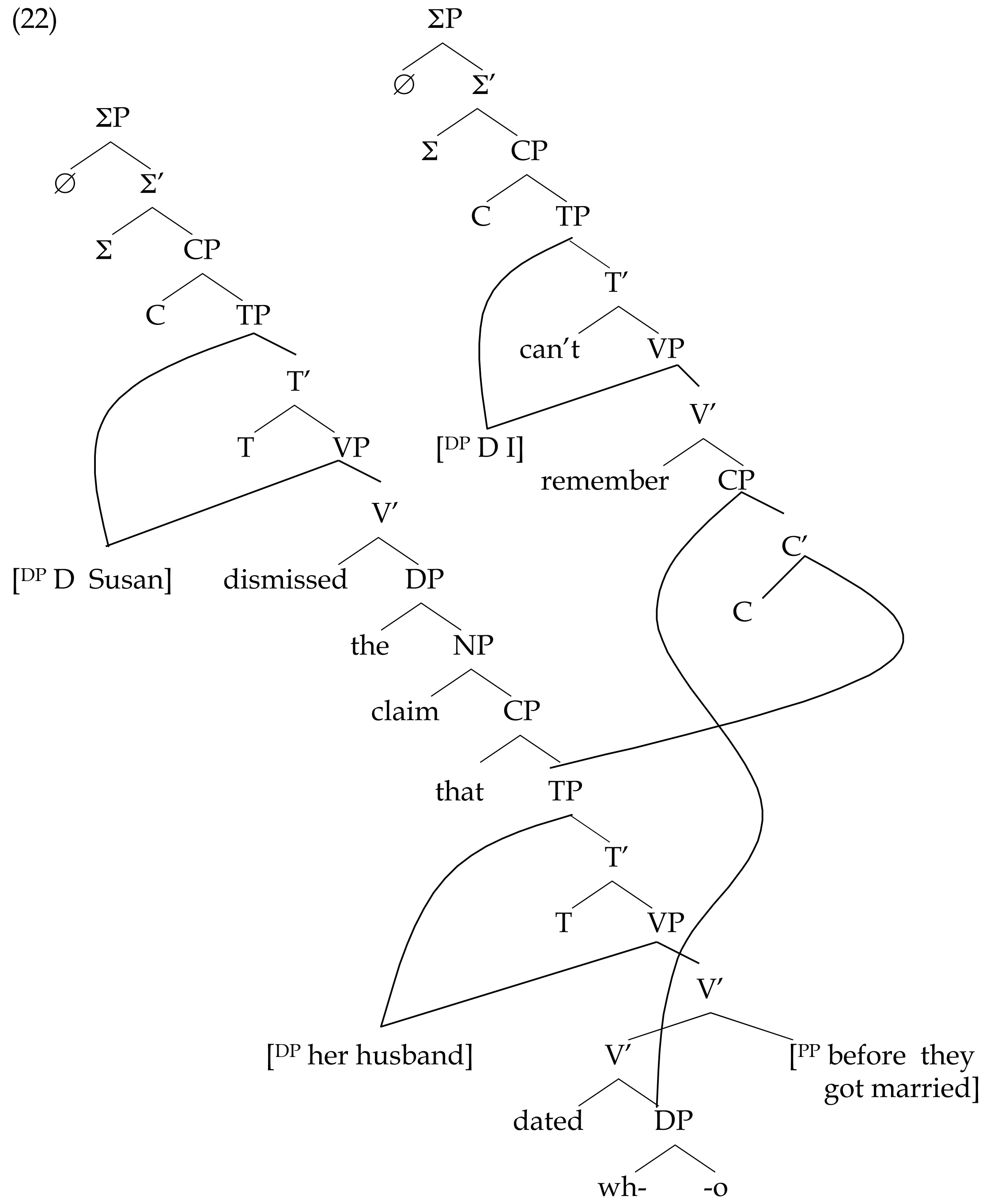
For this specific phenomenon, the top-to-bottom derivational dynamics adopted in this dissertation is not crucial. What makes those constructions possible is the multi-rootedness of the representation, so that the island is invisible to the subcomputation where the chain is formed.

For instance, the example in (17a) —repeated below as (21) — would be structured as in (22).

[140] In the case of cleft amalgams, what is extracted is a non-WH DP. But both movements share the property of having the highest chain-link in the specifier of a CP (or whatever specific functional category in that highest structural layer of the clause) which, in turn, is embedded inside a larger (non-shared) clause.

(21) Susan dismissed the claim that her husband dated *I can't remember who* before they got married

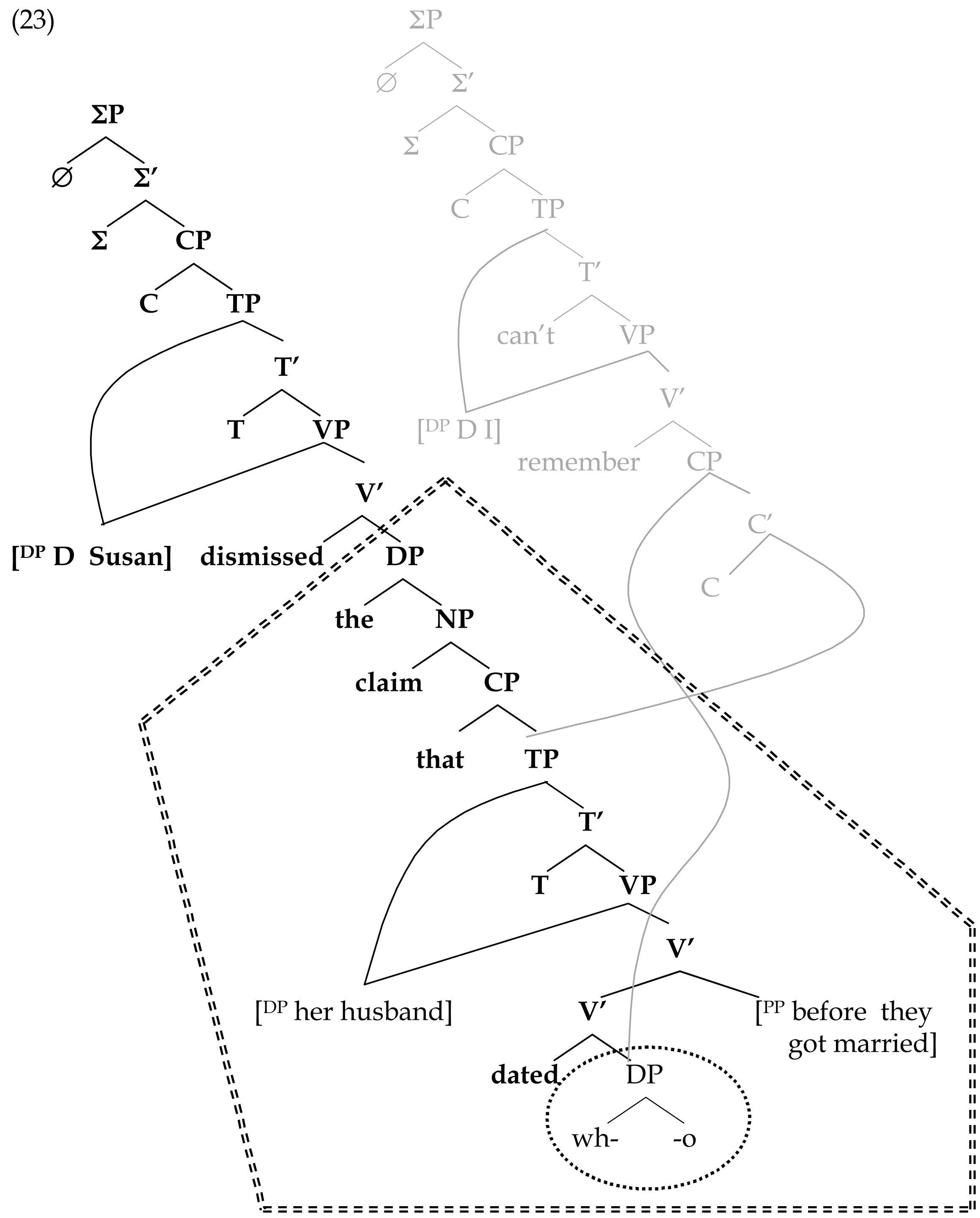
(22)



This is consistent with the meaning of (21). What the speaker cannot remember is not the identity of a woman ***x*** such that Susan dismissed the claim that her husband dated ***x*** before they got married. Rather, what the speaker cannot remember is the identity of the woman ***y*** such that Susan's husband dated ***y*** before they got married.

Notice that lower **<u>who</u>** is indeed inside an island relatively to the master clause, as explicitly indicated in (23).

(23)



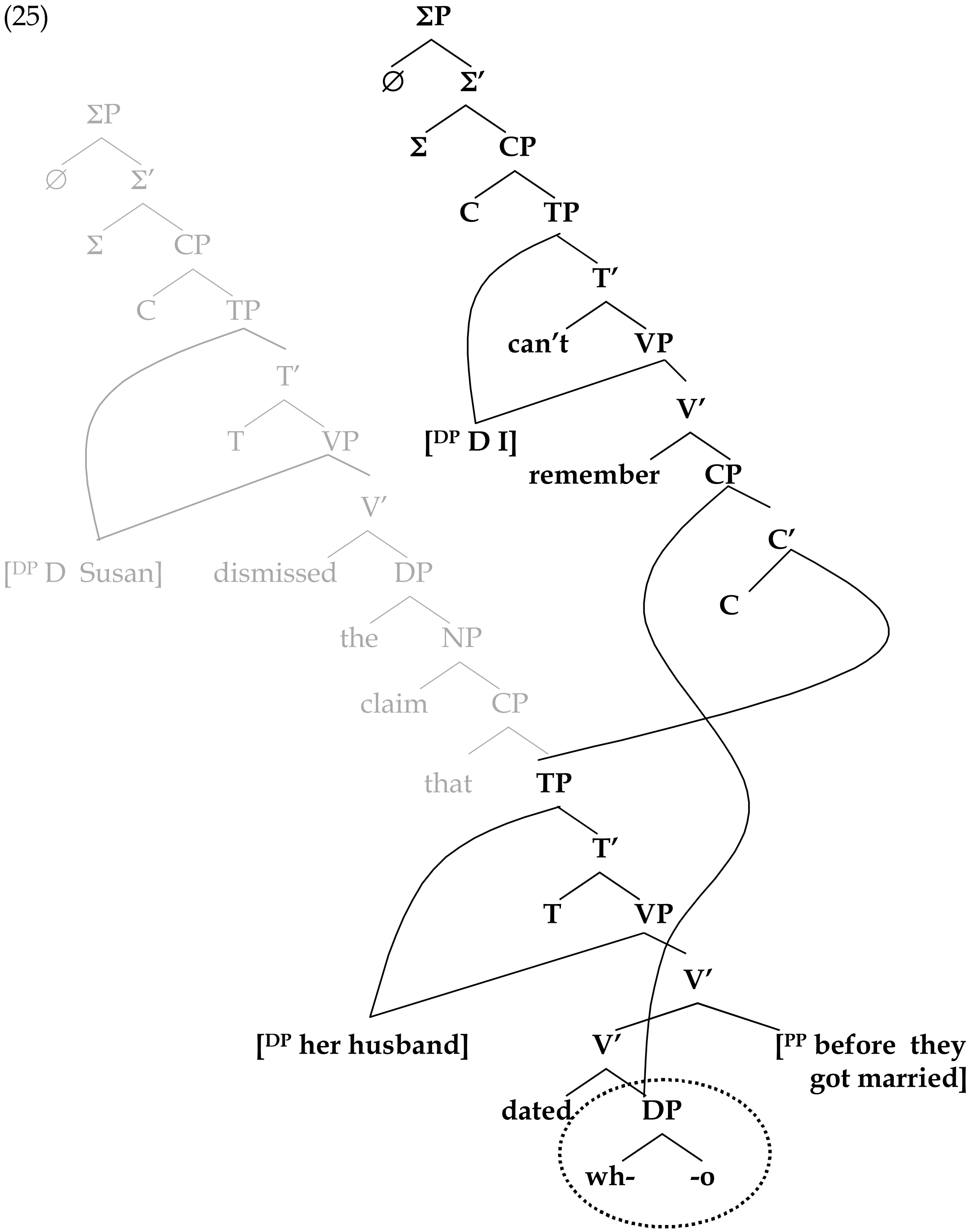
So, forming a chain with on e occurrence of **<u>who</u>** in that lower position and another one in the highest spec/CP of the master clause would constitute a violation of whatever principle makes the domain highlighted above an island, as shown in (24)

(24) * **Who$_1$** did Susan dismiss {the claim that her husband dated **t$_1$** before they got married}?

However, considering the whole structure, **<u>who</u>** is actually not a link of a chain with relatively to that domain. It is a chain-link only relatively to the subservient clause, which is built in a subcomputation that cannot even see that part of the structure where the island is, as shown in (25) and (26).[141]

[141] Given the system here proposed, such invisibility would be a consequence of the relevant lexical tokens not being in the intersection of the two reference sets.

(25)



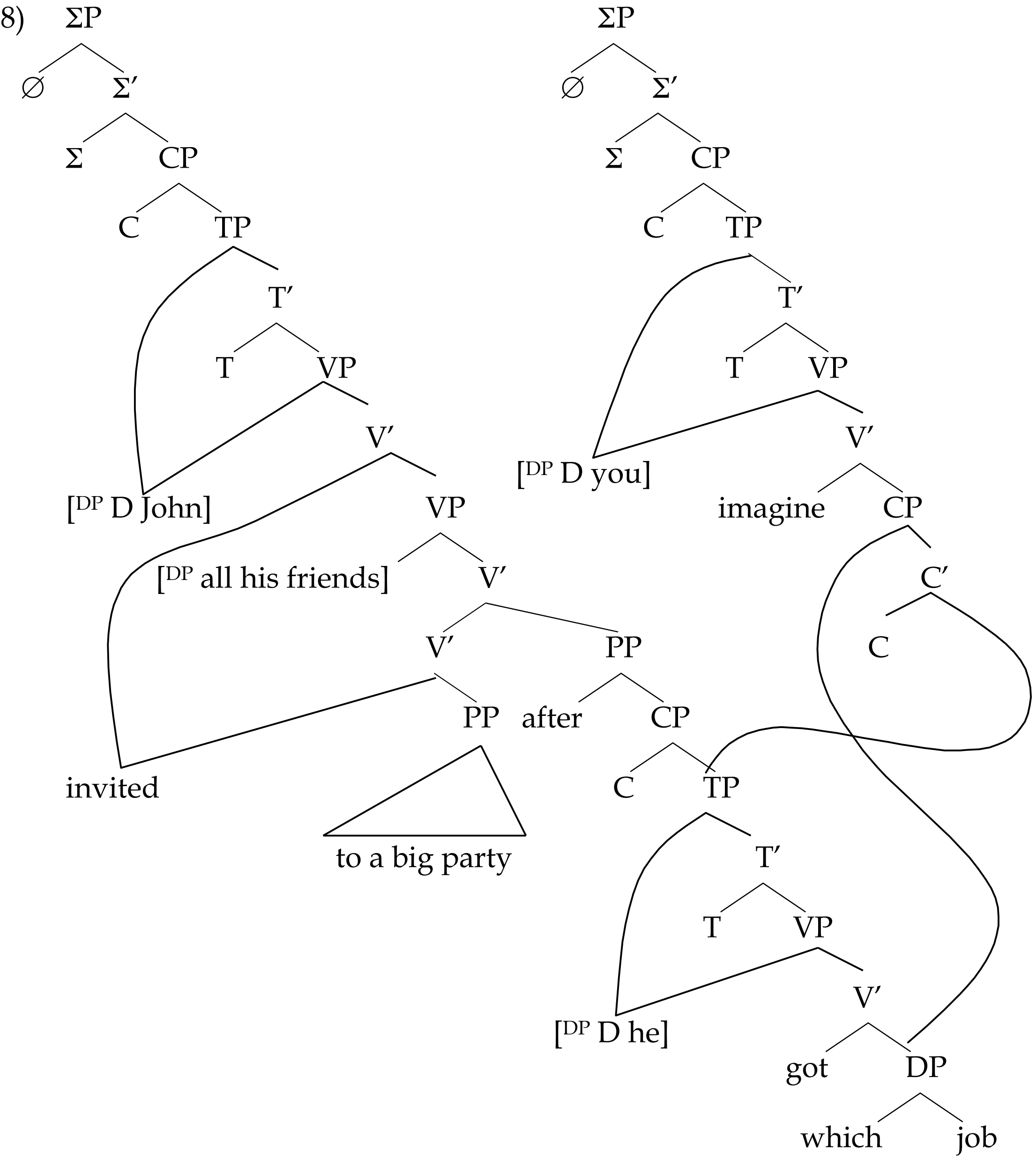
(26) I can′t remember **who₁** her husband dated **t₁** before they got married.

The same thing is true of (18a), repeated below as (27).

(27) John invited all his friends to a big party after he got *you can imagine which job*.

The corresponding representation would be as in (28).

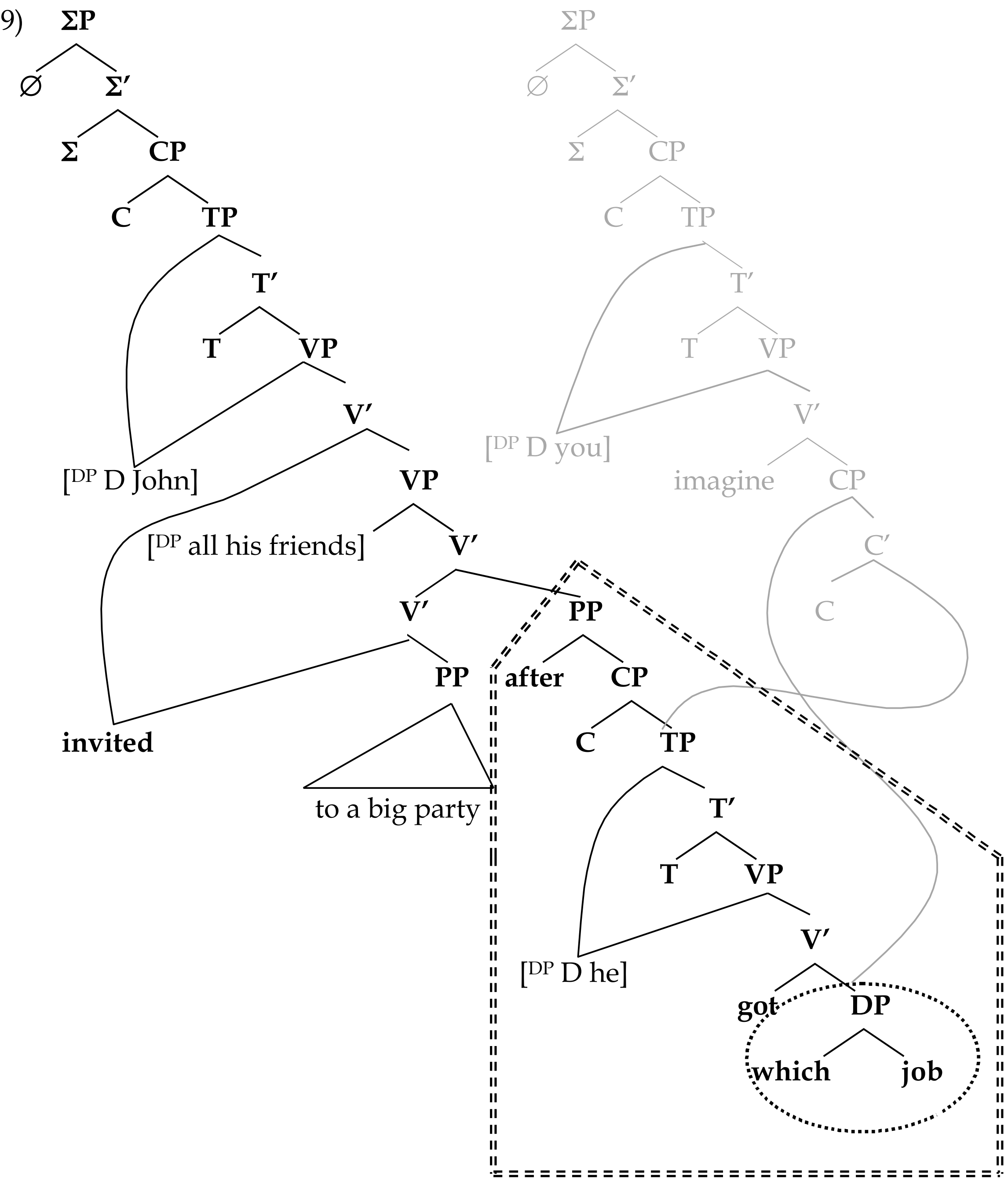
(28)



Again, this is consistent with the meaning that (27) has. What the listener can imagine is not the nature of ***x*** such that John invited all his friends to a big party after he got a job of the type ***x***. Rather, what the listener can imagine is simply the nature of ***x*** such that John got a job of the type ***x***.

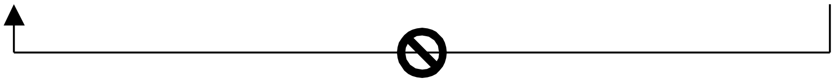
Notice that lower **which job** is indeed inside an island relatively to the master clause, as explicitly indicated in (29).

(29)



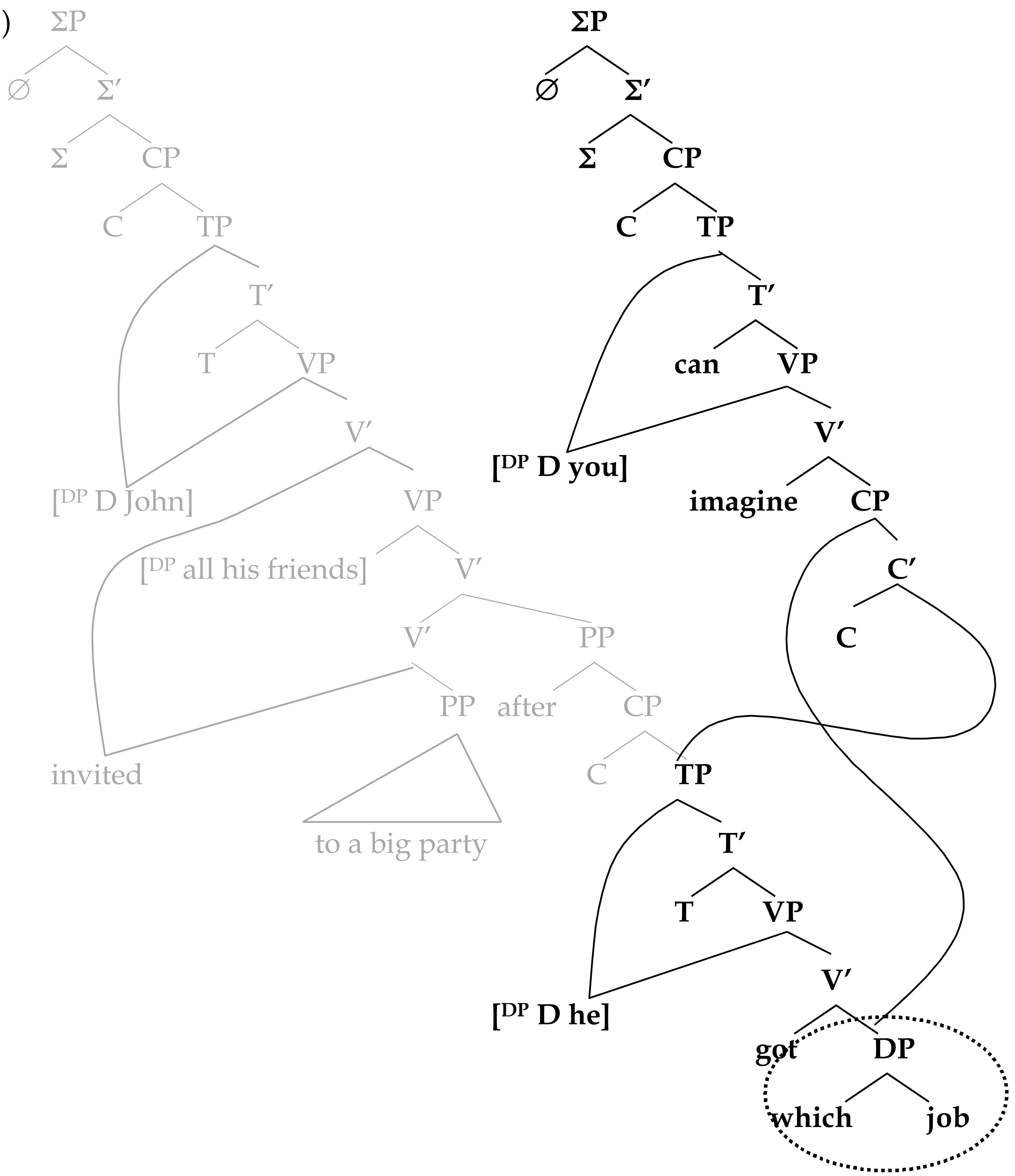
So, forming a chain with on e occurrence of **which job** in that lower position and another one in the highest spec/CP of the master clause would constitute a violation of whatever principle makes the domain highlighted above an island, as shown in (30)

(30) * **[Which job]**$_1$ did John invite all his friends to a party **{** after he got **t**$_1$ **}**



Once the whole structure is considered, it is easy to see that **<u>which job</u>** is actually not a link of a chain relatively to that domain, but only relatively to the subservient clause, which is built in a subcomputation to which the island is invisible, as shown in (31) and (32).

(31)



(32) You can imagine **[which job]₁** he got **t₁**.

V.4. Cross-Linguistic Word Order Variation

As discussed in **§II.8**, there is an interesting cross-linguistic variation to be explained, which concerns those instances of syntactic amalgamation where the object of a preposition is the target 'clause invasion', as exemplified in (33) and (34).

(33) *English*

a: Bob gave money to I forgot who.

b: * Bob gave money I forgot to who

(34) *Romance (Portuguese)*

a: * Bob deu dinheiro pra eu me esqueci quem.
Bob gave money to I REFL-forgot who.

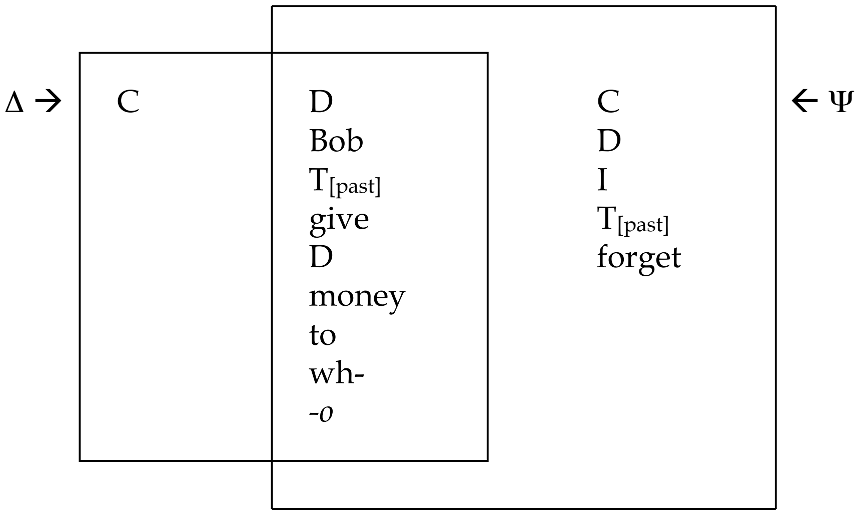
b: Bob deu dinheiro eu me esqueci pra quem.
Bob gave money I REFL-forgot to who

In **§II.8**, I have shown that this pattern poses a real problem for sluicing-based approaches to amalgamation, requiring extra *ad hoc* assumptions to account for the contrast.

Now, I will show how the facts can follow straightforwardly from the system here proposed.

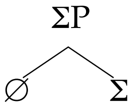
Let us consider the English case in (33) first. The starting point would be the intersecting numerations in (35).

(35)



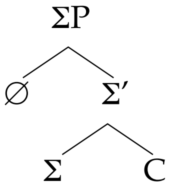
The system begins by randomly zooming into numeration Δ. The lexical tokens in that set start being combined in the usual top-to-bottom fashion, as in (36a) through (36d).

(36) a:

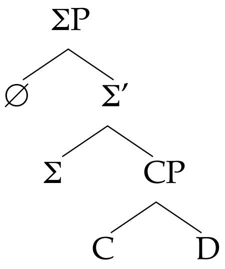


(starting axiom)

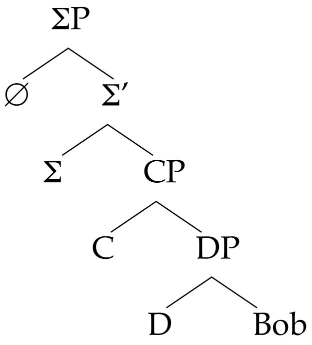
b:



c:

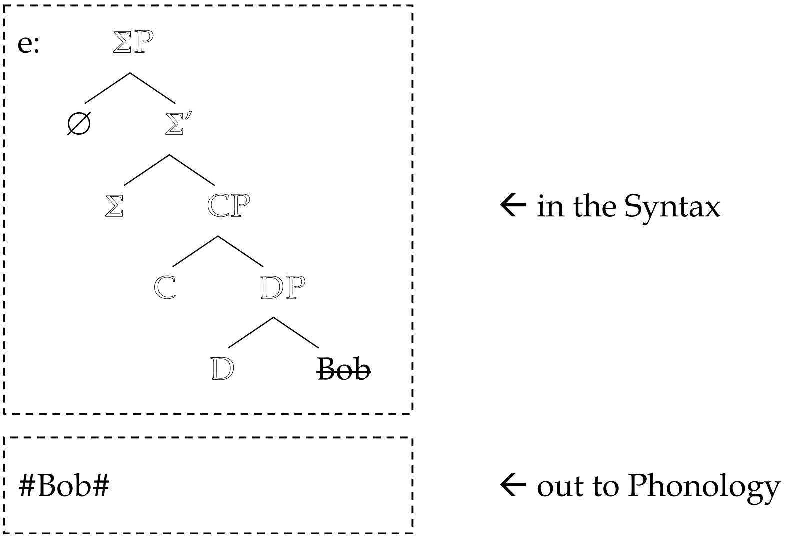


d:



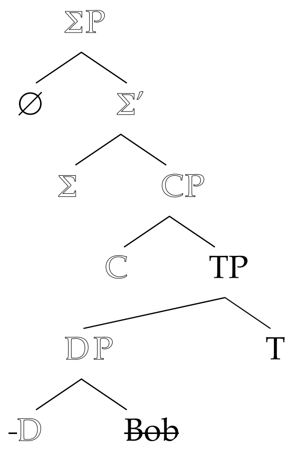
At this point, spell-out needs to apply to ensure LCA-satisfaction in the subsequent steps, as shown in (36e).

(36)

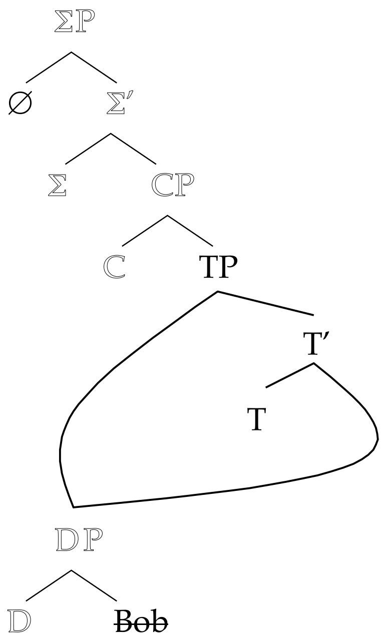


The derivation proceeds with the construction of the master clause, all the way down to the direct object, as in (36f) through (36j).

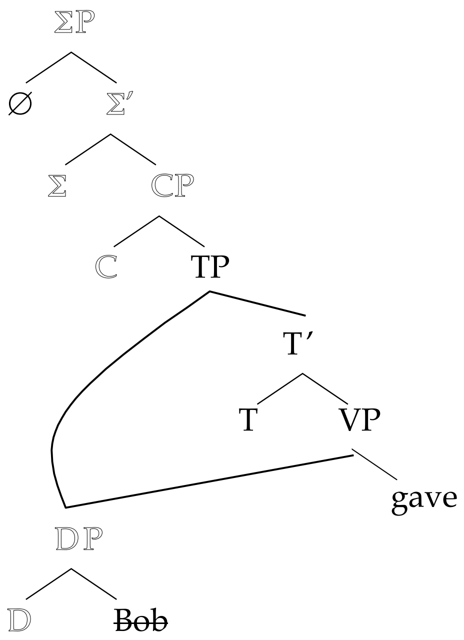
(36) f:



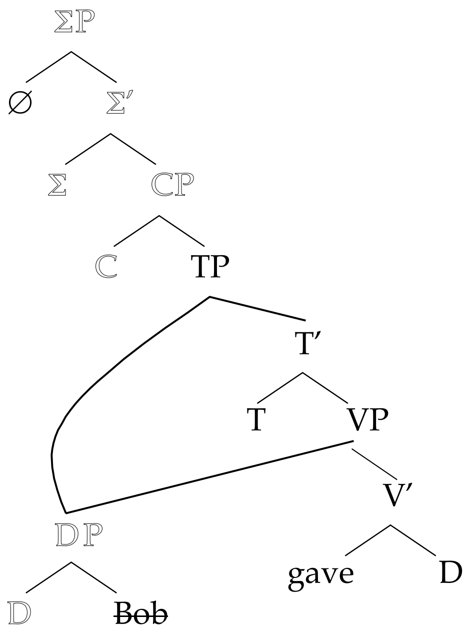
g:



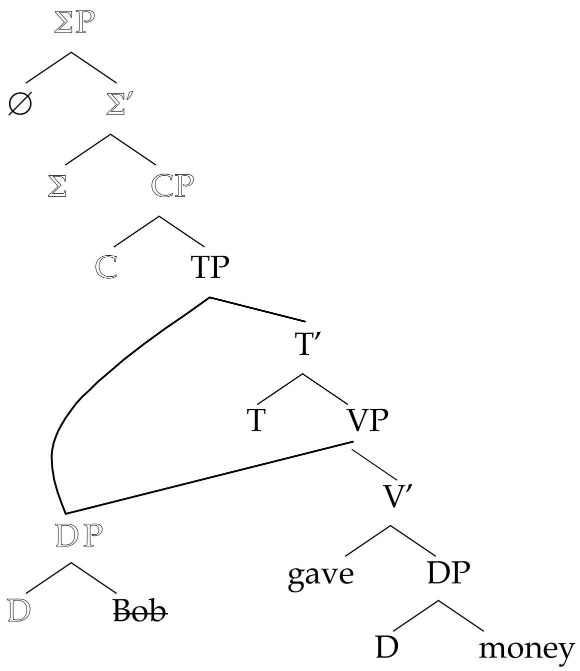
h:



i:

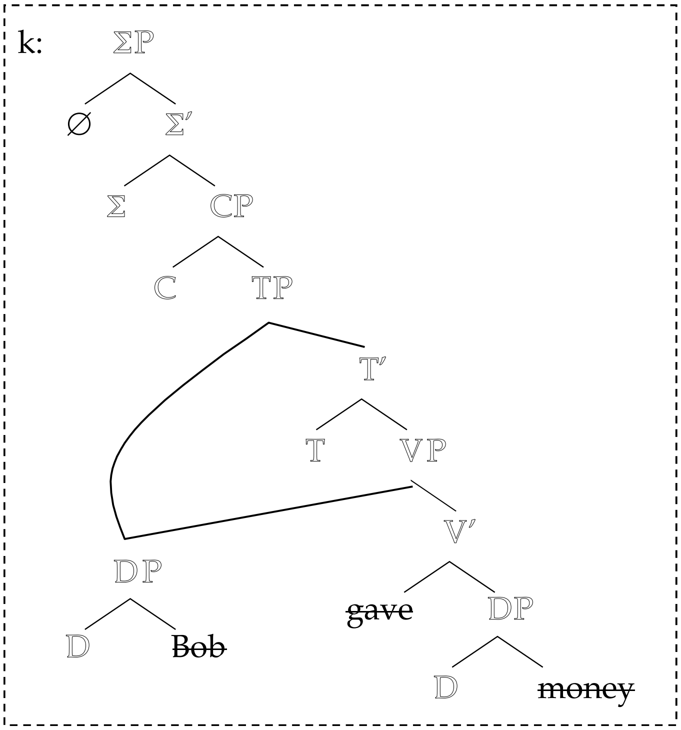


j:



Once again, needs to apply to ensure LCA-satisfaction in the subsequent steps, as shown in (36k).

(36)

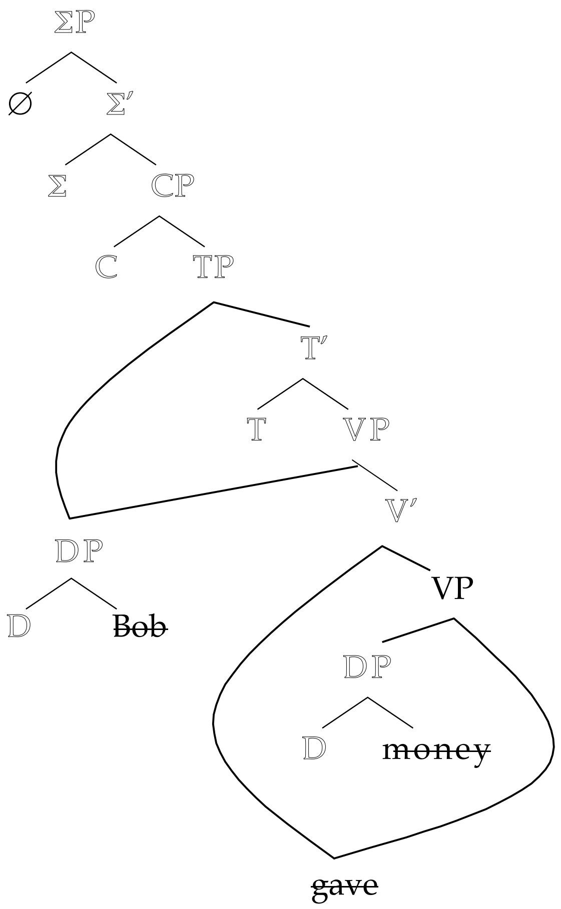


← in the Syntax

[#Bob#]$^{\cap}$[#gave#$^{\cap}$#money#] ← out to Phonology

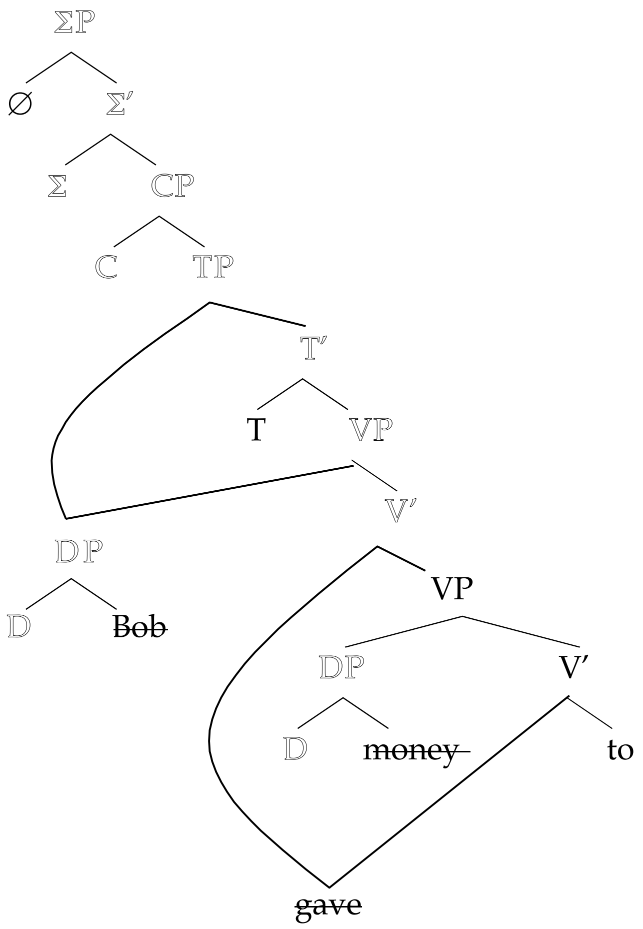
In the next step, the verb gave lowers as remerges as a new sister to the DP money, so that it can find itself in the spine of the tree at the subsequent stage in order for the indirect object to under sisterhood, as required by the *Local Merge* condition.

(36) l:



Then the indirect object starts to be built, with the introduction of the preposition **to** as the sister of **gave**, as in (36m).

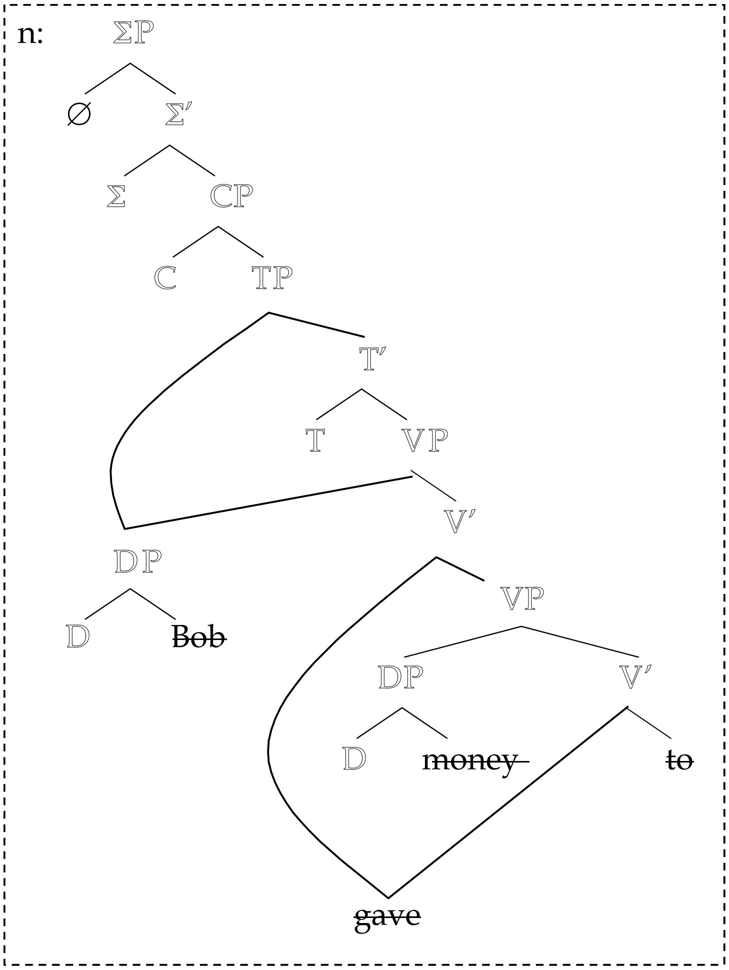
(36) m:



At this point, the first derivational round is forced to terminate, leaving an incomplete phrase-marker to be finished in the next round. This is so because, if the WH-phrase **what** is built at this point, it would be impossible for it to be further remerged in the lowest spec/CP position of the subservient clause, due to the lack of c-command, as discussed in the previous sections.

Before the second round begins, the remaining structure is spelled-out, as in (36n).

(36)



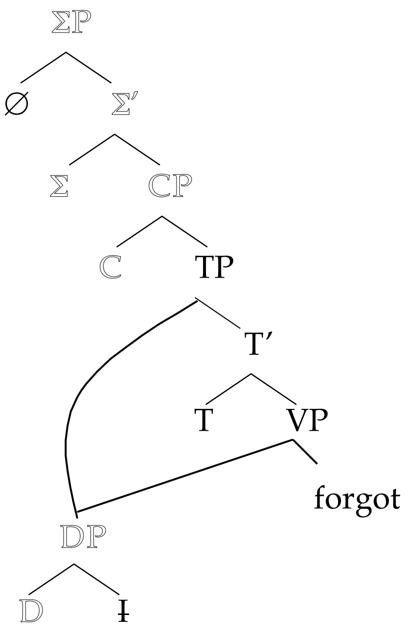
← in the Syntax

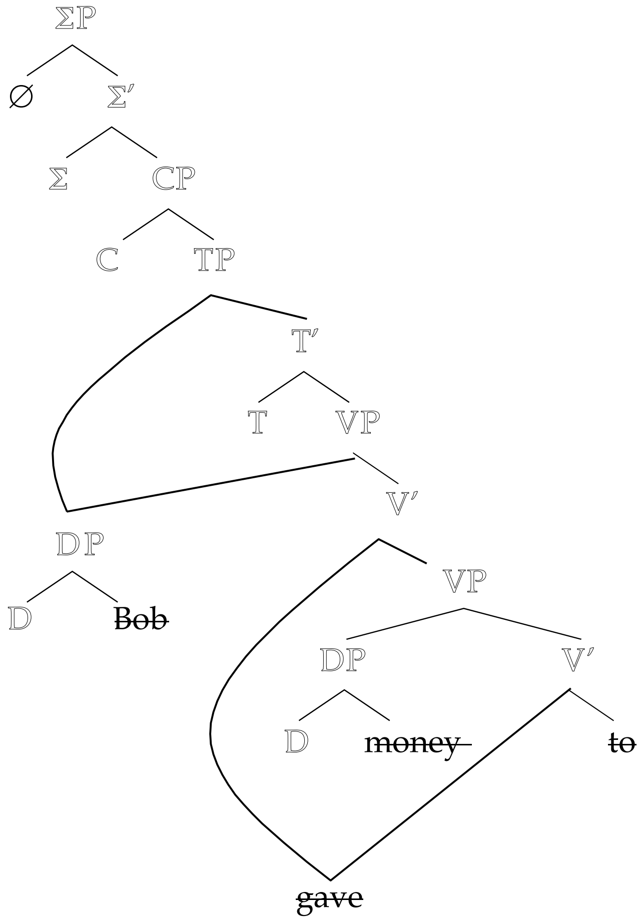
[#Bob#]$^{\cap}$[#gave#$^{\cap}$#money#]$^{\cap}$[#to#]

← out to Phonology

The system then shifts to numeration Ψ. First the lexical tokens in Ψ that are not part of the intersection start being combined in the usual top-to-bottom fashion, all the way down to the matrix verb **<u>forgot</u>**, as in (36o).

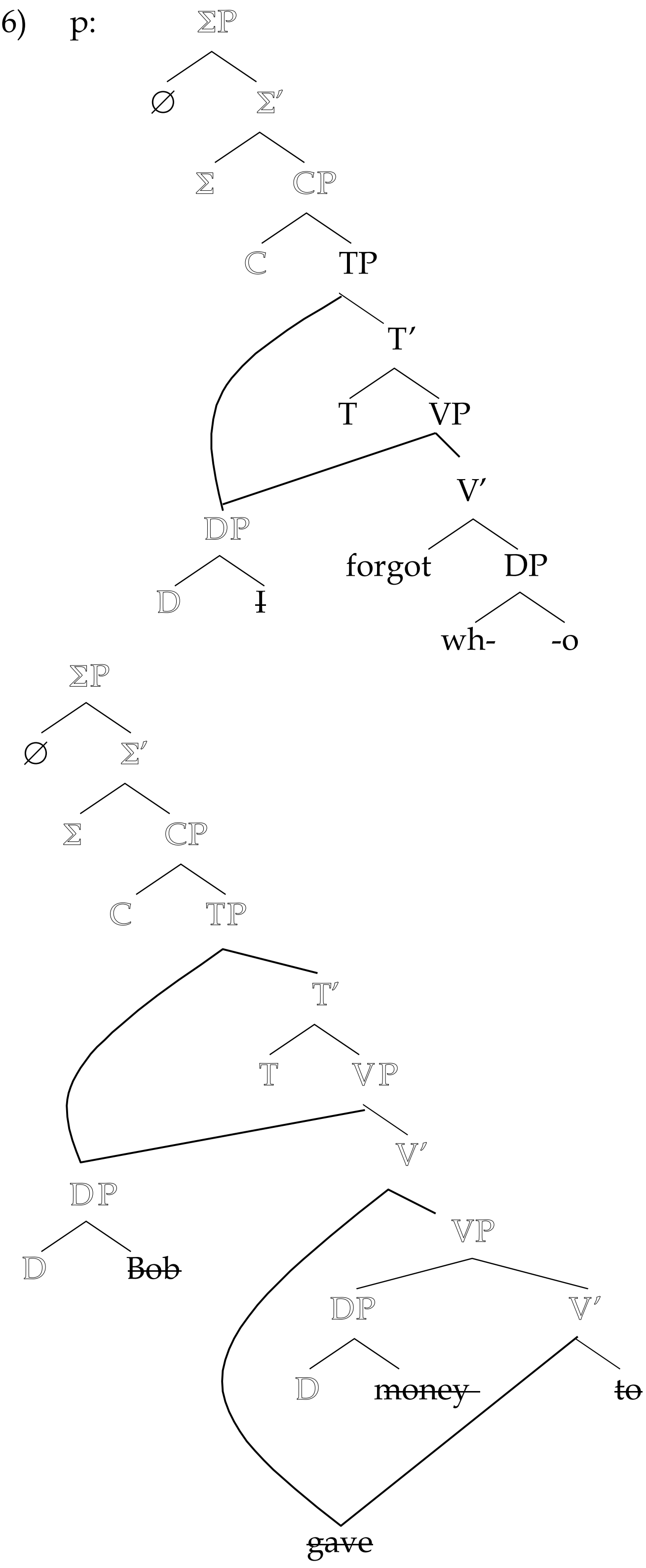
(36) o:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
forgot
DP
D
~~I~~


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~Bob~~
VP
DP
V′
D
~~money~~
~~to~~
~~gave~~

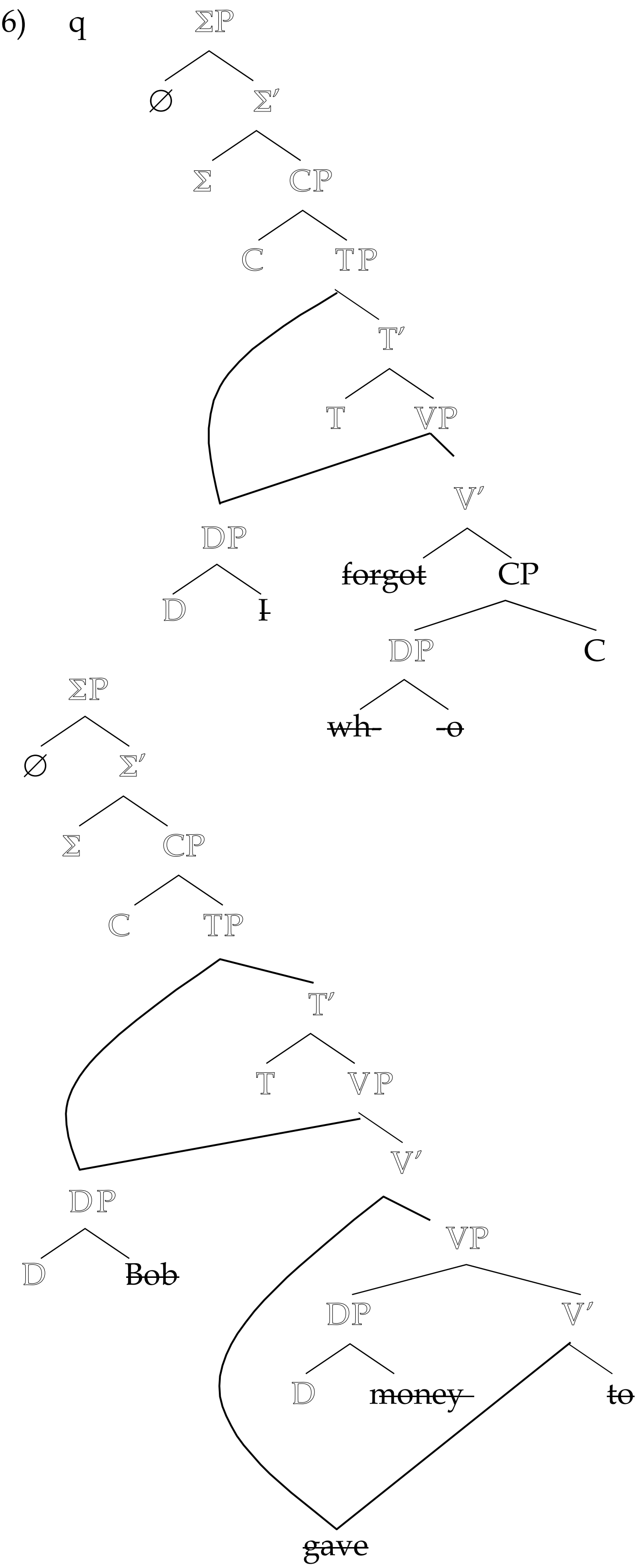
Subsequently, the WH-phrase **who** is built as a temporary sister to **forgot**, as in (36p).

(36) p:

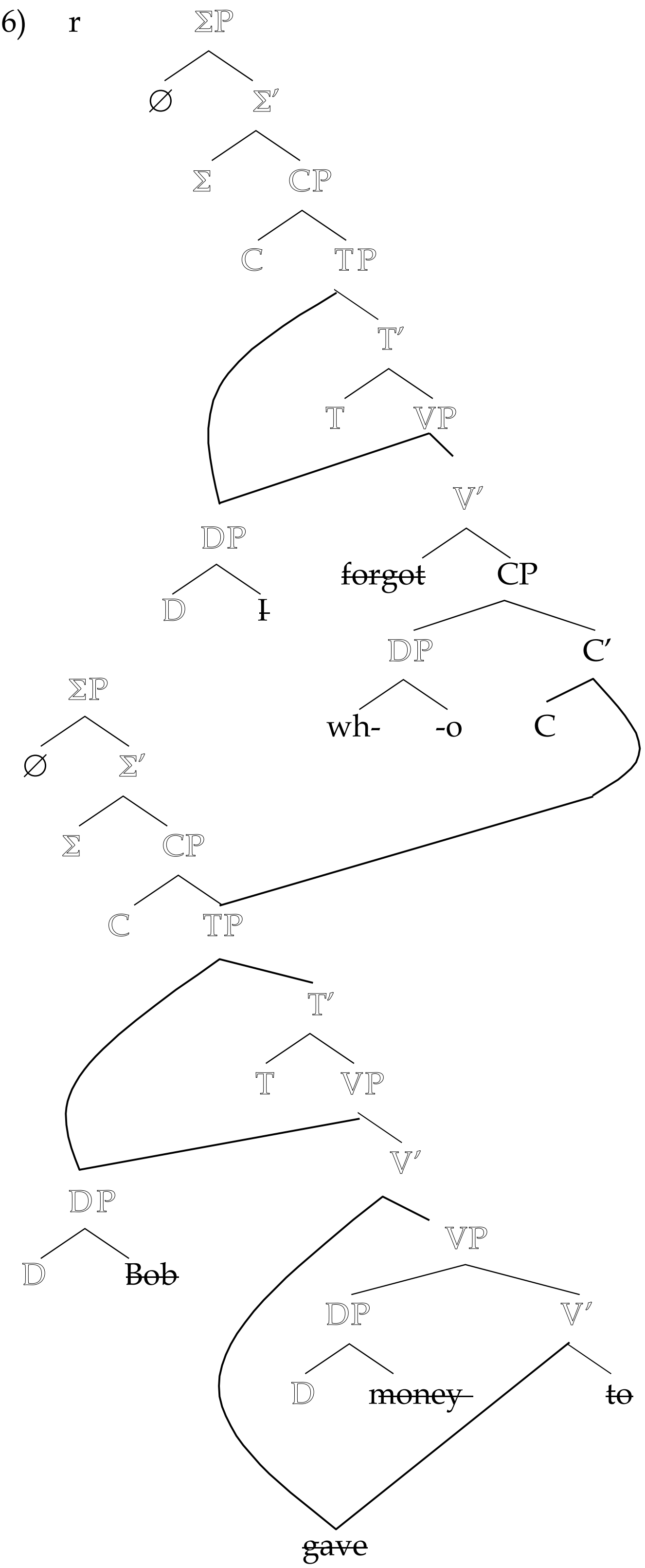

ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~I~~
forgot
DP
wh-
-o
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~Bob~~
VP
DP
V′
D
~~money~~
~~to~~
~~gave~~

After that, spell-out applies, and the construction of the subservient clause continues with the merge of $C_{[+WH]}$ as the (temporary) sister to **<u>who</u>**, as in (36q), and subsequent remerge of the TP of the master clause as the new sister to $C_{[+WH]}$, as in (36r). Eventually, who lowers into its theta-position inside the shared TP, as in (36s).

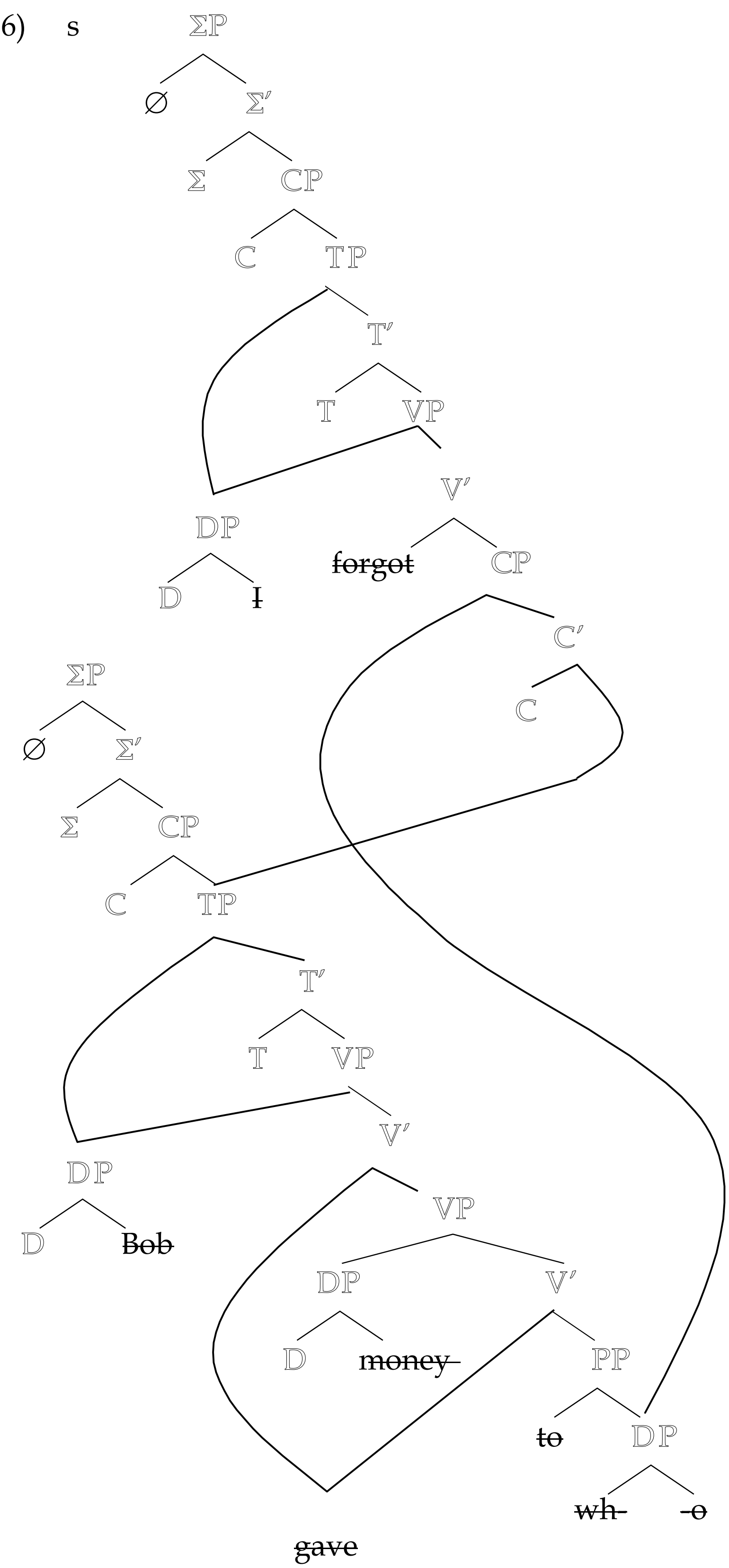
(36) q


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~I~~
~~forgot~~
CP
DP
C
~~wh-~~
~~-o~~
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~Bob~~
VP
DP
V′
D
~~money~~
~~to~~
~~gave~~

(36) r


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~I~~
~~forgot~~
CP
DP
C′
wh-
-o
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~Bob~~
VP
DP
V′
D
~~money~~
~~to~~
~~gave~~

(36)



[#Bob#]$^{\cap}$[#gave#$^{\cap}$#money#]$^{\cap}$[#to#]$^{\cap}$[#I#$^{\cap}$#forgot#$^{\cap}$[#who#]

The crucial step in the derivation above is the one between (36-l) and (36-m). It is at that point that the system took the direction towards (33a) and not (33b) (repeated below as (37) and (38)).

(37) Bob gave money to I forgot who.

(38) * Bob gave money I forgot to who

In (36m), the system is pushing the derivation as far as it can go by integrating the preposition **to** to the phrase marker. If the first derivational round goes beyond that point, and the WH-phrase is constructed at the bottom of the tree, then that very WH-phrase will not be able to be remerged into the subservient clause in the next derivational round, since it will fail to c-command the target of remerge. That would leave WH-features unchecked in the structure. Therefore, spell-out applies in (37n), and the first derivational round terminates, leaving an incomplete phrase marker to be finished in the second round.

Now, let us consider Romance, which exhibits the opposite word-order pattern, as repeated below in (39).

(39) *Romance (Portuguese)*

a: * Bob deu dinheiro pra eu me esqueci quem.

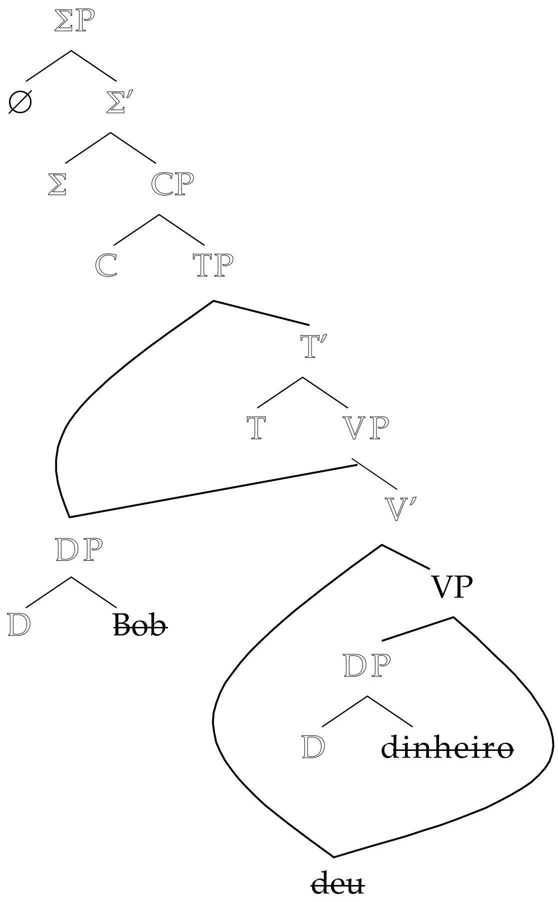
Bob gave money to I REFL-forgot who.

b: Bob deu dinheiro eu me esqueci pra quem.

Bob gave money I REFL-forgot to who

The derivation for (39) would be identical to the one of (38) up to the crucial point, which is the equivalent of (36-l). This is shown in (40a).

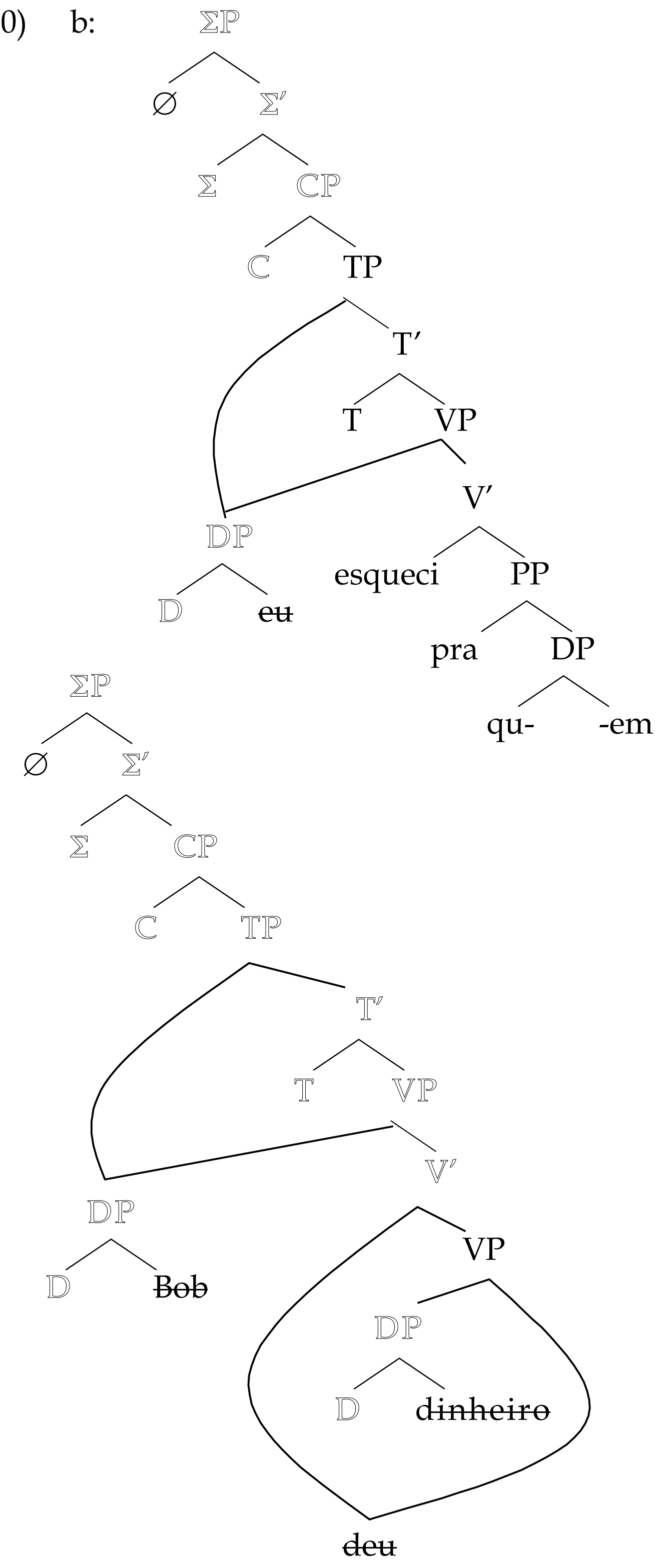
(40) a:



Just like in (36-l), the system pushes the derivation as far as it can go, which means that the introduction of the WH-phrase must be delayed, for the same reasons discussed above. In Romance, however, there is one extra requirement on WH-chains which makes it impossible for the derivation to go as far as in the English case. Not only the introduction of the WH-phrase must be delayed, but also the introduction of the preposition governing it. This is so because of whatever parametric setting ultimately requires pied-piping in Romance, demanding that the whole PP containing the WH-phrase be in the relevant spec/CP. If that is the case, then the first derivational round must terminate right before the introduction of the preposition.

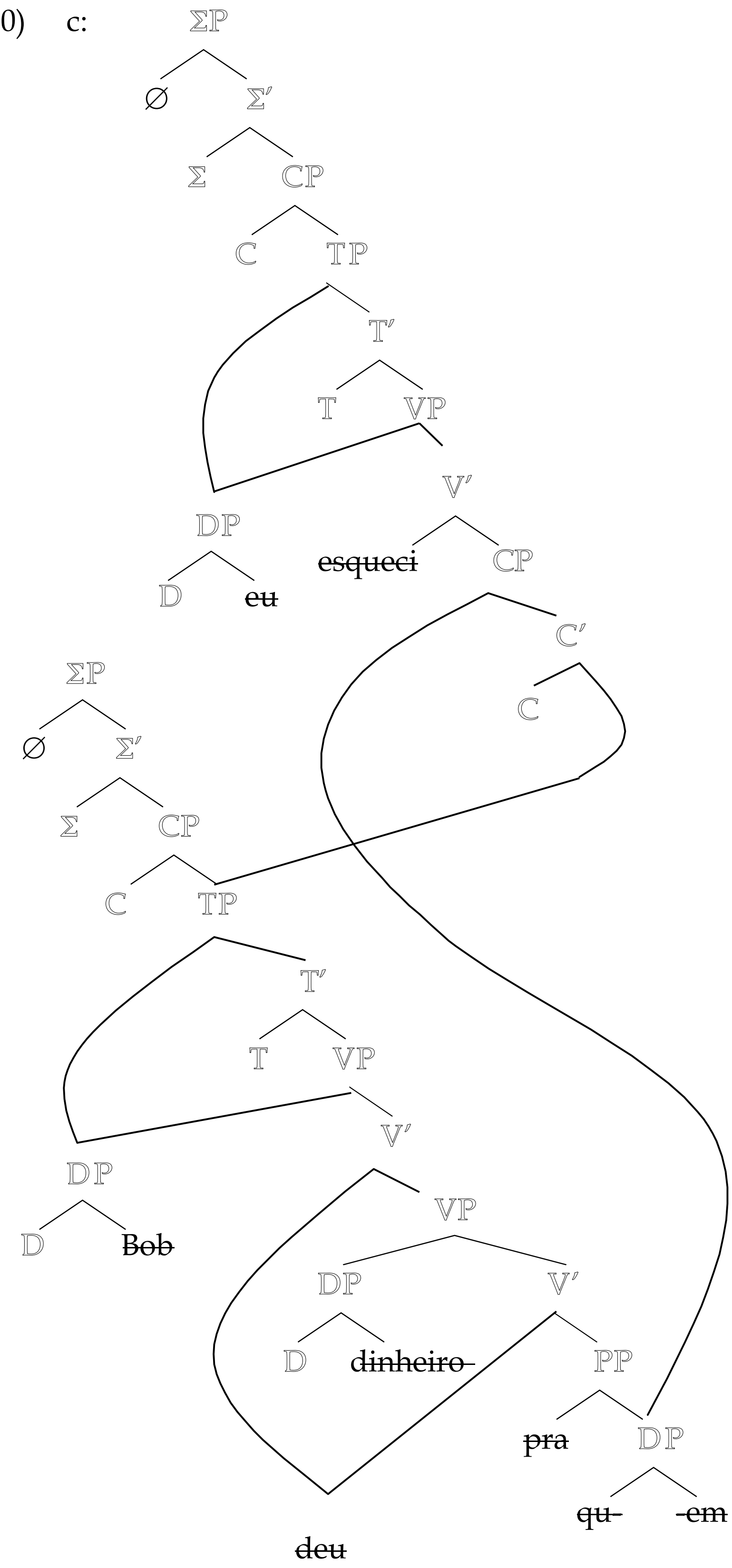
At the relevant point in the second derivational round, the preposition is introduced, immediately followed by the introduction of the WH-phrase, as shown in (40b).

(40) b:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~eu~~
esqueci
PP
pra
DP
qu-
-em
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~Bob~~
VP
DP
D
~~dinheiro~~
~~deu~~

Eventually, the incomplete TP is shared and further completed by the end of the second derivational round, when the (pied-piped) PP lowers/remerges into its theta-position, as in (40c).

(40) c:



[#Bob#]$^{\cap}$[#deu#$^{\cap}$#dinheiro#]$^{\cap}$[#eu#$^{\cap}$#me#$^{\cap}$#esqueci#$^{\cap}$#pra#$^{\cap}$#quem#]

V.5. Multiple Amalgamation

In **§II.3**, I have shown that there can be multiple 'invasive clauses' for the same 'invaded clause', as in Lakoff's (1974) classical example reproduced in (41).

(41) John invited *you will never guess how many people* to *you can imagine what kind of a party* at *it should be obvious where* with *God only knows what purpose in mind*, despite *you can guess what pressures*.

As discussed in **§III.3.5.2**, such constructions pose a serious problem to the remnant-movement analysis to amalgamation, which wrongly predicts that one invasive clause should contain the other, in a way that is inconsistent with the actual semantic interpretation. Moreover, locality constraints on movement would have to be violated in order to generate multiple amalgams, as also discussed in **§III.3.5.2**.

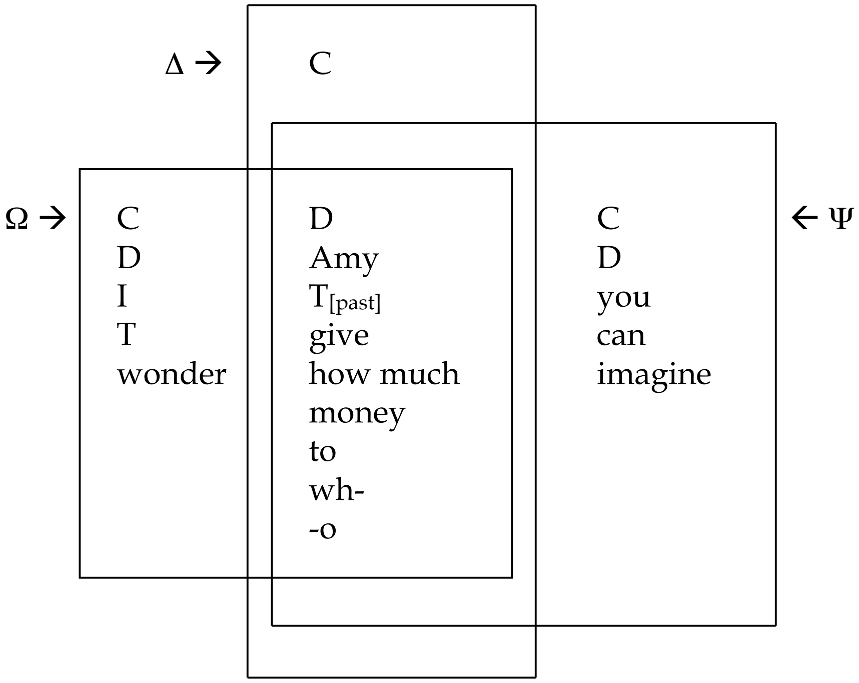
This section is dedicated to a demonstration of how multiple amalgams are generated in the framework here proposed. In fact, there is nothing special about the mechanics of multiple amalgamation under this approach. Those constructions are just like ordinary syntactic amalgams, except that they are generated from an input with more numerations, which yields more than one parallel subservient clauses, aside from the master clause.

For concreteness, consider (42).

(42) Amy gave I wonder how much money to you know who.

The input to (42) would be as in (43) below.

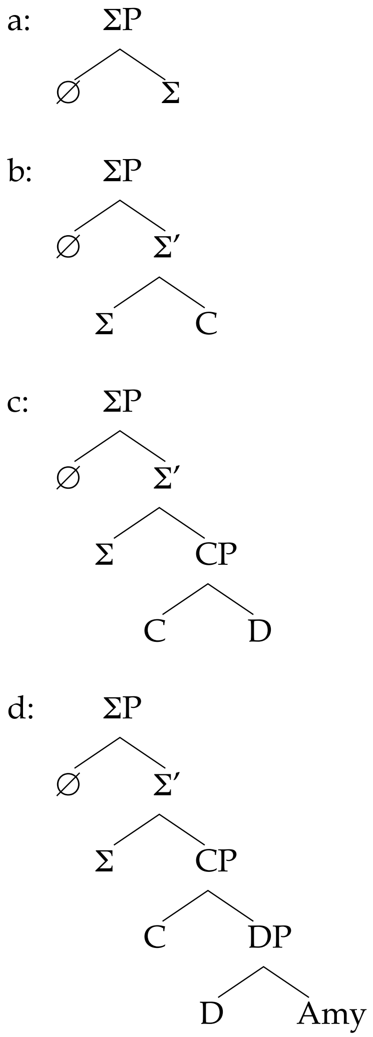
(43)



As usual, the system randomly takes one of the intersecting numerations as the starting point. In this case, the starting point is Δ.

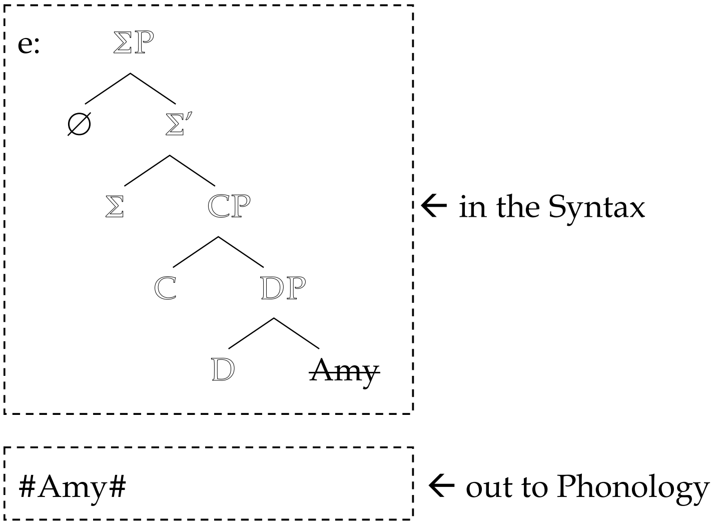
Then, the subcomputation that leads to the master clause begins with the starting axiom, as in (44a). Then, the derivation proceeds by tucking in lexical items, one by one, at the very bottom of the tree, up to step (44d).

(44)



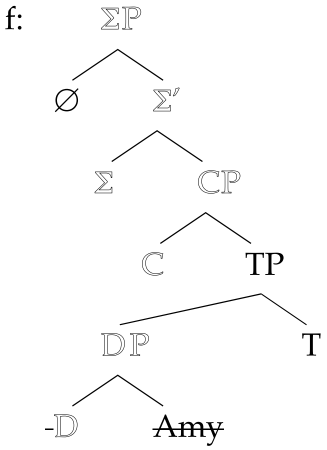
At this point, spell-out needs to apply in order to guarantee that the LCA will be satisfied in the consecutive steps, as shown in (44e).

(44)

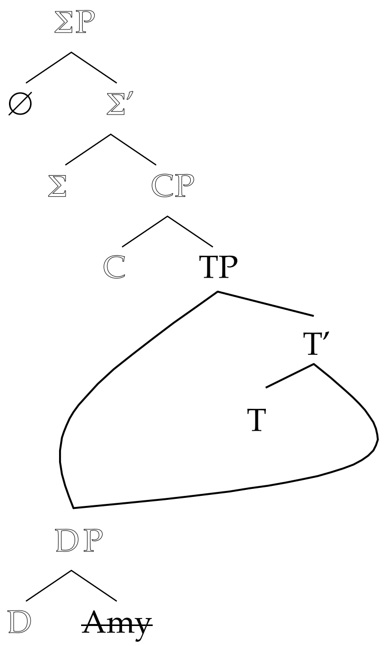


After that, the derivation proceeds in the usual (top-to-bottom) fashion with the introduction of T (cf. (44f)), the lowering of the subject (cf. (44g)), and the introduction of the verb (cf. (44h)).

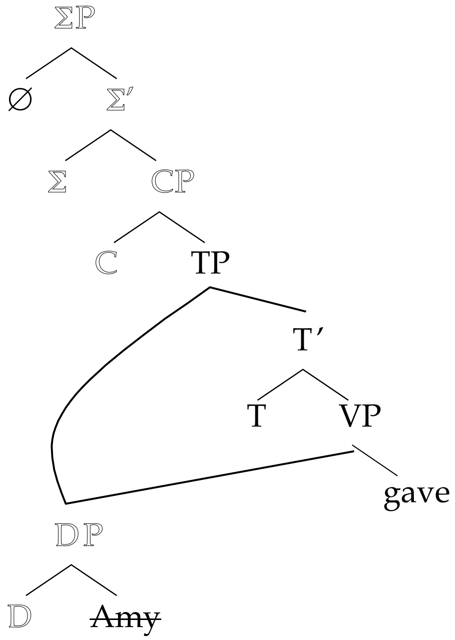
(44)



(44) g:

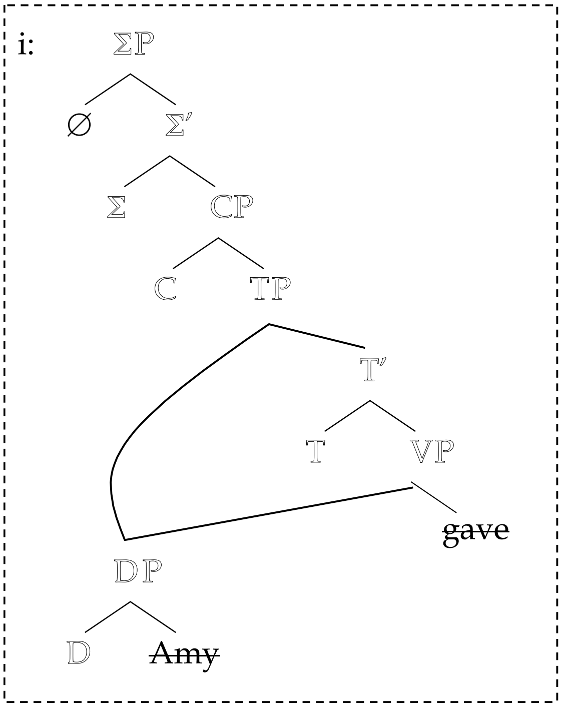


h:



As discussed in the previous sections, the system must delay the construction and integration of the WH-phase until the next derivational round, or else that WH-phase will not be able to be merged at the relevant spec/CP due to the lack of c-command. Therefore, the structure in (44h) must be spelled-out, as shown in (44i).

(44)



← in the Syntax

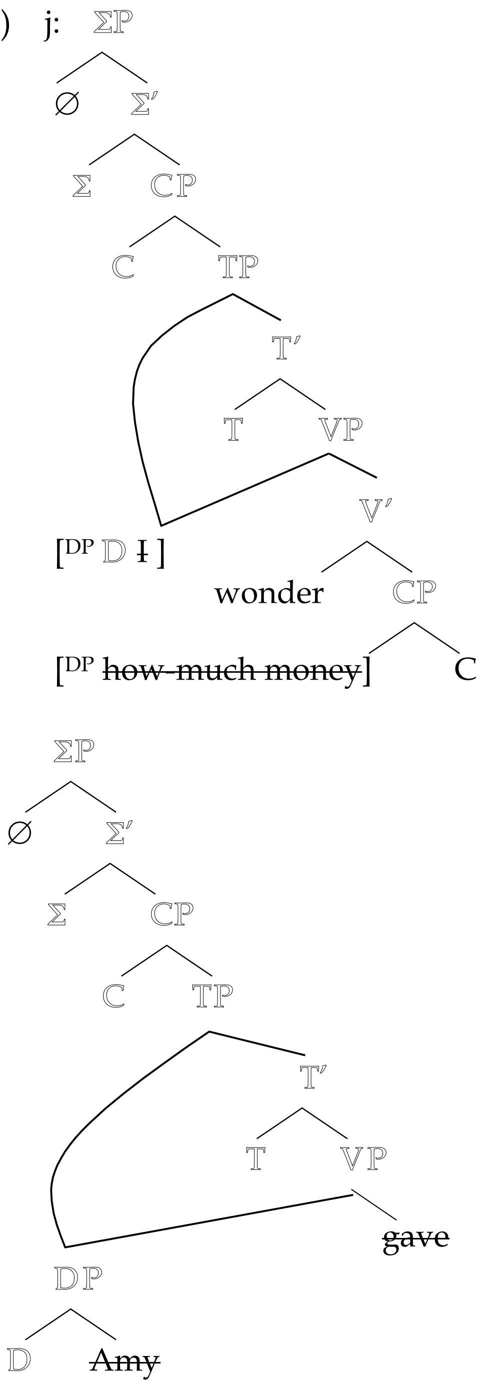
[#Amy#]$^{\frown}$[#gave#]

← out to Phonology

The computational system then shifts its attention to numeration Ω (it could have been Ψ, as I will discuss later) to continue the structure-building process. This means that, in a second derivational round, a parallel matrix clause will be built from the items in Ω, which will be ***subservient*** to the (master) matrix clause built from Δ in the first derivational round.

The higher portion of that subservient clause is built in the usual fashion, by integrating the relevant lexical tokens in the non-shared part of Ω, into a parallel phrase marker built from the top downwards, via successive applications of tucking-in. Thus, at some point, the derivational workspace will contain two unconnected matrix clauses as in (44j), whose corresponding PF-string is as in (44k).

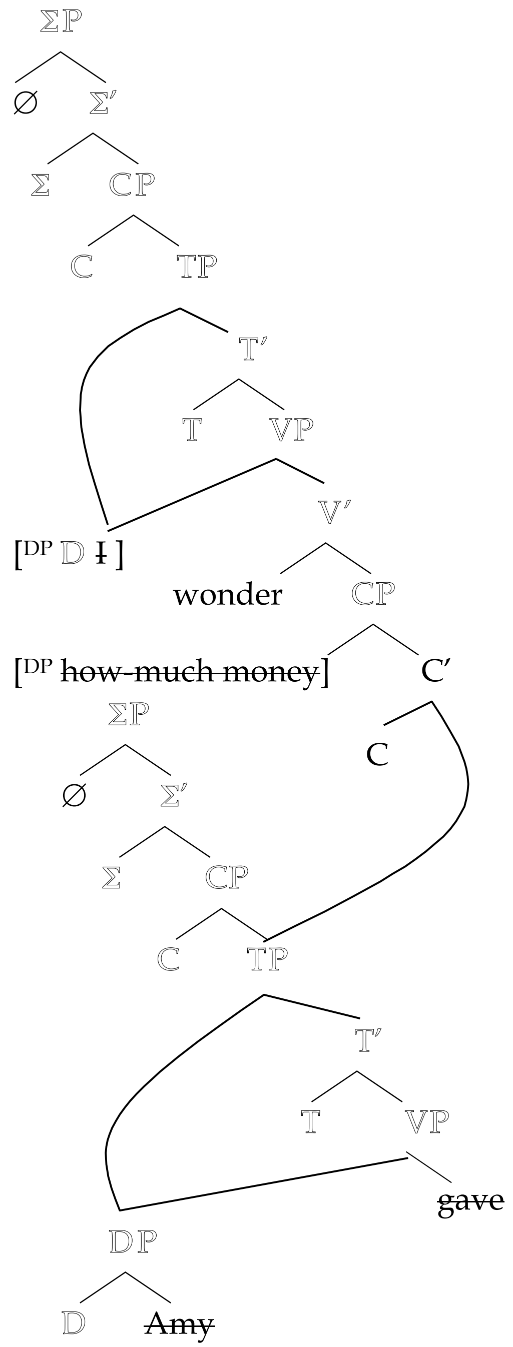
(44) j:



(44) k: [#Amy#]$^{\cap}$[#gave#]$^{\cap}$[#I#$^{\cap}$#wonder#$^{\cap}$#how-much#$^{\cap}$#money#]

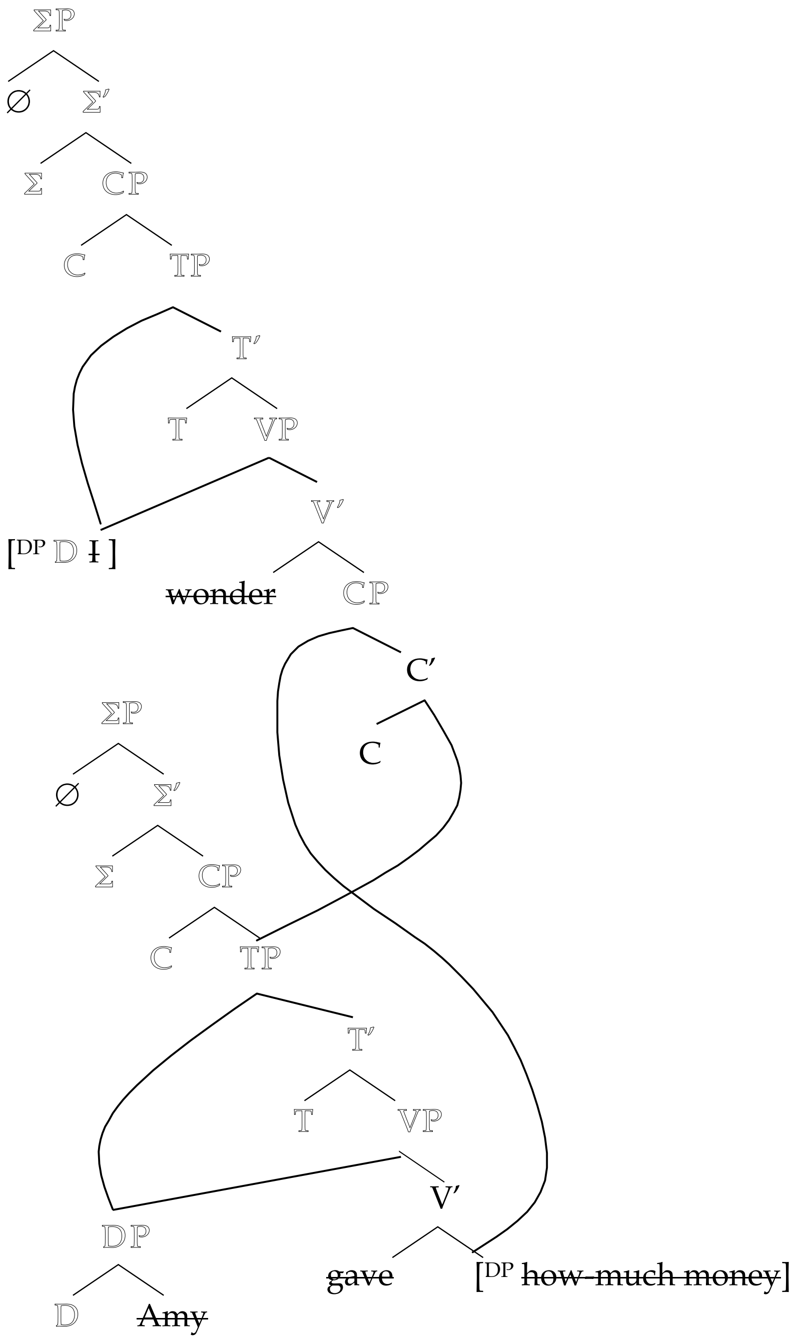
The next step consists of the sharing of the TP of the master clause, which remerges as the sister of the lower C of the subservient clause, as in (44-l)

(44-l)

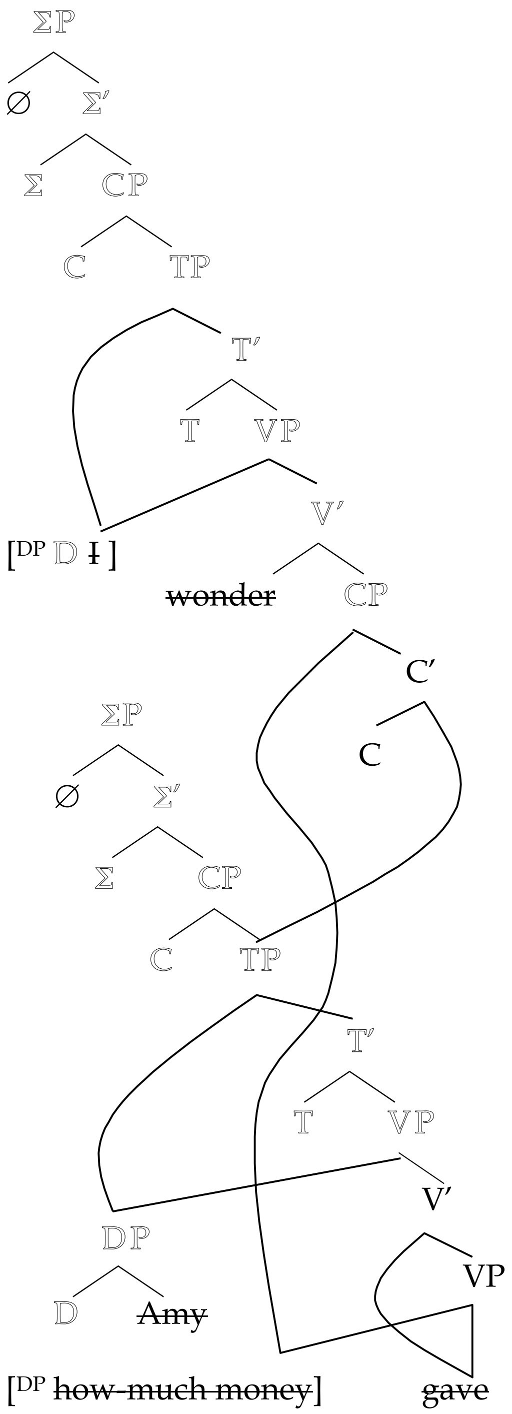


The second derivational round, then, proceeds in the usual fashion, taking lexical items from the intersection between Δ and Ω, and integrating them into the shared TP, continuing the job left unfinished in the first derivational round. This is shown in (44m) through (44p)

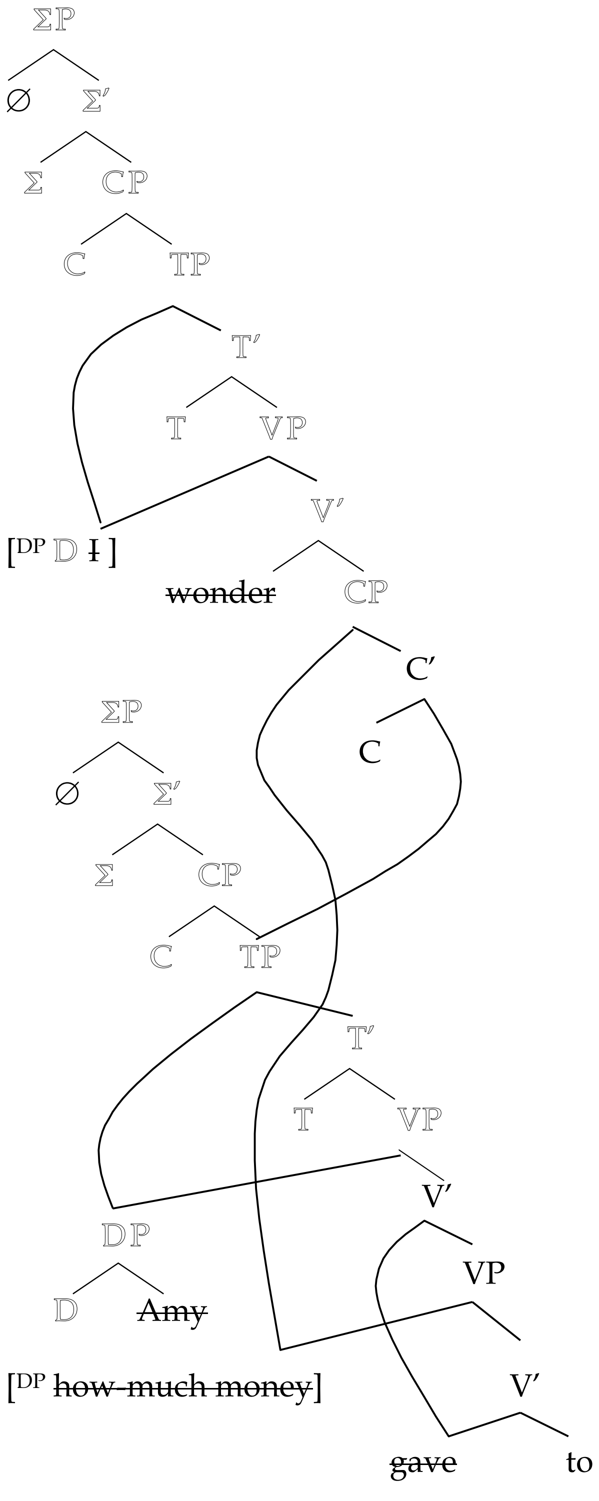
(44) m:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
[DP D ~~I~~]
~~wonder~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~Amy~~
~~gave~~
[DP ~~how-much money~~]

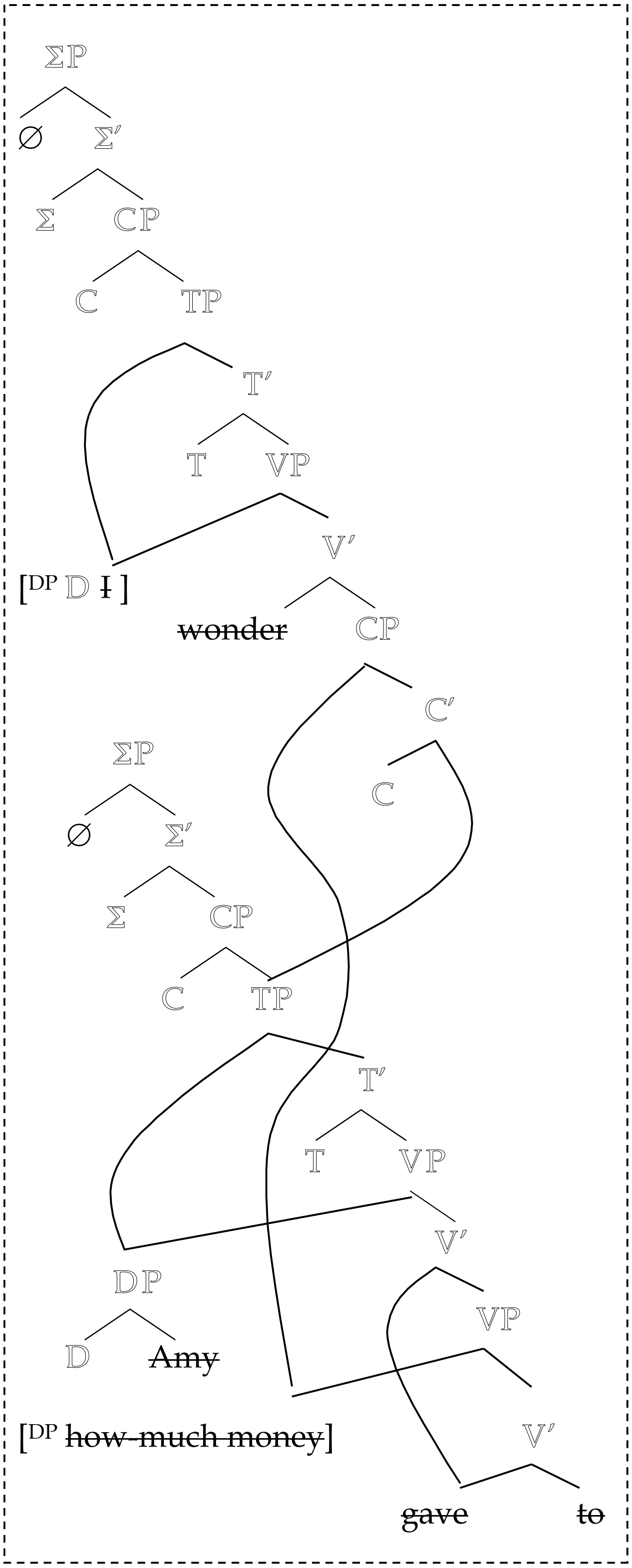
(44) n:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
[DP D ~~I~~]
~~wonder~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
VP
DP
D
~~Amy~~
[DP ~~how-much money~~]
~~gave~~

(44) o:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
[DP D ~~I~~]
V′
~~wonder~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
VP
D
~~Amy~~
V′
[DP ~~how much money~~]
~~gave~~
to

(44) p:



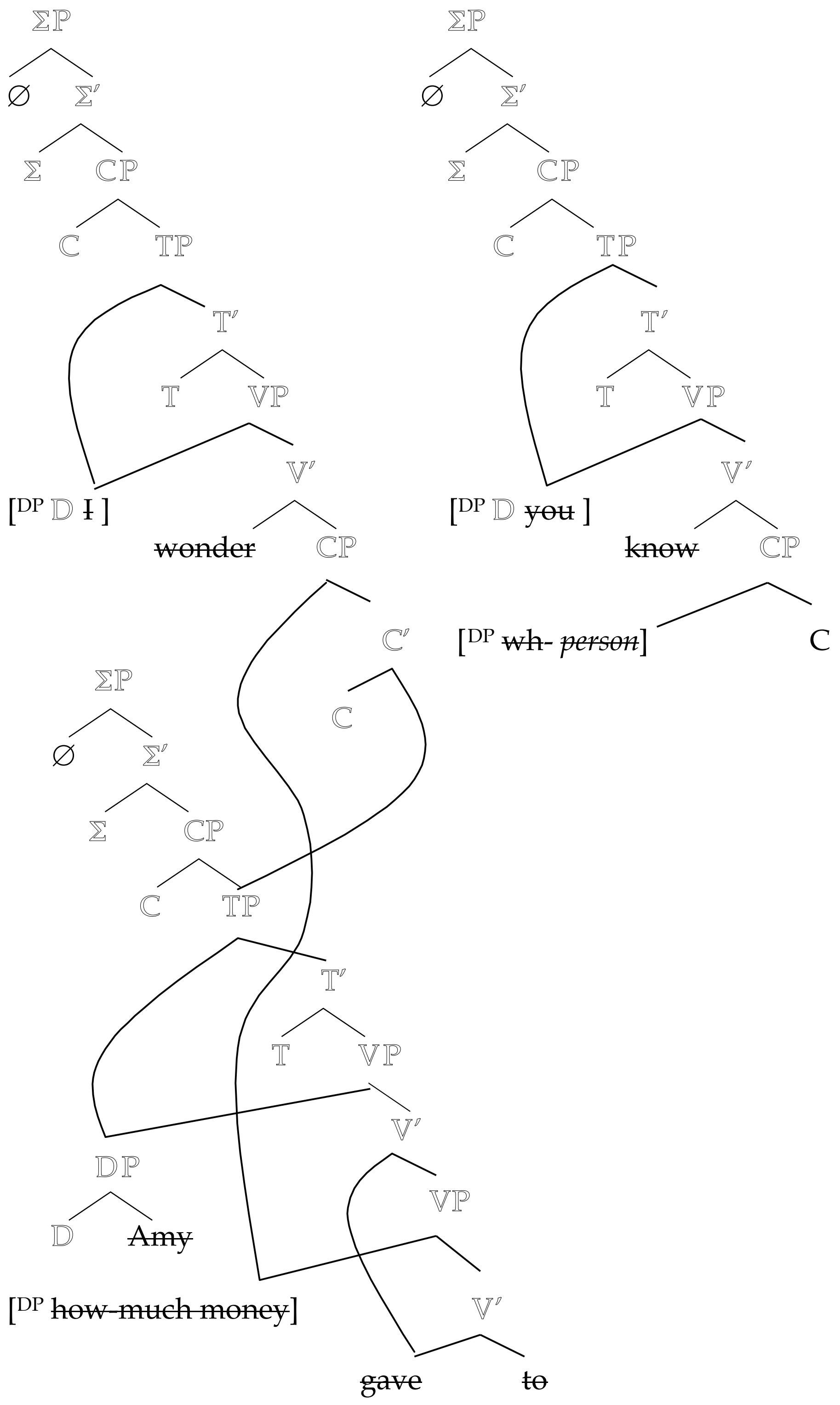
[#Amy#]$^{\cap}$[#gave#]$^{\cap}$[#I#$^{\cap}$#wonder#$^{\cap}$#how-much#$^{\cap}$#money#]$^{\cap}$[#to#]

The second derivational round ends at this point, right before the introduction of the WH-phrase, for reasons already discussed.

The computational system then shifts its attention to numeration Ψ, and the structure-building process continues. This means that a third derivational round starts, which will eventually build yet another parallel matrix clause (from the lexical tokens in Ψ). That new matrix clause will be 'behind' the two previous one, and it will also figure as ***subservient*** to the (master) matrix clause built from Δ in the first derivational round.

The higher portion of the new subservient clause is built in the usual fashion, by integrating the relevant lexical tokens in the non-shared part of Ψ, into a parallel phrase marker built from the top downwards, via successive applications of tucking-in. Thus, at some point, the derivational workspace will contain two matrix clauses connected at an embedded TP node, plus an unconnected parallel matrix clause, as in (44q), whose corresponding PF-string is as in (44r).

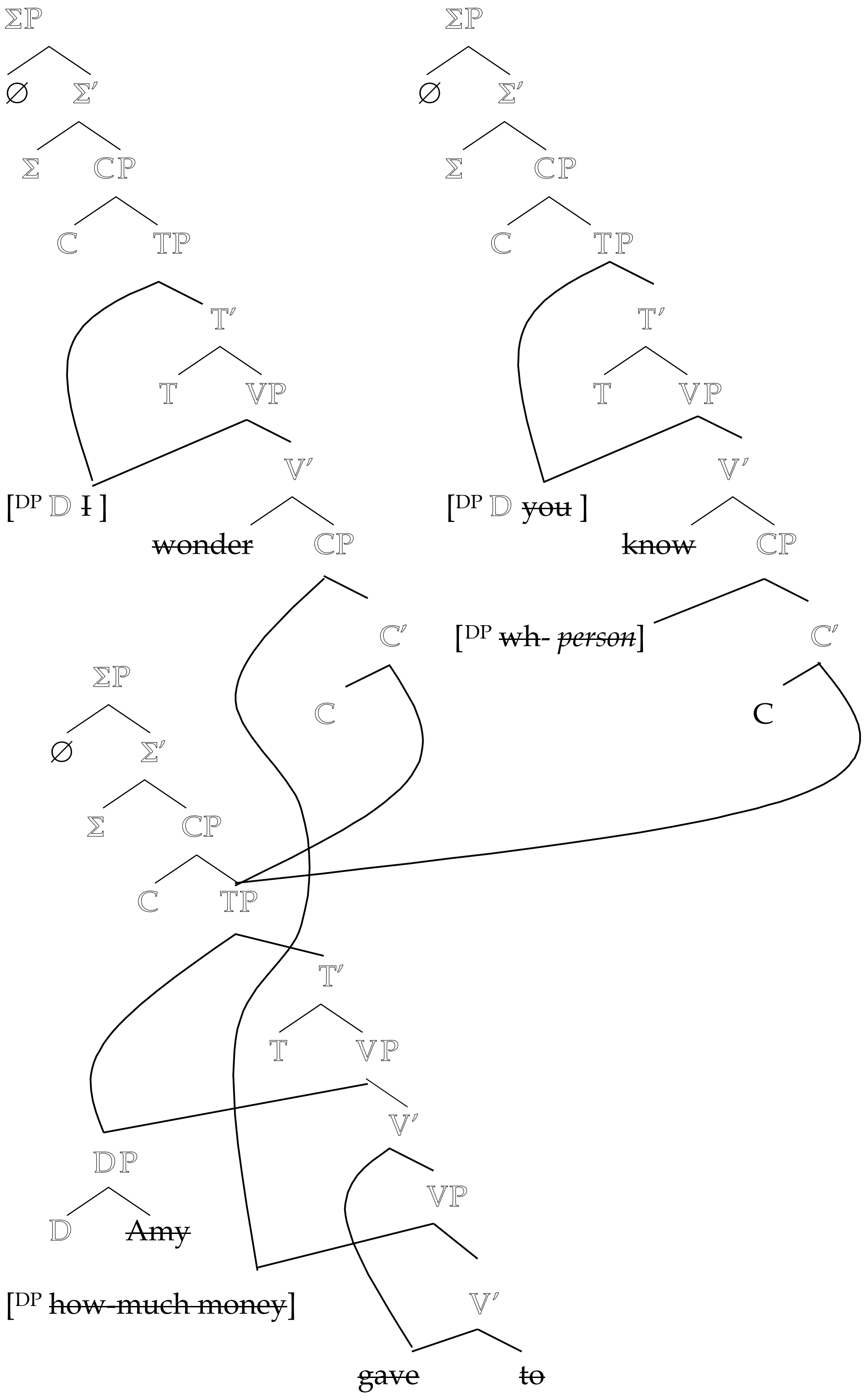
(44) q:



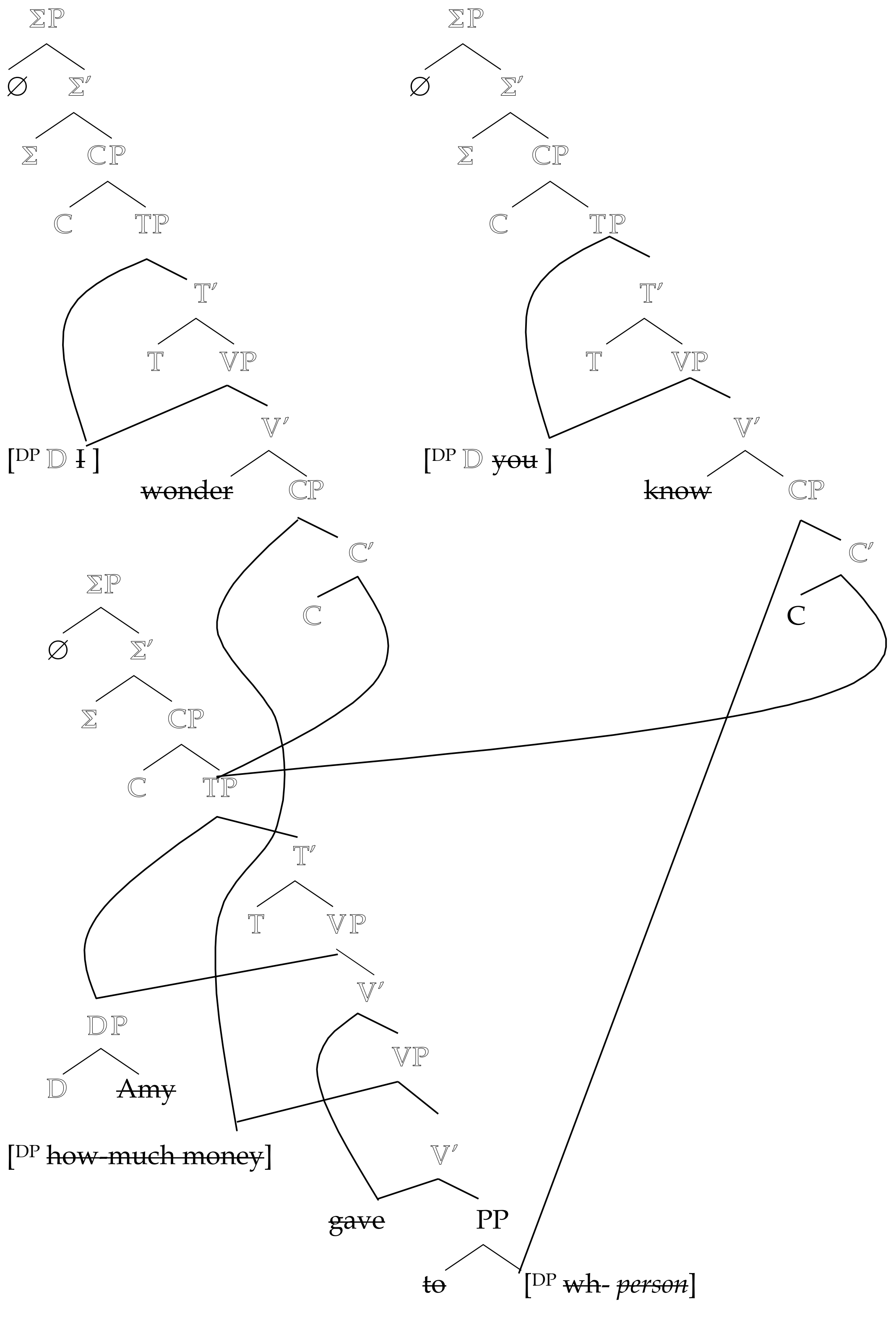
(44r) [#Amy#]$^{\cap}$[#gave#]$^{\cap}$[#I#$^{\cap}$#wonder#$^{\cap}$#how-much#$^{\cap}$#money#]$^{\cap}$[#to#]

Then, the embedded TP is shared one again, and gains a third sister: the lower C of the second subservient clause, as in (44s). Subsequently, the WH-phrase who is remerged in its theta-position inside the shared TP, as in (44t). Eventually, the whole structure in (44t) undergoes spell-out, and the final representation gets associated with the PF-string in (44u).

(44) s:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
[DP D ~~I~~]
~~wonder~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
[DP D ~~you~~]
~~know~~
CP
[DP ~~wh-~~ *person*]
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~Amy~~
VP
[DP ~~how-much money~~]
V′
~~gave~~
~~to~~

(44) t:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
[DP D ~~I~~]
~~wonder~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
[DP D ~~you~~]
~~know~~
CP
C′
C
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
~~Amy~~
VP
[DP ~~how-much money~~]
V′
~~gave~~
PP
~~to~~
[DP ~~wh-~~ *person*]

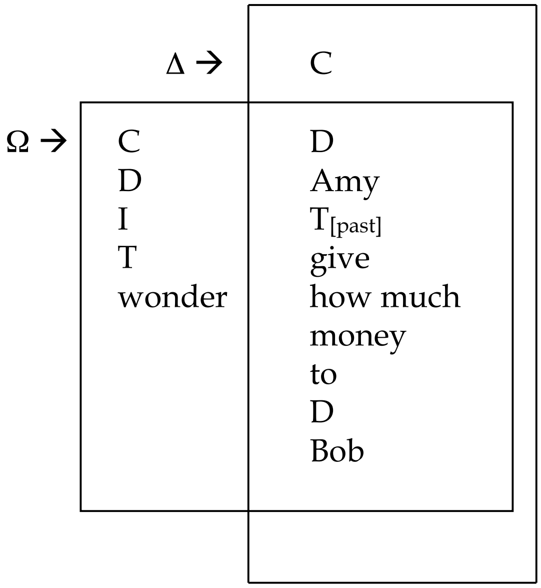
(44u)

[#Amy#]⁀[#gave#]⁀[#I#⁀#wonder#⁀#how-much#⁀#money#]⁀[#to#]⁀[#you#⁀#know#⁀#who#]

There is one important issue about multiple amalgamation which has not been discussed yet. I have been assuming all along that, when the input contains multiple intersecting numerations, the order in which those numerations are mapped to phrase markers is the mere result of a random choice. Thus the status of 'master clause' or 'subservient clause' is not encoded in the numeration in any way. However, in all derivations demonstrated so far, the outcome crucially depended on that particular choice of which numeration to be mapped at a given derivational round. The question, then, is: what would happen if the computational system does things in a different order?

In order to address this issue, let us first consider cases of simple amalgamation. Take (45) as the staring point.

(45)



In cases like this, there would be two matrix clauses to be built, one from each numeration. Notice that one numeration (i.e. Ω) has many lexical tokens in it besides the ones in the intersection, whereas the other one (i.e. Δ) has only one token of C besides the shared lexical tokens. Eventually, the matrix clause built from Ω will be a statement that the speaker wonders how much is the amount of money ***x***, such that Amy gave ***x*** to Bob. On the other hand, the matrix clause to be built from Δ will be a statement that Amy gave an amount ***x*** of money to Bob. In principle, either matrix clause can have the status of the master clause (which makes the other one subservient to it). That will depend only on which numeration is randomly chosen for the first and for the second derivational round.

This system predicts that both possibilities should surface, as neither is more economical than the other and both converge. The two possibilities are given in (46a) and (46b), which correspond to the structures in (47) and (48), respectively.[142]

(46) a: *order of derivational rounds:* <*Ω, Δ*>

I wonder how much money Amy gave to Bob.

b: *order of derivational rounds:* <*Δ, Ω*>

Amy gave I wonder how much money to Bob.

[142] In what follows, traces are mere notational devices, to be understood as remerged phrases.

(47)
$$\begin{array}{r} [^{CP} \\ [^{CP}\,[^{TP}\text{ I }[^{VP}\text{ wonder }[^{CP}\,[\text{how-much money}]_1 \end{array} \left[\begin{array}{c} \text{Amy gave } t_1 \text{ to Bob} \end{array}\right] \begin{array}{l}] \\ ]\,]\,]\,] \end{array}$$

(48)
$$\begin{array}{r} [^{CP}\,[^{TP}\text{ I }[^{VP}\text{ wonder }[^{CP}\,[\text{how-much money}]_1 \\ [^{CP} \end{array} \left[\begin{array}{c} \text{Amy gave } t_1 \text{ to Bob} \end{array}\right] \begin{array}{l}]\,]\,]\,] \\ ] \end{array}$$

The notational convention used in (47) and (48) presents the master clause at the top, and the subservient clause at the bottom.

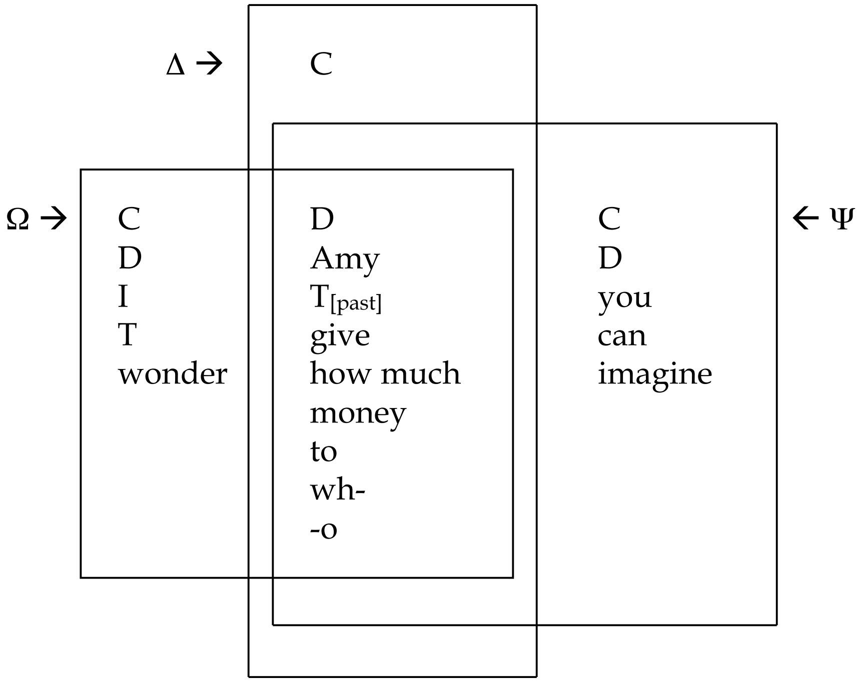
Notice that (46b)/(48) is homophonous to the simpler structure in (49), which comes from a simpler input, containing only the numeration Ω.

(49) [^{CP} [^{TP} I [^{VP} wonder [^{CP} [how-much money]$_1$ [Amy gave t$_1$ to Bob]]]]]

Now, let us go back to cases of multiple amalgamation, where things get more interesting.

Consider again the input in (43), repeated below as (50)

(50)



The logical possibilities are the ones listed in (51), and their corresponding structures are the ones in (52). All of these are equally economical and equally convergent. Notice that they do not all correspond to the same LF and/or the same PF representations

(51) a: *order of derivational rounds: <Δ, Ω, Ψ>*

Amy gave I wonder how much money to you know who.

b: *order of derivational rounds: <Δ, Ψ, Ω>*

Amy gave you know how much money to I wonder who.

c: *order of derivational rounds: <Ω, Δ, Ψ>*

I wonder how much money Amy gave to you know who.

d: *order of derivational rounds: <Ω, Ψ, Δ>*

I wonder how much money Amy gave to you know who.

e: *order of derivational rounds: <Ψ, Ω, Δ>*

You know how much money Amy gave to I wonder who.

f: *order of derivational rounds: <Ψ, Δ, Ω>*

You know how much money Amy gave to I wonder who.

(52a)

$$\begin{array}{r} [^{CP} \\ [^{CP}\,[^{TP}\text{ I }[^{VP}\text{ wonder }[^{CP}\,[\text{how-much money}]_1 \\ [^{CP}\,[^{TP}\text{ you }[^{VP}\text{ know }[^{CP}\text{ who}_2 \end{array} \left[\begin{array}{c} \\ \text{Amy gave } t_1 \text{ to } t_2 \\ \\ \end{array}\right] \begin{array}{l}] \\ ]\,]\,]\,] \\ ]\,]\,]\,] \end{array}$$

(52b)

$$\begin{array}{r} [^{CP} \\ [^{CP}\,[^{TP}\text{ you }[^{VP}\text{ know }[^{CP}\,[\text{how-much money}]_1 \\ [^{CP}\,[^{TP}\text{ I }[^{VP}\text{ wonder }\;[^{CP}\text{ who}_2 \end{array} \left[\begin{array}{c} \\ \text{Amy gave } t_1 \text{ to } t_2 \\ \\ \end{array}\right] \begin{array}{l}] \\ ]\,]\,]\,] \\ ]\,]\,]\,] \end{array}$$

(52c)
$$\begin{array}{l} [^{CP}\,[^{TP}\,\text{I}\,[^{VP}\,\text{wonder}\,[^{CP}\,[\text{how-much money}]_1 \\ \qquad\qquad\qquad\qquad\quad [^{CP} \\ [^{CP}\,[^{TP}\,\text{you}\,[^{VP}\,\text{know}\,[^{CP}\,\text{who}_2 \end{array} \left[\begin{array}{c} \\ \text{Amy gave } t_1 \text{ to } t_2 \\ \\ \end{array}\right] \begin{array}{l}]\,]\,]\,] \\ ] \\ ]\,]\,]\,] \end{array}$$

(52d)
$$\begin{array}{l} [^{CP}\,[^{TP}\,\text{I}\,[^{VP}\,\text{wonder}\,[^{CP}\,[\text{how-much money}]_1 \\ [^{CP}\,[^{TP}\,\text{you}\,[^{VP}\,\text{know}\,[^{CP}\,\text{who}_2 \\ \qquad\qquad\qquad\qquad\quad [^{CP} \end{array} \left[\begin{array}{c} \\ \text{Amy gave } t_1 \text{ to } t_2 \\ \\ \end{array}\right] \begin{array}{l}]\,]\,]\,] \\ ]\,]\,]\,] \\ ] \end{array}$$

(52e)
$$\begin{array}{l} [^{CP}\,[^{TP}\,\text{you}\,[^{VP}\,\text{know}\,[^{CP}\,[\text{how-much money}]_1 \\ \qquad\qquad\qquad\qquad\quad [^{CP} \\ [^{CP}\,[^{TP}\,\text{I}\,[^{VP}\,\text{wonder}\,[^{CP}\,\text{who}_2 \end{array} \left[\begin{array}{c} \\ \text{Amy gave } t_1 \text{ to } t_2 \\ \\ \end{array}\right] \begin{array}{l}]\,]\,]\,] \\ ] \\ ]\,]\,]\,] \end{array}$$

(52f)
$$\begin{array}{l} [^{CP}\,[^{TP}\,\text{you}\,[^{VP}\,\text{know}\,[^{CP}\,[\text{how-much money}]_1 \\ [^{CP}\,[^{TP}\,\text{I}\,[^{VP}\,\text{wonder}\,[^{CP}\,\text{who}_2 \\ \qquad\qquad\qquad\qquad\quad [^{CP} \end{array} \left[\begin{array}{c} \\ \text{Amy gave } t_1 \text{ to } t_2 \\ \\ \end{array}\right] \begin{array}{l}]\,]\,]\,] \\ ]\,]\,]\,] \\ ] \end{array}$$

Even more interesting is the fact that the six structures above do not even exhaust all the logical possibilities of a legitimate output for the same input. Regardless of which numeration is chosen for each round, some extra flexibility arises whenever there is more than one WH-phrase whose terminals are in the intersection of numerations. In such cases, there are other attested structures, where superiority effects appear not to hold. This will be the subject of the next section.

V.6. Hidden Superiority as Relativized Relativized Minimality

V.6.0. The Phenomenon

As shown in **§II.6**, syntactic amalgams appear not to exhibit superiority effects.

For instance, in (53), there are two WH-phrases (which would in principle be competing for the same position(s)), but the pattern that obtains resembles the one in (54) rather than (55), as if one of the WH-phrases were behaving like a pronoun for all intents and purposes.

(53) a: I'll find out [how much money]$_1$ Bob gave t_1 to you can imagine [who]
b: I'll find out [who]$_2$ Bob gave you can imagine [how much money] to t_2

(54) a. I'll find out [how much money]$_1$ Bob gave t_1 to [someone]$_2$
b. I'll find out [who]$_2$ Bob gave [some money]$_1$ to t_2

(55) a: I'll find out [how much money]$_1$ Bob gave t_1 to [whom]$_2$
b: * I'll find out [who]$_2$ Bob gave [how much money]$_1$ to t_2

Before getting into the details of how the system proposed here would handle the phenomenon, let us first establish a metalanguage that can help us state the problem more precisely. As an expository device, in **§V.1**, I will be talking about movement in terms of raising rather than lowering, and in terms of traces and/or copies rather than remerged elements. Later on, the partial conclusions will be formalized in **§V.2** in terms of the dynamic top-to-bottom derivational system that I advocate for in this dissertation.

V.6.1. The General Idea

As argued by Kitahara (1997), the so-called *Superiority Condition on Transformations* (Chomsky 1973) can be formalized in terms of Chomsky's (1995) *Minimal Link Condition* (MLC), itself a reformulation of Rizzi's (1990) *Relativized Minimality* (RM).[143] (56a) and (56b) are the outputs of two competing derivations, such that (56a) wins over (56b) because the chain links in (56a) – i.e. *how much money*$_1$ & *t*$_1$ – are closer to each other than are the chain links in (56b) – i.e. *who*$_2$ & *t*$_2$. Before any movement, *how much money* is closer to the target position (i.e. the embedded spec/CP) than *who* is. Thus, *who* cannot move there.

(56) a: I'll find out [how much money]$_1$ Bob gave t$_1$ to [whom]$_2$

b: * I'll find out [who]$_2$ Bob gave [how much money]$_1$ to t$_2$

[143]. Rizzi's (1990) original formulation of RM was not initially meant to handle Superiority. For present purposes, though, it's safe to take Superiority as an instance of RM, since the descriptive characterization of the former shares with the later the general idea of minimizing distance between chain links and precluding intervention; which, technicalities aside, is the essence of MLC (cf. Kitahara 1997). Alternatively, Chomsky (1981), Jaeggli (1981) and Aoun, Hornstein & Sportiche (1981) propose that Superiority reduces to the ECP applied at LF. May (1985) takes Superiority to be a subcase of Pesetsky's Path Containment Condition. Hornstein (1995, 2001), inspired by Chierchia's work, takes Superiority to be a subcase of Weak Cross Over. As far as I can see, all these takes on superiority are compatible with my present proposal that shared constituency may obfuscate superiority effects.

Such contrast does not exist in (54), repeated below as (57).

(57) a: I'll find out [how much money]$_1$ Bob gave t$_1$ to [someone]$_2$

b: I'll find out [who]$_2$ Bob gave [some money]$_1$ to t$_2$

In this case, the chain links in (57a) – i.e. *[how much money]$_1$* and *t$_1$* – are also closer to each other than are the chain links in (57b) – i.e. *[who]$_2$* and *t$_2$*. But that does not make (57b) ungrammatical. In fact, the derivations that yield (57a) and (57b) do not compete to begin with, since they start out from distinct numerations. Thus, the competitor to be compared with (57b) should be (58), which resembles (56a) with respect to the distance between chain links. As expected, (58) is not acceptable, as opposed to (56a).

(58) * I'll find out [some money]$_1$ Bob gave t$_1$ to [whom]$_2$

(57b) wins over (58) because, before any movement, it is irrelevant that *some money* is closer to the target position (i.e. the embedded spec/CP) than *whom* is, since the attracting head (i.e. the embedded C) attracts to its checking domain only phrases that can match its WH-feature. *Last Resort* rules out the movement of *some money* in (58), given that no feature gets checked that way. Therefore, *some money* is invisible to RM/MLC in (57b), which makes the chain links *who$_2$* and *t$_2$* as close as possible in the technical sense. This is the essence of RM:

phrases that are close to the attracting head should block the movement of distant phrases if and only if they are all of the same kind. *Closeness* and *Minimality* of chain-links are calculated *relatively* to the features of the attractor and the ones of the potential moving phrases.[144]

In (56a) as well as in (56b), both *how much money* and *who(m)* are in the c-command path of the embedded C and both can check its [+WH] feature; hence they are both attractable. Since *how much money* is closer to the embedded C than *who* is, then (56a) is grammatical but (56b) is not.

The contrast described above is not observed in cases where one of the WHs is part of a syntactic amalgam.[145] Notice that (59) patterns like (57) rather than like (56), even though both sentences in (59) have two WHs apparently under the scope of the attracting C, similarly to (55). By the MLC, *how much money* should count as the closest WH to the embedded C (thus blocking the movement of *who* in (59b)), but it behaves as a non-WH version of itself (i.e. *some money*) for the purposes of moving *who* in (59b), which, unlike (56b), is just as acceptable as (57b).[146]

[144]. Technically speaking, α can move to the checking domain of a head β iff α is attractable by β; and there is no γ which is also attractable by β, γ being closer to β than α is. We say that α is attractable by β iff α has the relevant kind of feature that could potentially check a feature of β; and β c-commands α. We say that γ is closer to β than α is if and only if β c-commands both α and γ, and moreover γ c-commands α.

[145]. Standard examples of Superiority typically involve competition between a subject and an object (e.g. *who bought what*?/**what did who buy?*) rather than two objects, which may raise further issues regarding equidistance. All my examples involve double objects because syntactic amalgamation cannot affect *bona fide* subjects to begin with (e.g. **I wonder what*$_1$ *you can imagine who bought t*$_1$) (cf. Guimarães 2003a/b/c).

[146]. Some English speakers judge (59b) as somewhat degraded in comparison to (59a). As suggested to me by Howard Lasnik (personal communication), this

(59) a: I'll find out [how much money]$_1$ Bob gave t$_1$ to you can imagine [who]

b: I'll find out [who]$_2$ Bob gave you can imagine [how much money] to t$_2$

For all intents and purposes, a WH that is amalgamated with syntactic chunks of a certain kind (e.g. *you'll never guess*, *God knows*, *you can imagine*, etc) does not count as a WH. Further evidence for this is shown in (60) and (61).

(60) a: Amy wonders [how much money]$_1$ Bob gave t$_1$ to Tom.

may be due to parsing difficulties associated with a highly complex material intervening between the WH and the stranded preposition that selects it; as independently attested in cases like *"who did you give that Beatles record autographed by George Harrison that you got in London last year to?"* which is less acceptable than *"who did you give that Beatles record to?"*. Crucially, even for those speakers, (59b) is much more acceptable than (56b), which is just plain impossible. Interestingly, such degrading effect does not exist at all in Romance (exemplified below with Portuguese), where the analogues of (59a) and (59b) are both equally acceptable, which is consistent with the reasoning just sketched, since WH-movement must involve pied-piping in Romance (cf. §II.8; §V.4).

(i) Eu vou descobrir quanto dinheiro Bob deu você pode imaginar pra quem.

I will discover how-much money Bob gave you can imagine to who.

(ii) Eu vou descobrir pra quem Bob deu você pode imaginar quanto dinheiro.

I will discover to who Bob gave you can imagine how-much money.

b: * Amy wonders God knows [how much money]$_1$ Bob gave t$_1$ to Tom.

c: * Amy wonders [some money]$_1$ Bob gave t$_1$ to Tom.

(61) a: * Amy believes Bob gave [how much money] to Tom.

b: Amy believes Bob gave God knows [how much money] to Tom.

c: Amy believes Bob gave [some money] to Tom.

One could deny that this is a real problem under the assumption that *how much money* in (59b) is deeply embedded inside a complex constituent also containing the parenthetic-like string, as the brackets in (62b) indicate. That way, *how much money* would not c-command *who*, hence not counting as an intervener according to the MLC (Kitahara 1997; Uriagereka 1999).

(62) a: I'll find out **[how much money]$_1$** Bob gave **t$_1$** to *[you can imagine* ***who]***

b: I'll find out **who$_2$** Bob gave *[you can imagine* ***how much money]*** to **t$_2$**

But what kind of complex constituent would that be? For (62b), one could speculate that *you can imagine how much* is some sort of complex modifier whose sister is *money*; or even that *you can imagine how* is the sister of *much money*. But that reasoning would not carry over to *you can imagine who* in (62a) since there is no NP which that would the modifier of. In face of that, one might take *you can imagine* to be a constituent that takes *who* (62a) or *how much money* (62b) as its sister. That poses the problem of having an unsaturated verb (*imagine*) inside the modifier, or having to stipulate an *ad hoc* empty category there, not to mention the mysterious nature of that kind modification, not found anywhere other than amalgams.[147]

Now, let us see how the problem can be once we assume that syntactic amalgams involve multiply-rooted phrase markers. Let us focus on the problematic case (53b), repeated below as (63).

(63) I'll find out [who]$_2$ Bob gave you can imagine [how much money] to t$_2$

The key property of this construction is the fact that it conveys two independent parallel messages (cf. **§II.4**). In (63), what is being imagined is not just the size of amount of x money, but the size of amount of x money such that Bob gave x to a person y. But (63) cannot be just a convoluted version of (64a), as it also includes another chunk of structure (i.e. *I'll find out...*) to which *[who$_2$ Bob gave [how much money]$_1$ to t$_2$]* is subordinated, as an indirect question about the identity of that person y that was given an amount x of money by Bob. So, besides (64a) there is also (64b).

(64) a: You can imagine [^{CP} [how much money]$_1$ [^{IP} Bob gave t$_1$ to whom]]

b: I'll find out [who]$_2$ Bob gave [how much money]$_1$ to t$_2$

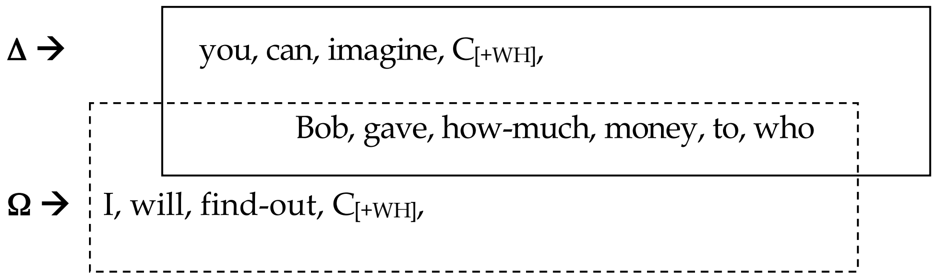
It seems, thus, that (64a) and (64b) somehow collapse at the paratactic level, yielding (63). Intriguingly, (64b) is a legitimate structure as part of this

[147]. Also notice that those parenthetic-like string always contain verbs that (under the relevant reading) select only CPs as their complements, rather than pure DPs. For instance, *"Homer drank I wonder how many beers at the party"* is possible, but *"I wonder 75 beers"* and *"How many beers do I wonder?"* are not.

more complex paratactic construction, even though it is ungrammatical when in isolation – as in (55b) – due to a violation of the MLC. Therefore, the problem with the absence of contrast in (59) is real. In what follows, I propose that (59b) indeed obeys the MLC at the relevant derivational step, but the complex interaction of parallel structures masks superiority effects. I further claim that this complex interaction is not paratactic, but syntactic.

Following the proposal made in **§IV.1**, let us take the input to the syntactic computations that generate (53a) and (53b) would be as in the Venn diagram in (65), irrelevant functional elements omitted.

(65)


Δ →
you, can, imagine, $C_{[+WH]}$,
Bob, gave, how-much, money, to, who
Ω →
I, will, find-out, $C_{[+WH]}$,

Such intersections allow local computations to interfere with one another to some extent, with paratactic effects emerging from syntax pushed to limit.

We have been assuming that syntactic representations may exhibit multiply-rooted phrase markers with parallel trees that share some constituent(s) somewhere in between the roots and the terminals (*via* multi-motherhood). From that perspective, the actual structure of (63) would be (66), which – linearization matters aside – involves two parallel matrix clauses (i.e. *you can imagine...* and *I'll* *find out...*) sharing the same subordinate IP (i.e. *Bob gave [how much money] to [whom]*).

(66) [^{CP} I'll find out [^{CP} [who]$_2$ C [^{IP}]]

Bob gave t$_1$ to t$_2$

[^{CP} you can imagine [^{CP} [how much money]$_1$ C]]

The derivation of (66) starts with the computational system randomly selecting the numeration **Δ** as the array of lexical tokens to be syntactically integrated first. Right after the embedded IP is built, (67) obtains.

(67) [^{IP} Bob gave [how much money]$_1$ to [who]$_2$]

This IP is then embedded inside a CP that will eventually be a sentential complement inside the matrix clause that corresponds to numeration **Δ**.

(68) [^{CP} C$_{[+WH]}$ [^{IP} Bob gave [how much money]$_1$ to [who]$_2$]]

At this point, C attracts the closest WH under its scope (i.e. *how much money*) in accordance with the MLC, as in (69).

(69) [^{CP} [how much money]$_1$ $C_{\cancel{[+WH]}}$ [^{IP} Bob gave t$_1$ to [who]$_2$]]

The derivation proceeds in the usual fashion, and the matrix clause corresponding to numeration **Δ** is eventually built, as in (70).

(70) [$^{\text{CP-}\Delta}$ you can imagine [$^{\text{CP}}$ [how much money]$_1$ C [$^{\text{IP}}$ Bob gave t$_1$ to [who]$_2$]]]

Once the (sub)derivation corresponding to numeration **Δ** is over, then the entire remnant IP (from which *how much money* has been moved) is taken and incorporated into the derivation corresponding to numeration **Ω**, being remerged with another C, which cannot detect the (now-moved) WH *how much money* under its scope at that derivational point, as in (71).

(71) [$^{\text{CP}}$ C$_{[+\text{WH}]}$ [$^{\text{IP}}$]]
Bob gave t$_1$ to [who]$_2$
[$^{\text{CP-}\Delta}$ you can imagine [$^{\text{CP}}$ [how much money]$_1$ C]]

What actually happens in (71) is that the very same token of the IP *[Bob gave t$_1$ to who]* remains as a daughter of the embedded CP (and as a sister of the embedded C) under the root **CP-Δ** while being 'remerged' with another element from a parallel (sub)derivation (i.e. the C taken from numeration **Ω** in (65), which will eventually be the complementizer of the embedded clause under the other matrix clause (root **CP-Ω**) of the complex structure).[148] Thus,

[148]. This instance of shared constituency follows from derivational economy. When the computational system is done with the (sub)derivation that syntactically integrates the members

[^{IP} Bob gave t_1 to who] becomes a shared constituent, having two mothers in a complex multiply-rooted phrase marker.

Once *[^{IP} Bob gave t_1 to who]* gets remerged with another C in a parallel (sub)derivation, then *who* becomes the closest (and the only) attractable WH under the scope of that new attracting C. Then it moves to the spec/CP within the derivation that integrates the lexical tokens in **Ω**, in accordance with the MLC. This is the crucial step that explains why (53b) is possible.

(72) [^{CP} [who]$_2$ C [^{IP}]]

Bob gave t_1 to t_2

[$^{CP\text{-}\Delta}$ you can imagine [^{CP} [how much money]$_1$ C]]

The derivation proceeds in the usual fashion, and the matrix clause corresponding to numeration **Ω** is eventually built, as shown in (73).

(73) [$^{CP\text{-}\Omega}$ I'll find out [^{CP} [who]$_2$ C [^{IP}]]

Bob gave t_1 to t_2

[$^{CP\text{-}\Delta}$ you can imagine [^{CP} [how much money]$_1$ C]]

of set **Δ** and starts integrating the members of **Ω**, it identifies the intersection **Δ∩Ω**, given that some of the members of **Ω** are already in the derivational workspace, as the leaves of a (sub)tree. Then, why would the system select those same lexical tokens again, one by one, and build an identical clone of that same IP? It is more economical to just take that IP already built and incorporate it into the new (sub)derivation. It is a matter of choosing between one application of merge and many applications of select, merge, and move (copy, merge, delete).

Thus, the whole complex paratactic-like construction in (53b) is built with purely syntactic tools. Notice that there are two WH-chains in (73). Chain #1 is headed by *how much money* under root **CP-Δ**, and chain#2 is headed by *who* under root **CP-Ω**. The tails of both chains are inside an IP that is shared by the two roots. From a representational perspective, chain#2 seems to violate the MLC, since between its links there is the tail of chain#1. From a derivational perspective, though, both chains obey the MLC. In a nutshell, RM should be calculated relatively to each derivational domain and each derivational step. I call this *Relativized Relativized Minimality*.

Before moving on to other examples, let me clear up a very important issue that was overlooked in the exposition above. On the one hand, the rationale behind the idea of getting shared constituency through remerge in the derivational step in (71) is that this is the optimal way to make **all** lexical tokens in **Δ∩Ω** access **both** parallel (sub)derivations, therefore being integrated into **both** parallel (sub)representations. On the other hand, it is crucial that in step (72) the higher WH (i.e. *how much money*) be absent from the shared IP, which contains only a trace of it. Therefore, for all intents and purposes **none** of the terminals that constitute *how much money* is there when the embedded IP gets remerged and accesses the derivation.

The conflict between these two assumptions is obvious. If we take *t_1* in (71) to be a GB-style trace with no internal content (other than a category label and an index), then its not surprising that *who* should move in step (72) without violating the MLC; but this also entails that the system fails to map all items of **Ω** onto the corresponding phrase marker, since there would be no ‘occurrence’ of the tokens *how-much* and *money* anywhere under root **CP-Ω**. Conversely, if we take t_1 to be a minimalist-style copy of [^{DP} *how much money*] in the spec/CP under root **CP-Δ**, then it is obvious why *how-much* and *money* are entering the derivation that syntactically integrates the items in **Ω**; but it is mysterious, then, why this copy inside the shared IP does not block the movement of *who*, according to the MLC.

This problem goes away if we endorse the following two assumptions.

(74) Technical questions arise about the identity of α and its trace t(α) after a feature of a has been checked. The simplest assumption is that the features of a chain are considered a unit: if one is affected by an operation, all are. (Chomsky 1995: chapter 4, note 12)[149/150]

[149] This is equivalent to Hornstein’s (1995) *All for One Principle*, which states that *“Every link in a chain meets the morphological conditions satisfied by any link in a chain”*. Chomsky (2001) incorporates this idea into a system where Move is seen as Agree + Pied-Piping + Merge; where Pied-Piping requires phonological content.

[150] As it will be shown in the next section, this assumption can be derived as theorem if we take that movement, too, involves remerge/multi-motherhood as part of its inner-workings, rather than copy + merge (+ delete). That way, a so-called moved phrase is better understood as a pluripresent phrase, simultaneously occupying the head and the tail positions of a chain (cf. Bobaljik 1995; Drury 1998, 1999; Epstein, Groat, Kitahara & Kawashima 1998; Guimarães 1999, 2002, 2003b/c/d; Abels 2001; Gärtner 2002). If any feature of that single entity gets deleted, obviously the whole chain gets affected. It is this approach to movement that I am tacitly assuming here, although I keep using the copy-theoretical terminology and the trace-theoretical notation for expository reasons.

(75) [T]he *wh*-phrase has en uninterpretable feature [wh] and an interpretable feature [Q], which matches the uninterpretable probe [Q] of a complementizer in the final stage. (...) The *wh*-phrase is active until [wh] is checked and deleted. (Chomsky 2000: 128)[151]

Therefore, *how much money* indeed is inside the shared IP, c-commanding *who*. But, after having its [wh] feature checked in step (69), it becomes inactive. Hence, the MLC demands that *who* be attracted in step (72).

Although, by this relativized version of RM, *how much money* must be the first WH to move (since it is closer to either attractor), nothing forces it to move to the spec/CP under *imagine*. Alternatively, the MLC can be satisfied by the movement of *how much money* to the spec/CP under *find-out*; which causes *who* to further move to the spec/CP under *imagine*. This is exactly how (53a) is generated. The starting point is also be intersecting numerations in (65). Remember that the choice of which numeration to start with is random. So, if **Ω** is chosen, *how much money* moves in that first subcomputation, whose final output is (76). Then, the embedded IP gets remerged with the C from **Δ**, as in (77), which attracts *who*, as in (78), eventually yielding (79).

(76) [$^{\mathrm{CP}\text{-}\Omega}$ I'll find out [$^{\mathrm{CP}}$ [how much money]$_1$ C [$^{\mathrm{IP}}$ Bob gave t$_1$ to [who]$_2$]]]

[151] But see Guimarães (2003b/c) on successive cyclicity and defective intervention.

(77) [^{CP} C$_{[+WH]}$]]
Bob gave t$_1$ to [who]$_2$
[$^{CP\text{-}\Delta}$ you can imagine [^{CP} [how much money]$_1$ C]]

(78) [^{CP} [who]$_2$ C ^{IP}]]
Bob gave t$_1$ to t$_2$
[$^{CP\text{-}\Delta}$ you can imagine [^{CP} [how much money]$_1$ **C**]]

(79) [$^{CP\text{-}\Omega}$ I'll find out [^{CP} [who]$_2$ C ^{IP}]]
Bob gave t$_1$ to t$_2$
[$^{CP\text{-}\Delta}$ you can imagine [^{CP} [how much money]$_1$ C]]

This competing derivation is as economical as the one in (67-73), and it also produces a convergent representation. Therefore, such representation should be grammatical too; which indeed it is. The corresponding meanings for (53a)/(79) and (53b)/(73) would roughly be (80a) and (80b) respectively.[152]

(80) a: $\exists \boldsymbol{x}$, $\exists \boldsymbol{y}$ [[Bob gave an amount $\boldsymbol{x}$ of money to a person $\boldsymbol{y}$] & [you can imagine what the size of $\boldsymbol{x}$ is] & [I'll find out what the identity of $\boldsymbol{y}$ is]]

[152] This is obviously a rough oversimplification. See Guimarães (2003e) for details, especially with regards to how both variables $\boldsymbol{x}$ and $\boldsymbol{y}$ (the WH-traces) get bound in the same domain despite the absence of a single root in the syntactic representation.

b: $\exists x$, $\exists y$ [[Bob gave an amount x of money to a person y] & [you can imagine what the identity of y is] & [I'll find out what the size of x is]]

If complex syntactic amalgams indeed have the structure proposed above, it not obvious how they get mapped into a linear PF-string. Aside from the shared material, multiply-rooted phrase markers necessarily contain terminals dominated only by one of the roots, which do not stand in any relation to the terminals dominated only by another root. Whatever the linearization function is (e.g. Kayne's (1994) LCA, the head parameter, etc), it cannot establish precedence relations among all terminals in any deterministic way.[153]

V.6.2. 'Hidden Superiority' in Top-to-Bottom Derivations

Having presented the idea of superiority effects being masked by derivational 'circumstances', let' us now see how that applies to the specific 'generalized tucking-in' derivational system being proposed here.

Just like in the bottom-up implementation of above, we need to commit to the assumption in (81 (=75)).

[153] See Wilder (1999), Gärtner (2002) and Citko (2002) for related discussion.

(81) [T]he *wh*-phrase has en uninterpretable feature [wh] and an interpretable feature [Q], which matches the uninterpretable probe [Q] of a complementizer in the final stage. (...) The *wh*-phrase is active until [wh] is checked and deleted. (Chomsky 2000: 128)

In the sample derivations below the following notational convention will be adopted. Whenever a phrase bears an unchecked [wh] feature, it will be marked with a flag symbol (i.e. ⚐), while the checkmark symbol (i.e. ☑) will be used for phrases whose [wh] feature has been checked.

As for the Minimal Link Condition, nothing substantial needs to be changed. However, minor technical details must be redefined in accordance with the top-to-bottom mechanics of 'generalized tucking-in' derivations. This can be done in many ways. For the present purposes, I will not bother with details, and will simply follow the informal definition in (82), which captures the general idea.

(82) A new active WH-phrase Z cannot be merged anywhere in the c-command path of the highest WH-phrase X before X lowers (i.e. remerges) into its case/theta position(s).

Let us begin with the non-controversial case, repeated below in (83), whose corresponding input would be as in (84 (=65)).

(83) I'll find out how much money Bob gave to you can imagine who

(84)

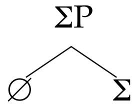
Δ → you, can, imagine, C$_{[+WH]}$,

Bob, gave, how-much, money, to, who

Ω → I, will, find-out, C$_{[+WH]}$,

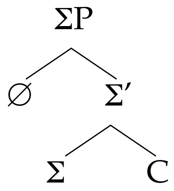
First, the system randomly takes Ω as the starting point for the first derivational round. The construction of the master clause begins in the usual fashion, with the starting axiom, as in (85a), and subsequent tucking-in of lexical tokens in Ω, one at a time, as follows.

(85) a:



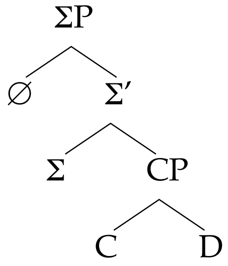
(starting axiom)

b:



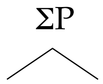
(merge **C̲**)

c:

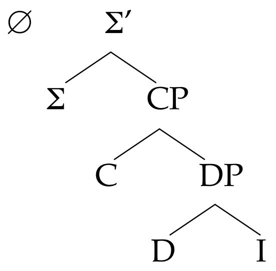


(merge **D̲**)

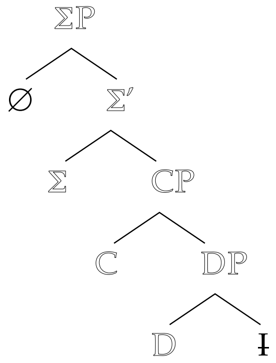
d:



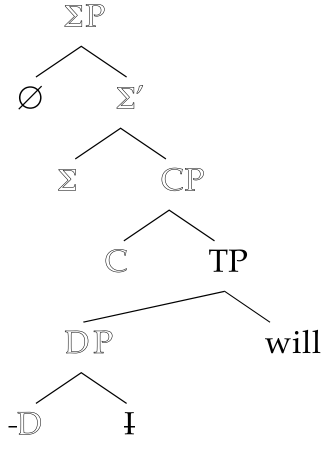
(merge **I̲**)



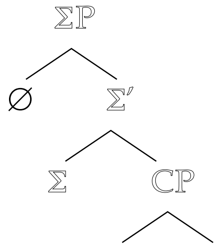
e: (spell-out)

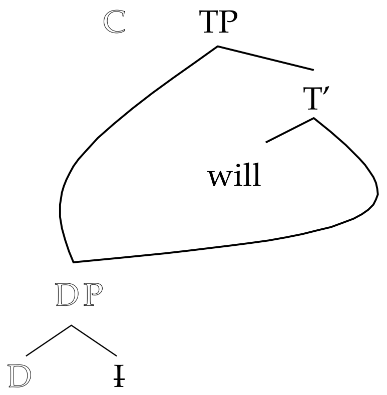


f: (merge **will**)

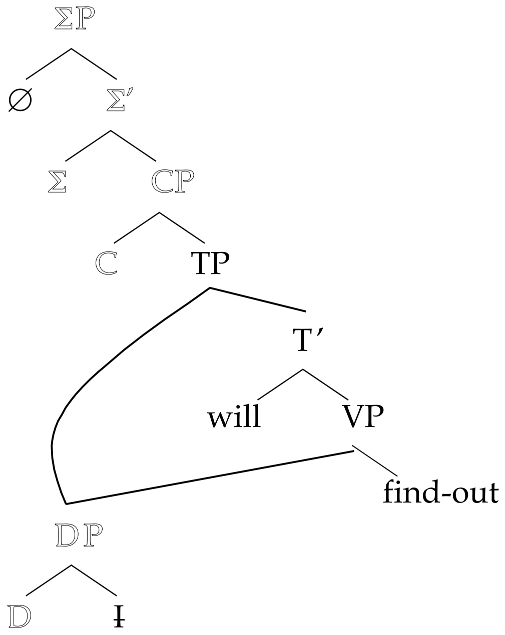


g: ((re)merge ***[^{DP} D I]***)

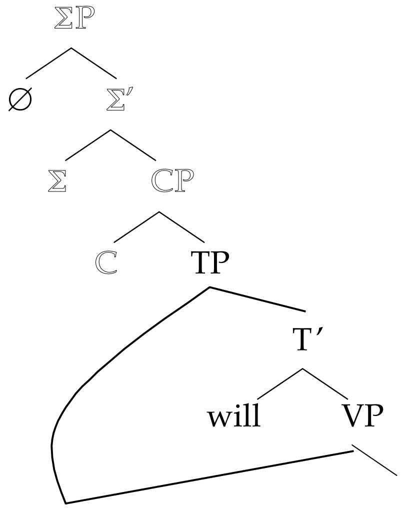


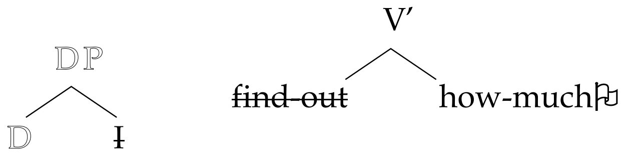


h: (merge **find-out**)

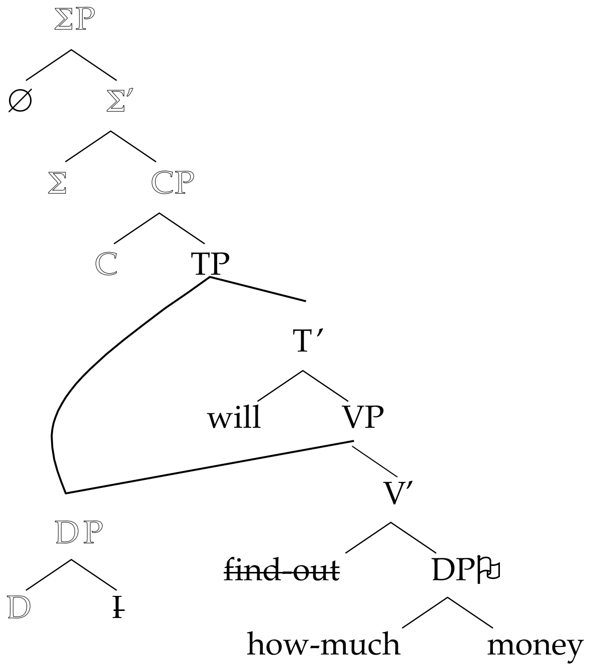


i: (merge **how-much**)

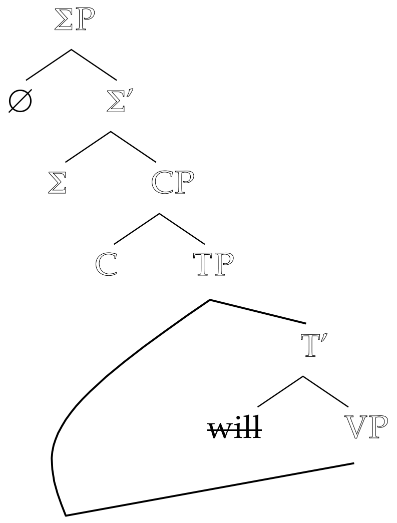


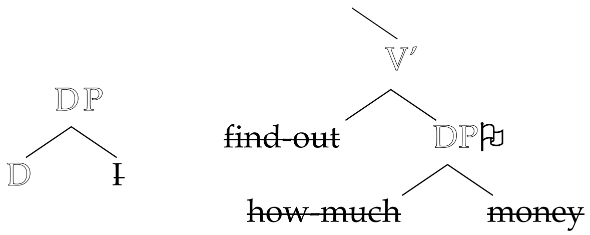


j: (merge **money**)



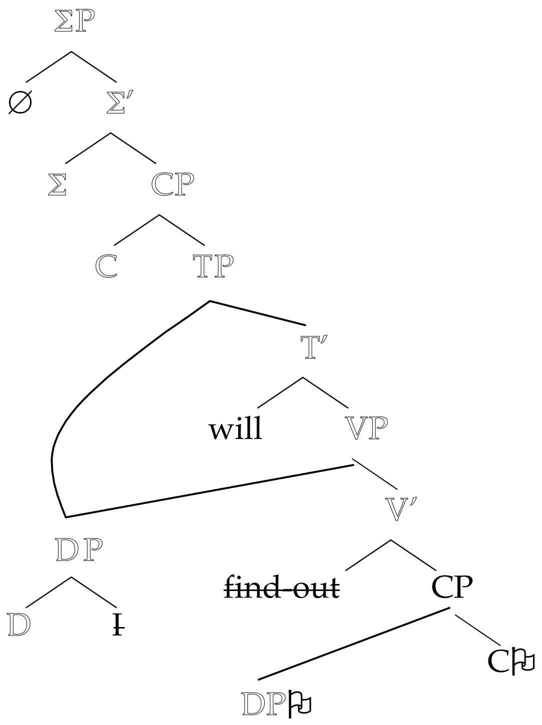
k: (spell-out)

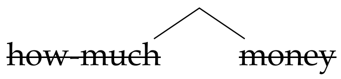




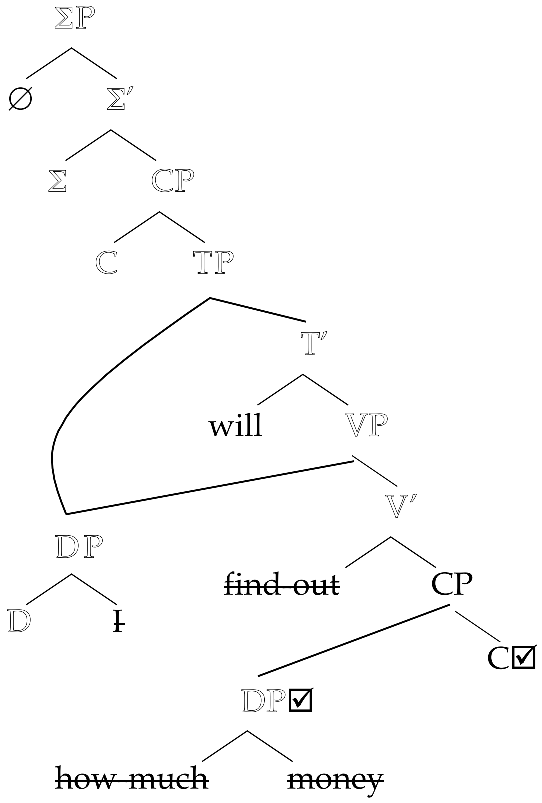
So far, the corresponding PF (super-)string is **[#I#]$^\cap$[#will#$^\cap$#find-out#$^\cap$#how-much#$^\cap$#money#]**. Notice that, at this point, the WH-phrase how much money is still active, in the sense of (81) above, and it needs to have its uninterpretable [wh] feature checked under sisterhood, which is done in the following steps, as shown below.

(85) l: (merge **<u>C</u>**)





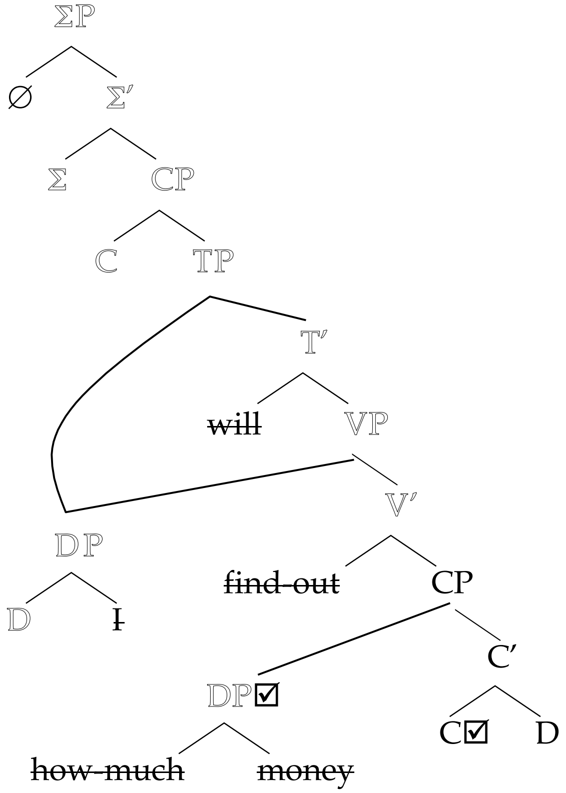
m:



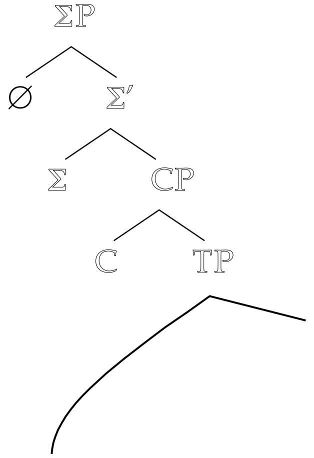
(feature checking)

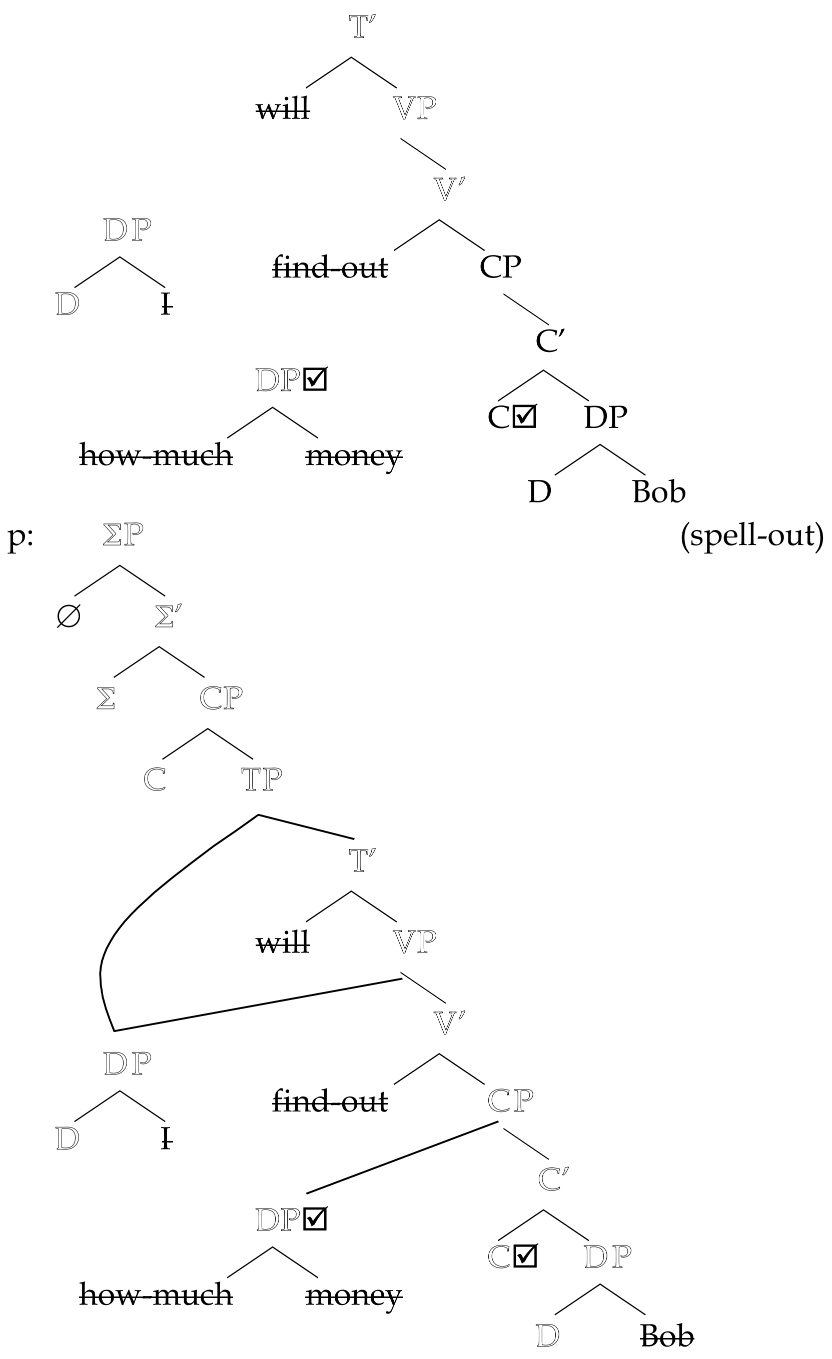
After that, the first derivational round proceeds in the usual fashion, as follows.

(85) n: (merge **D**)

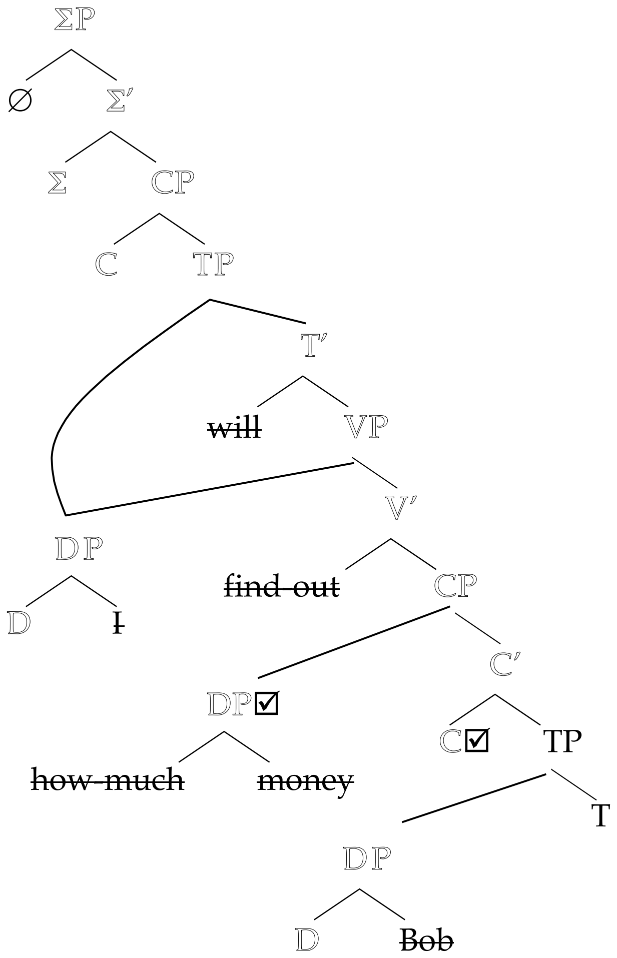


o: (merge **Bob**)



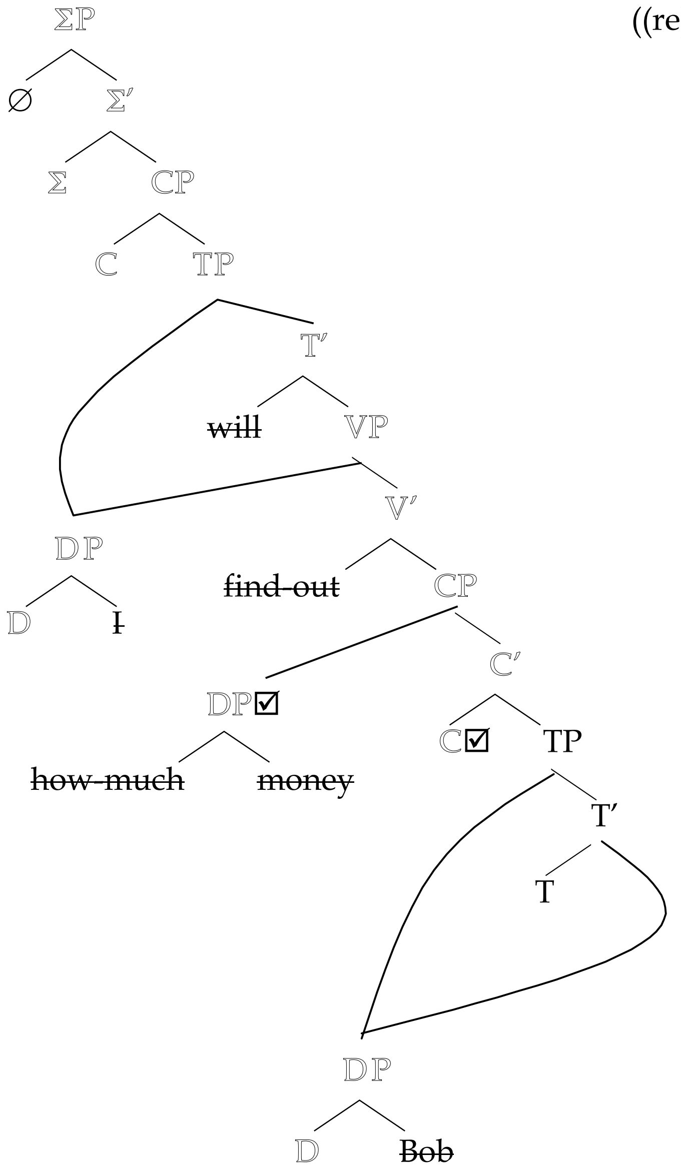

T′
~~will~~
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
DP☑
C☑
DP
~~how-much~~
~~money~~
D
Bob
p:
ΣP
(spell-out)
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
~~find-out~~
CP
D
~~I~~
C′
DP☑
C☑
DP
~~how-much~~
~~money~~
D
~~Bob~~

q:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
DP☑
~~how-much~~
~~money~~
C☑
TP
T
DP
D
~~Bob~~

(merge **<u>T</u>**)

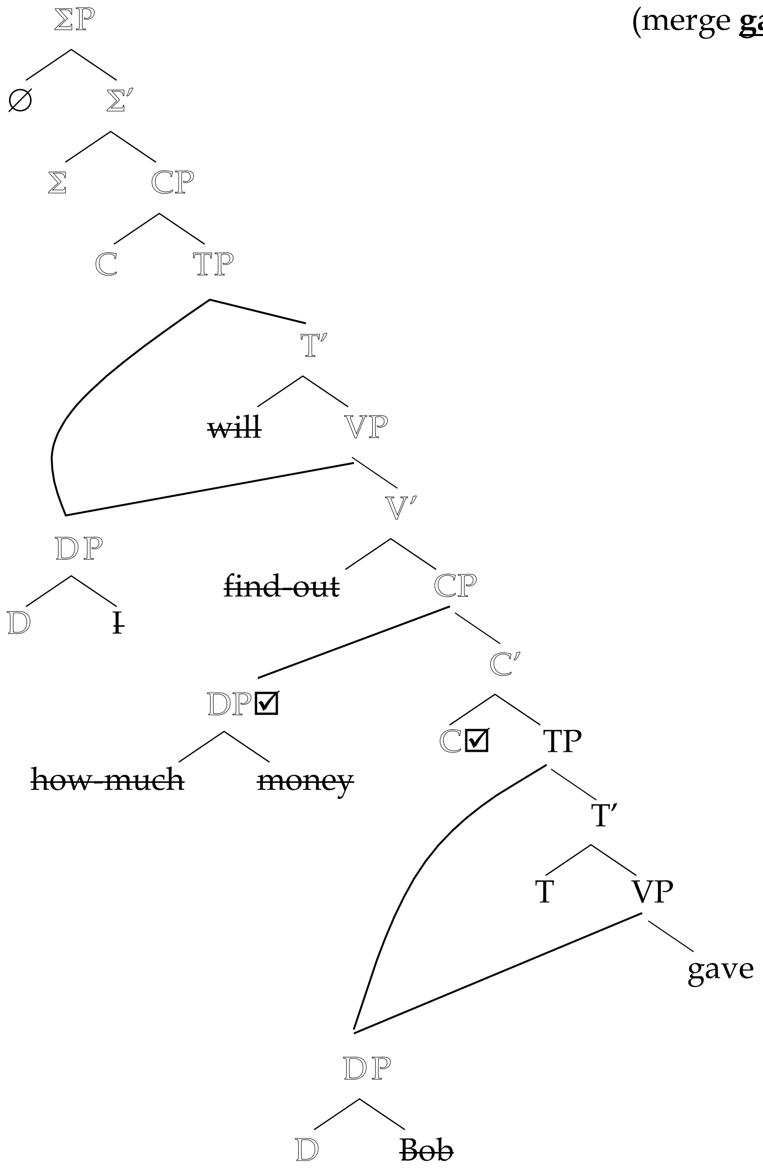
r:



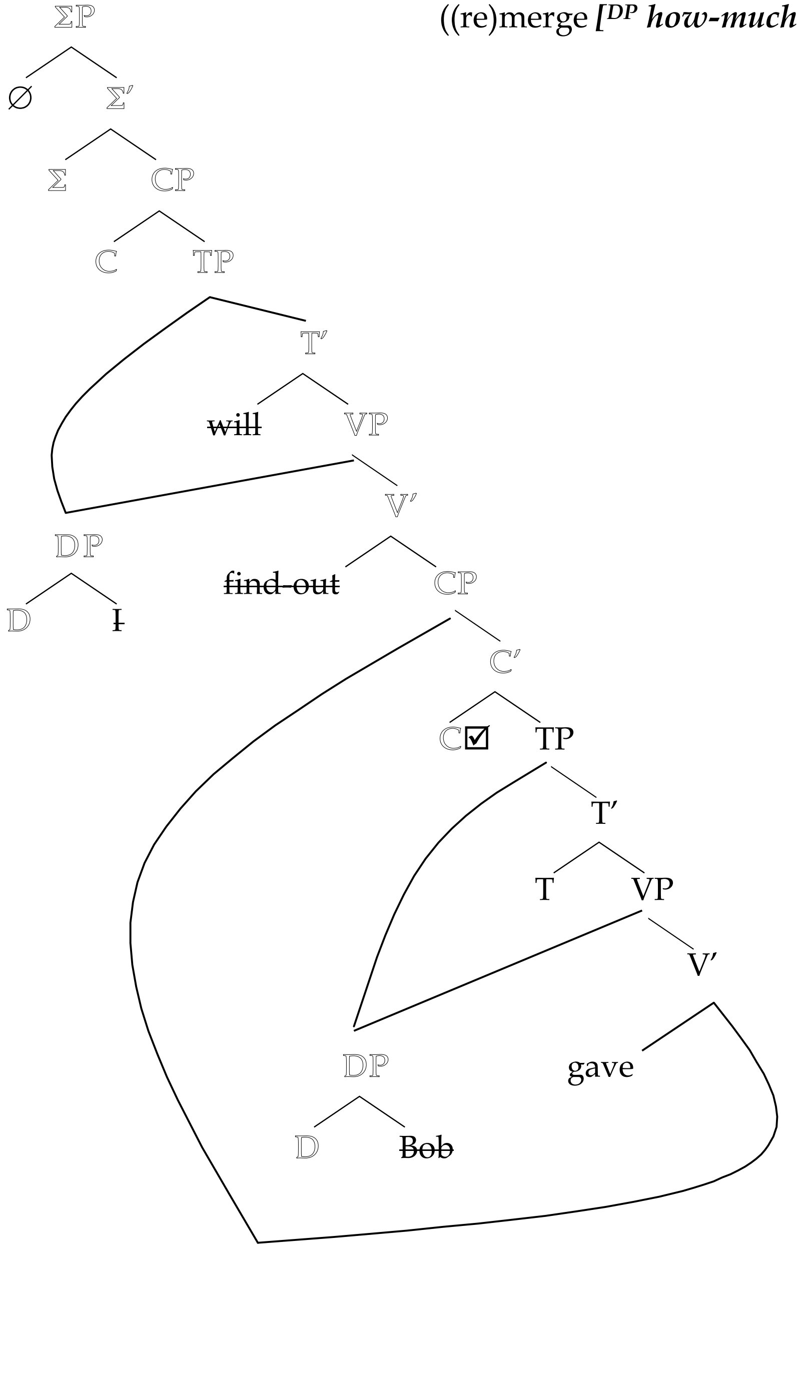
((re)merge ***[^{DP} D Bob]***)

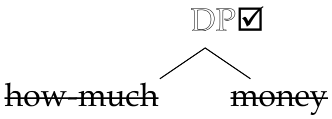
s:

(merge **gave**)

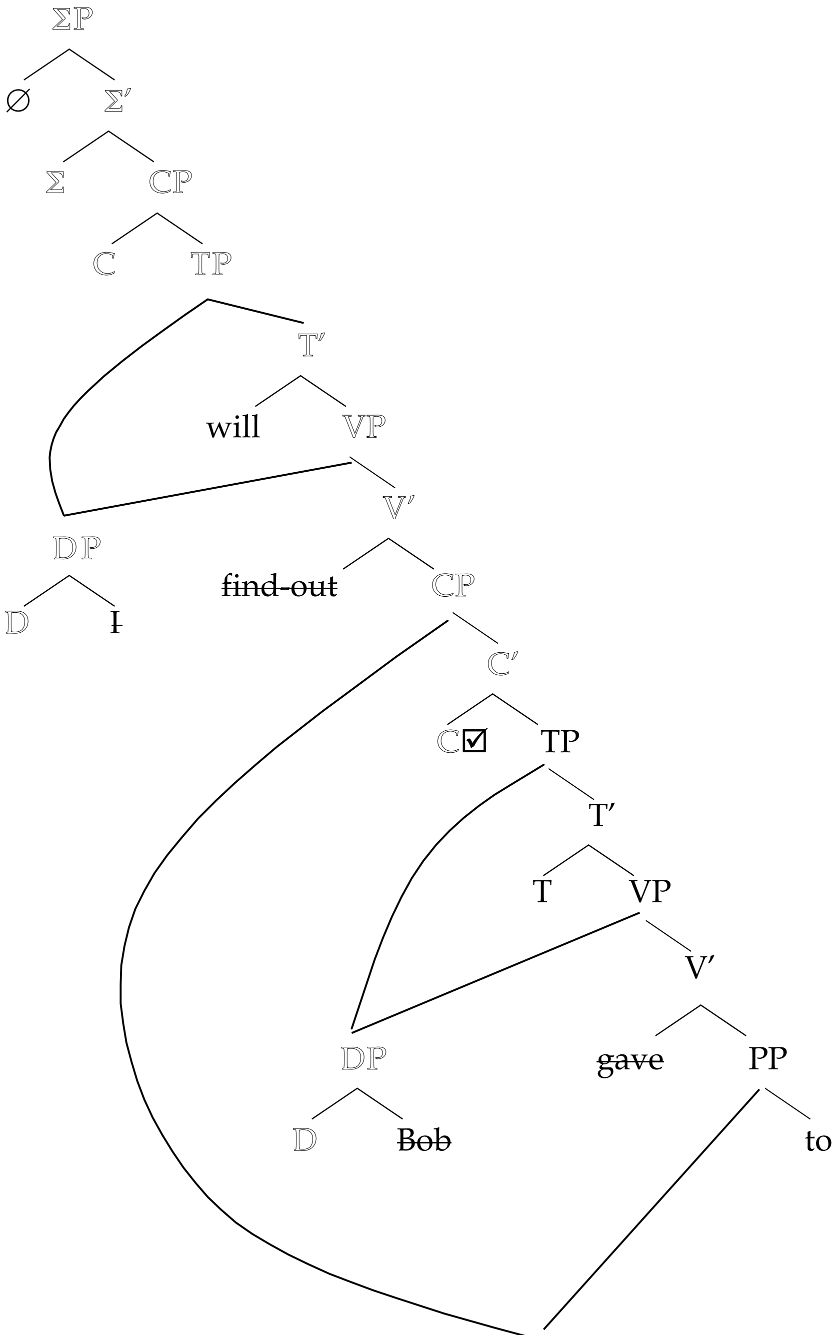

ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
DP☑
~~how-much~~
~~money~~
C☑
TP
T′
T
VP
gave
DP
D
~~Bob~~

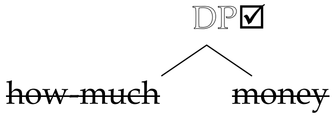
t: ((re)merge ***[^{DP} how-much money]***)




DP☑
~~how-much~~
~~money~~

u:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
will
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
C☑
TP
T′
T
VP
V′
DP
D
~~Bob~~
~~gave~~
PP
to



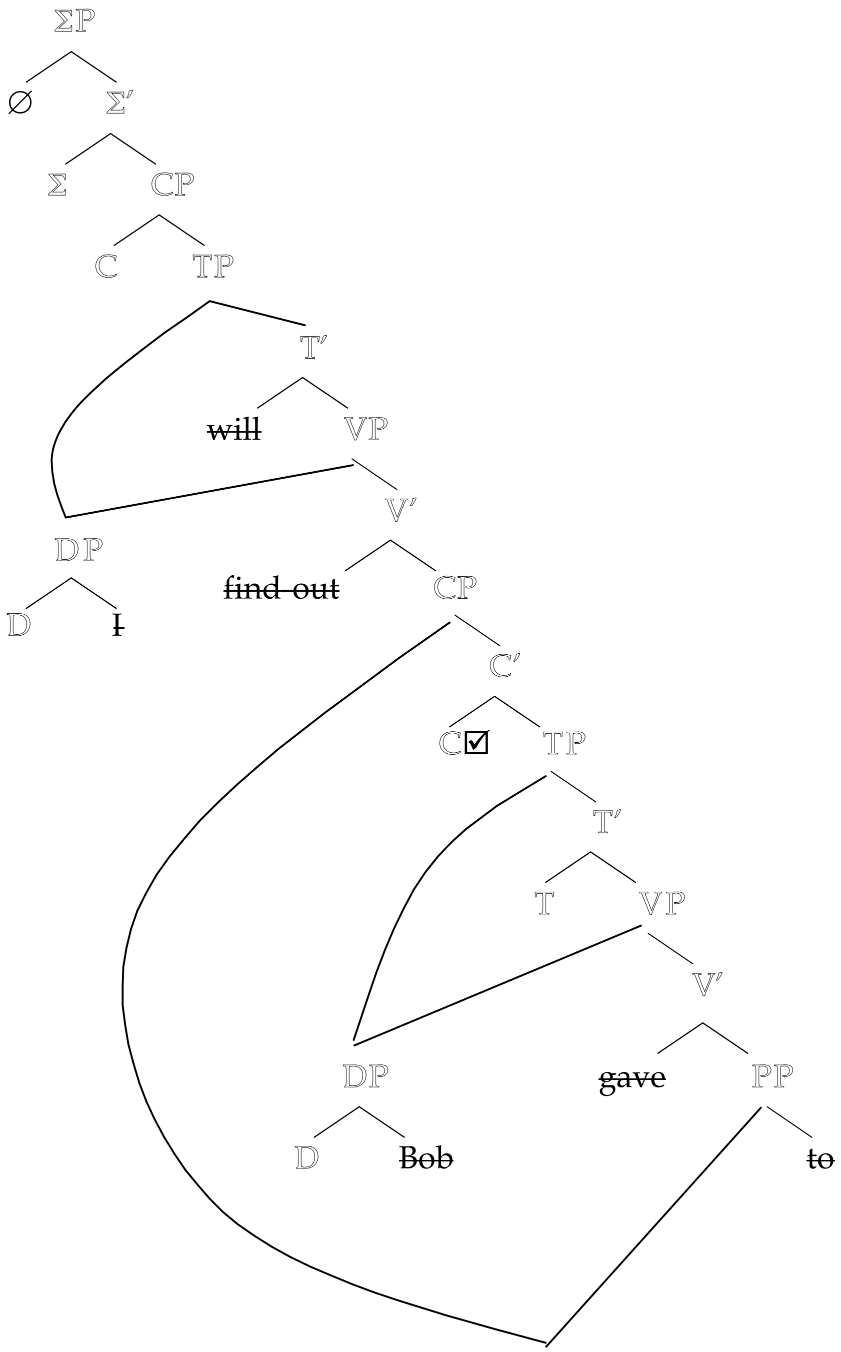
At this point, the natural continuation towards building the sentence corresponding to Ω would be to build the WH-phrase **who**, by tucking it in as the complement of **to**.

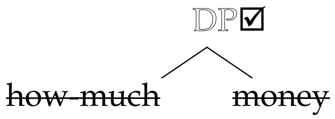
However, if that happens, the **who** will not be able to be later merged in the lower spec/CP of the subservient clause, since it will fail to c-command that [+WH] complementizer, as discussed previously for many other derivations along this chapter.

The only alternative that could lead to convergence, then, is the termination of the first derivational round, leaving the phrase marker corresponding to Ω as an incomplete structure. This is possible because all relevant lexical tokens necessary to build this chunk of the structure are shared by the numeration in Δ. That way, the subcomputation performed in the second derivational round can finish the job of building that chunk of structure left incomplete by in the first derivational round.

Therefore, for convergence reasons, the next step after immediately after (85u) must be the one in (85v), where the current structure undergoes spell-out.

(85) v:

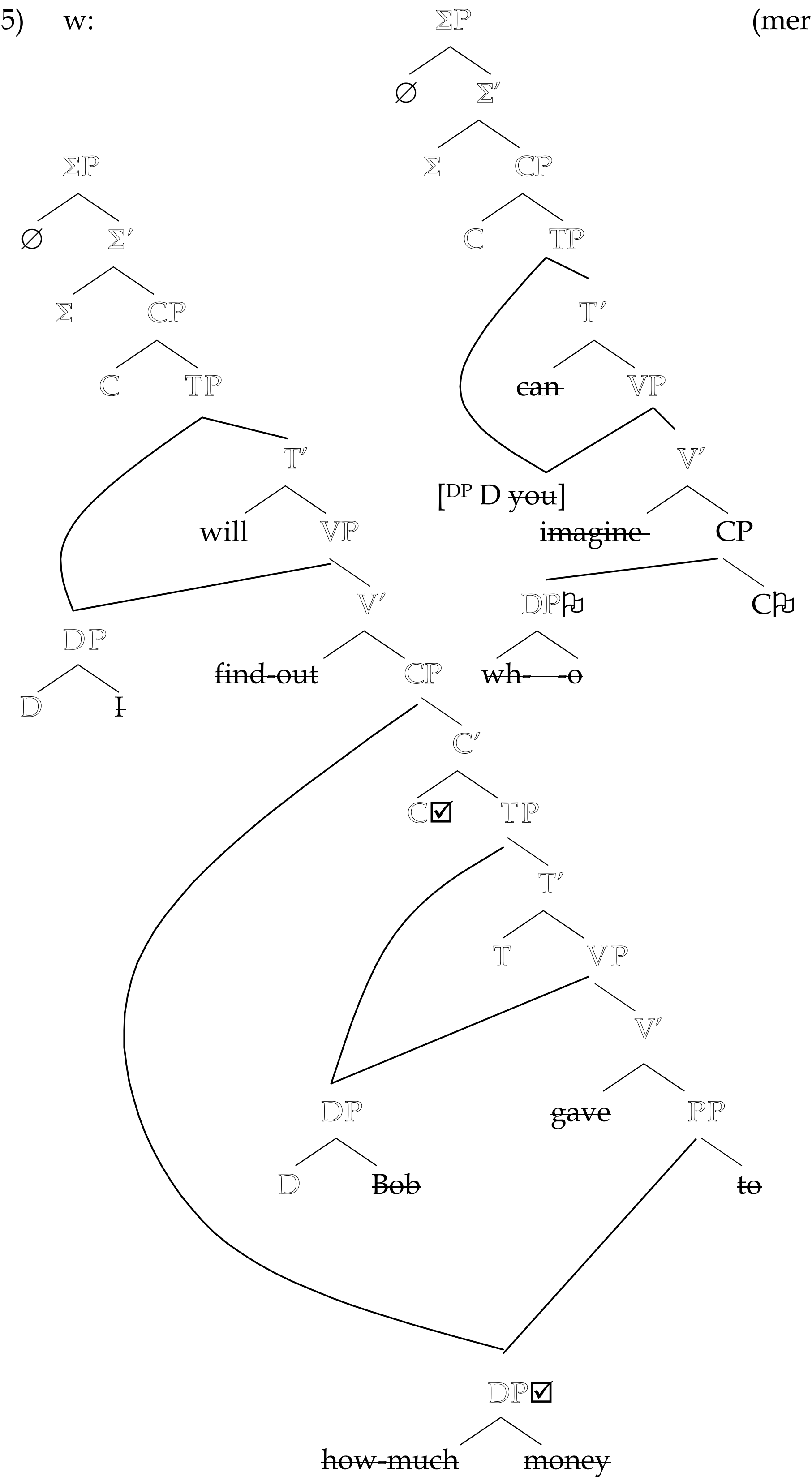

ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
C☑
TP
T′
T
VP
V′
DP
D
~~Bob~~
~~gave~~
PP
~~to~~



This marks the end of the first derivational flow. So far, the corresponding PF (super-)string is **[#I#]$^{\cap}$[#will#$^{\cap}$#find-out#$^{\cap}$#how-much#$^{\cap}$#money#]$^{\cap}$[#to#]**

The computational system, then, shifts its attention to numeration Δ, and starts the second derivational flow, which proceeds in the usual fashion, from the top downwards, up to the point where the structure in (85w) obtains.

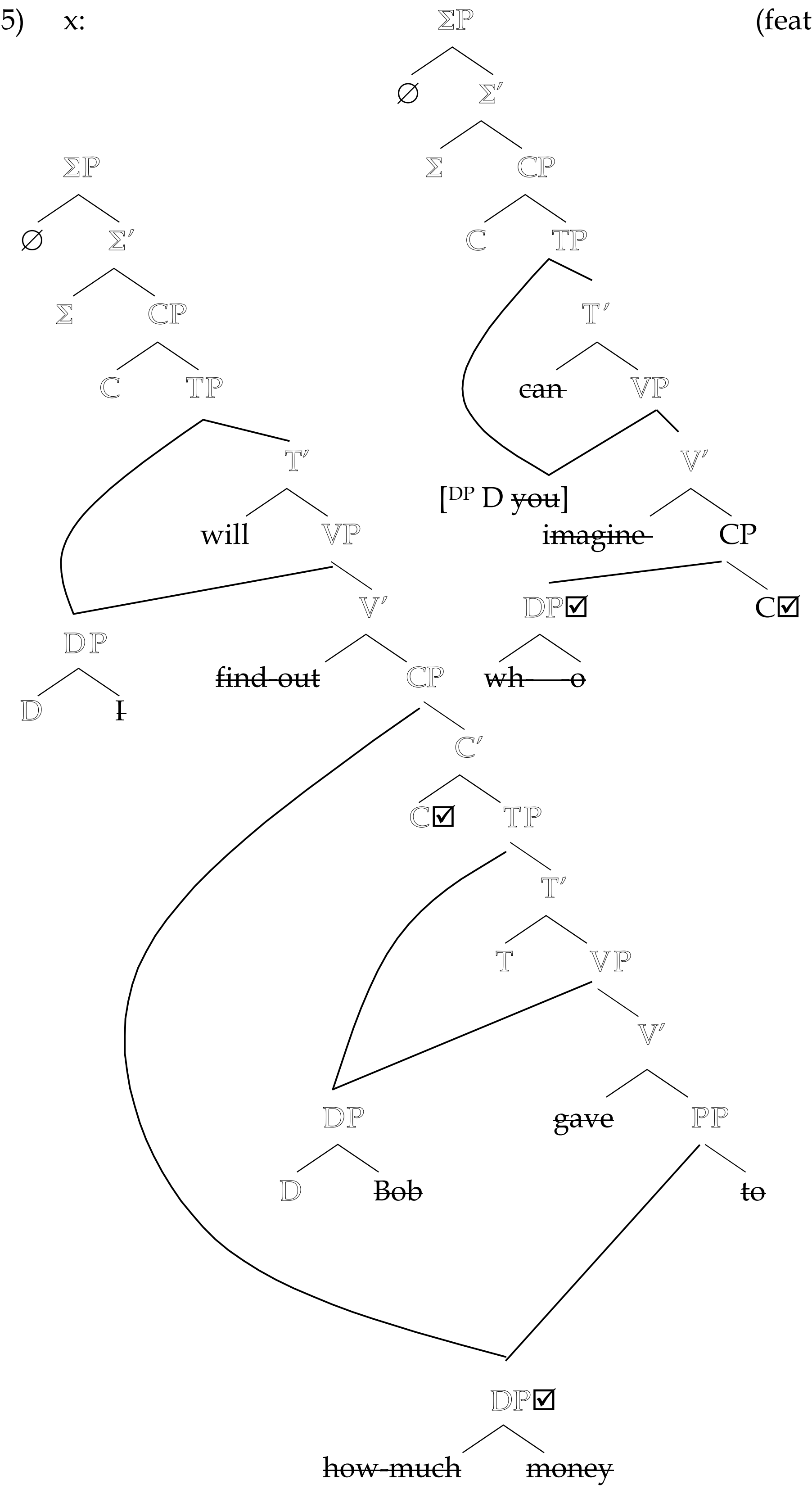
(85) w: (merge **C**)



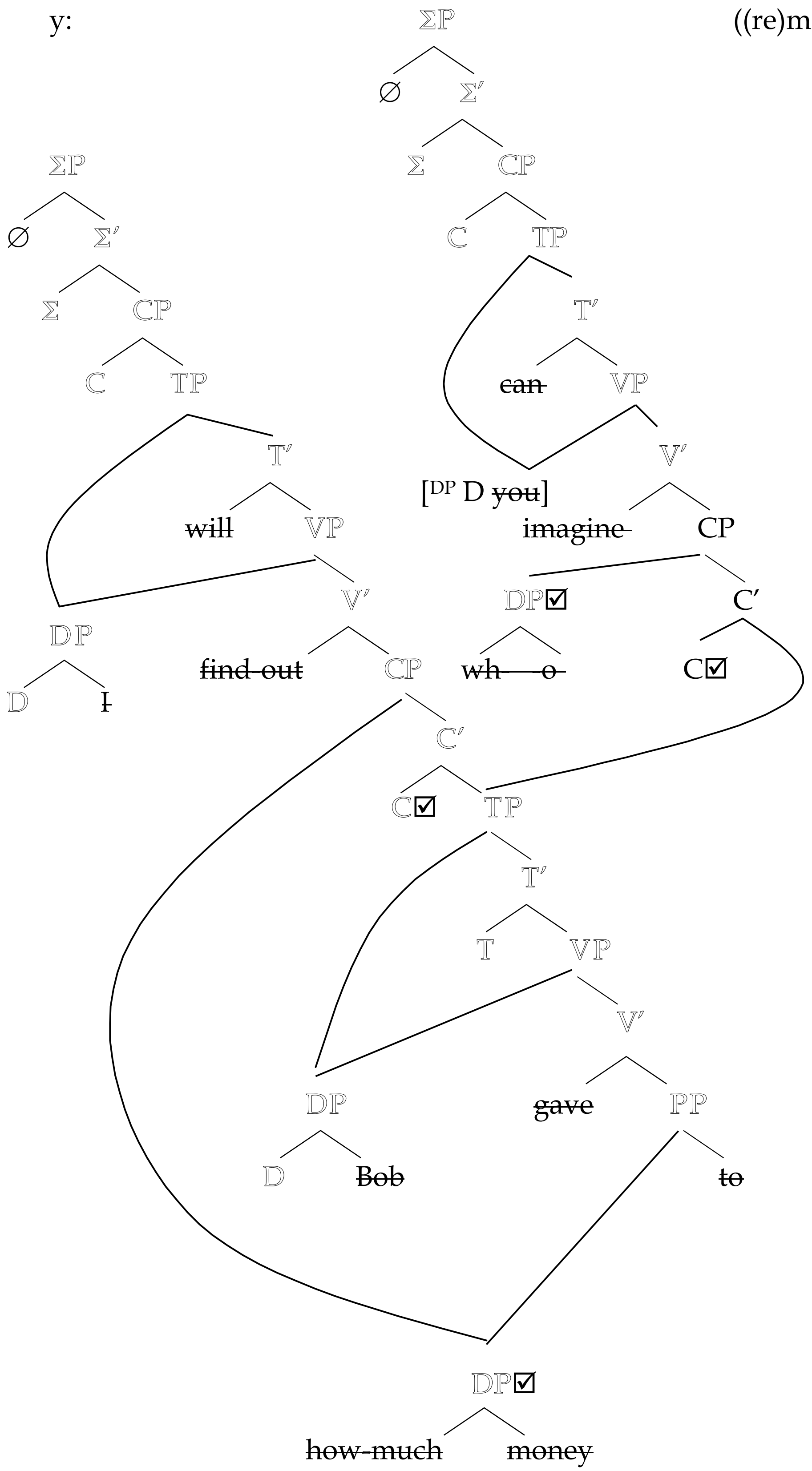
ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~can~~
VP
[DP D ~~you~~]
V′
~~imagine~~
CP
DP⚐
C⚐
~~wh- -o~~
ΣP
∅
Σ′
Σ
CP
C
TP
T′
will
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
C☑
TP
T′
T
VP
V′
DP
D
~~Bob~~
~~gave~~
PP
~~to~~
DP☑
~~how-much~~
~~money~~

The remainder of the derivation is as follows.

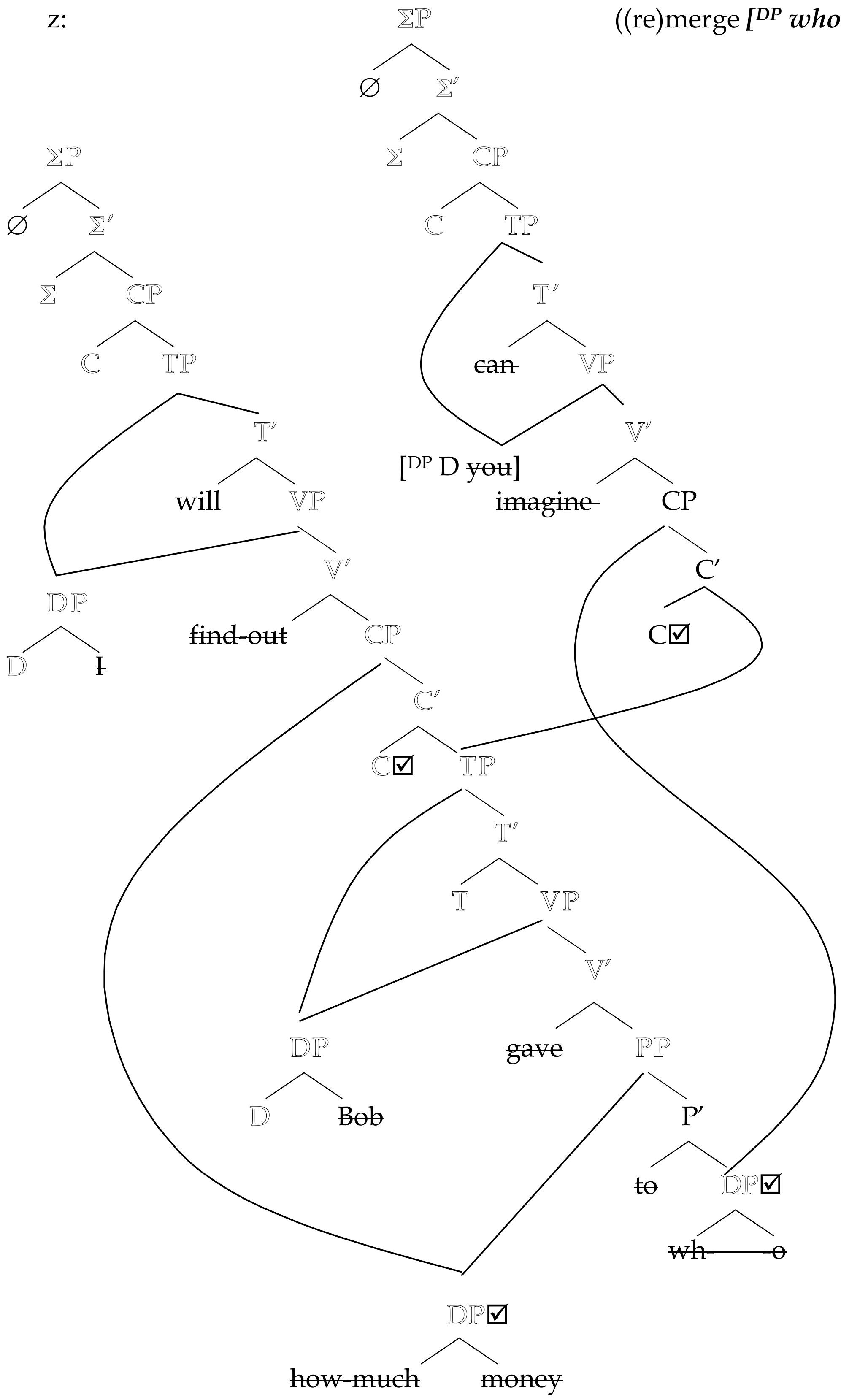
(85) x: (feature checking)



y: ((re)merge **TP**)

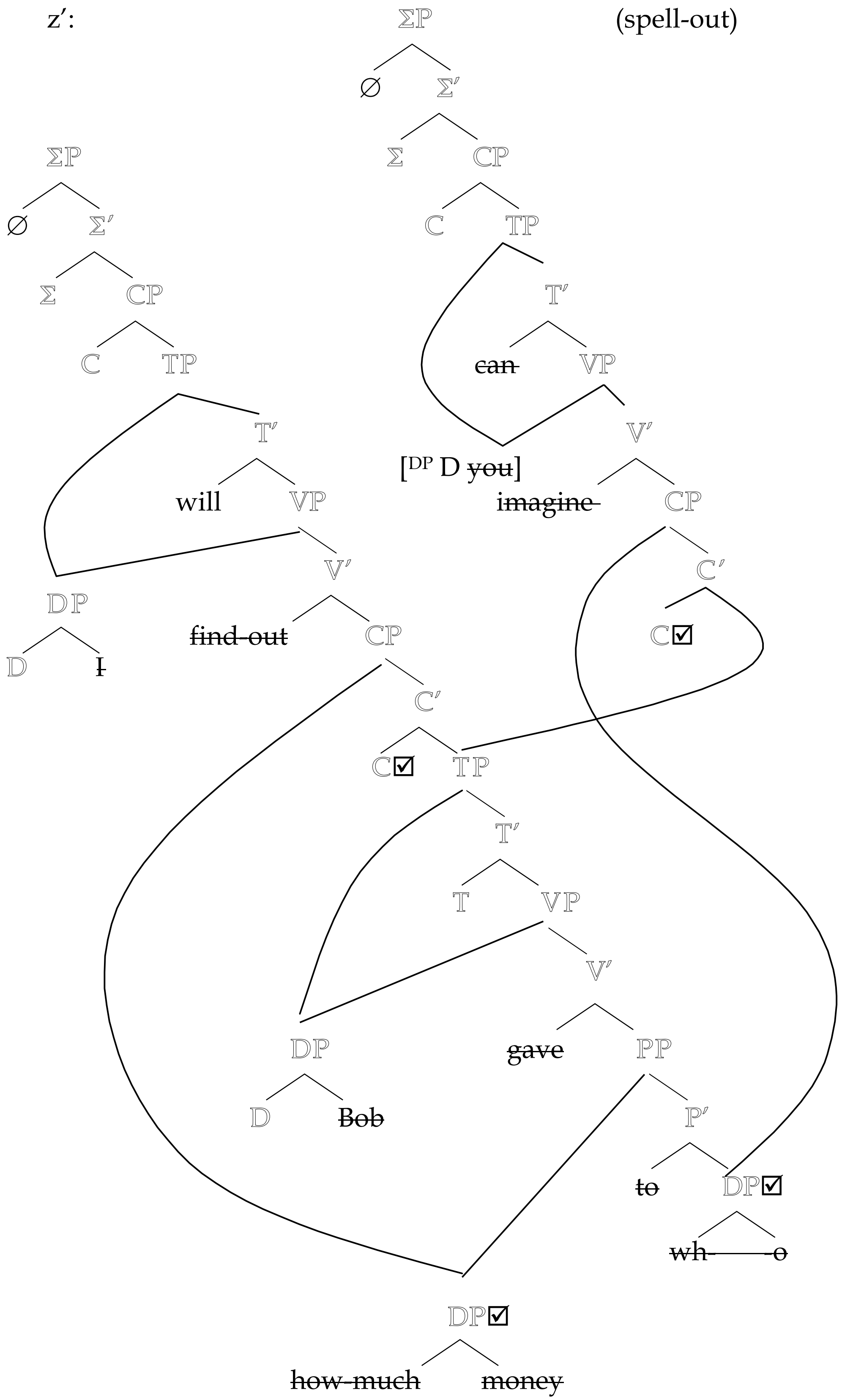

ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~can~~
VP
[DP D ~~you~~]
V′
~~imagine~~
CP
DP☑
~~wh- -o~~
C′
C☑
ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
C☑
TP
T′
T
VP
V′
DP
D
~~Bob~~
~~gave~~
PP
~~to~~
DP☑
~~how-much~~
~~money~~

z:



((re)merge ***[DP who]***)

z′:


ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~can~~
VP
[DP D ~~you~~]
V′
~~imagine~~
CP
C′
C☑
ΣP
∅
Σ′
Σ
CP
C
TP
T′
will
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
C☑
TP
T′
T
VP
V′
DP
D
~~Bob~~
~~gave~~
PP
P′
~~to~~
DP☑
~~wh o~~
DP☑
~~how-much~~
~~money~~

(spell-out)

The corresponding PF (super-)string is **[#I#]$^\cap$[#will#$^\cap$#find-out#$^\cap$#how-much#$^\cap$#money#]$^\cap$[#to#]$^\cap$[#you#]$^\cap$[#can#$^\cap$#imagine#$^\cap$#who#]**. Notice that, at no step in (85), the MLC (as defined in (81)) has been violated.

Now, let us take a look at the controversial case in (86).

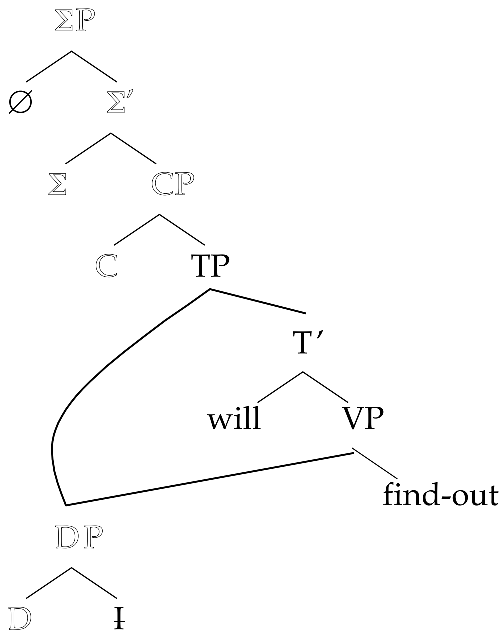
(86) I'll find out who Bob gave you can imagine how much money to.

The corresponding derivation would be as in (87).

First, the system randomly takes Ω as the starting point for the first derivational round.

The first steps are identical to the ones on derivation in (85-86), yielding the structure in (87a).

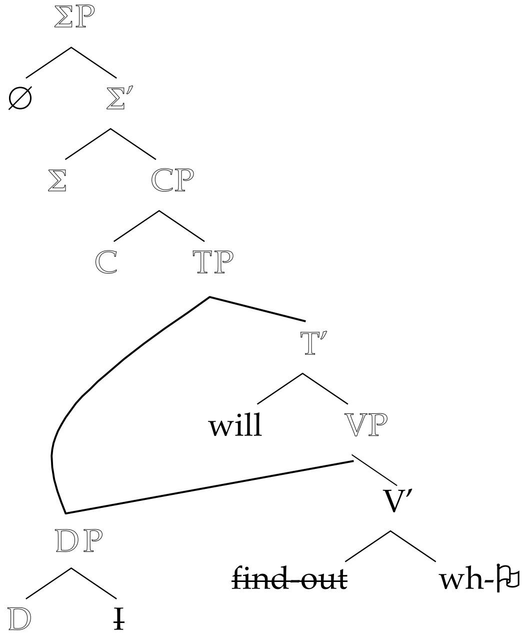
(87) a:



(merge **find-out**)

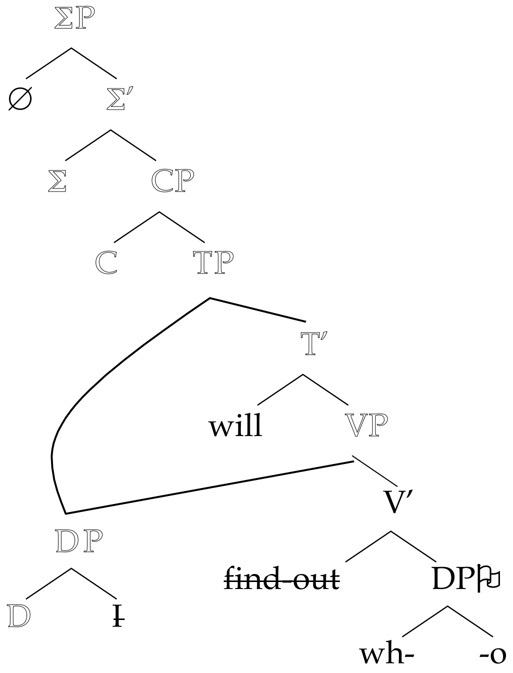
At this point, nothing prevents the system from tucking in **who**, as in (87b-d), instead of **how much money** as the temporary complement of find-out, since all relevant lexical tokens are present in relevant numeration.

(87) b:

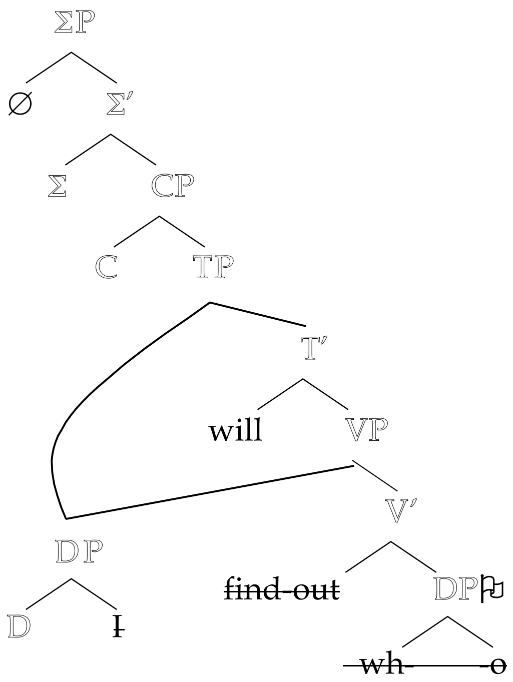


(merge **wh**)

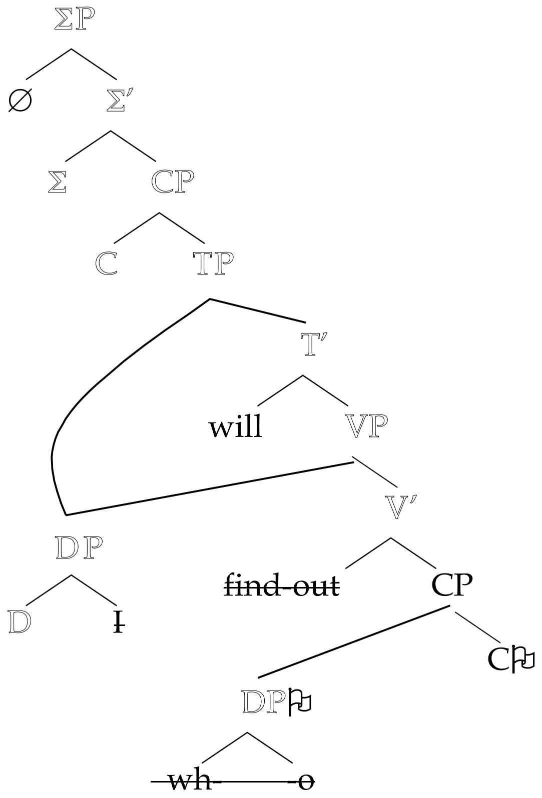
c:



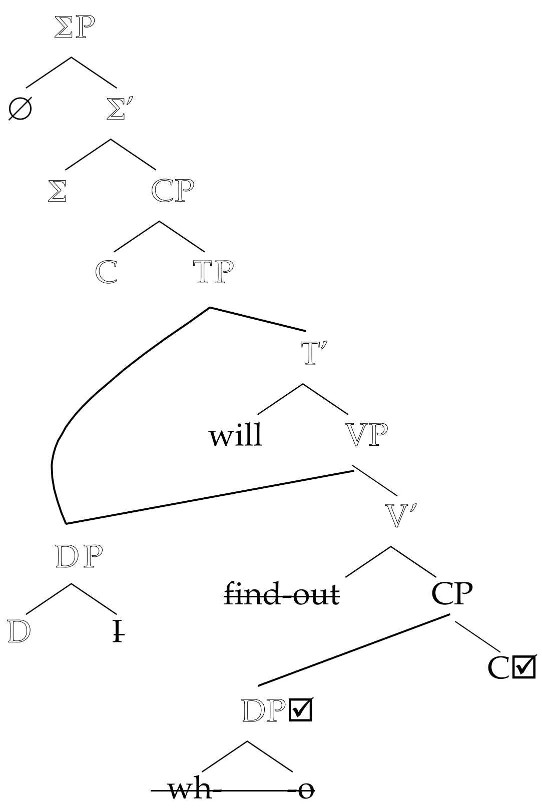
(merge ***person***)

d:  (spell-out)

The derivation, then, proceeds in the usual fashion, as follows.

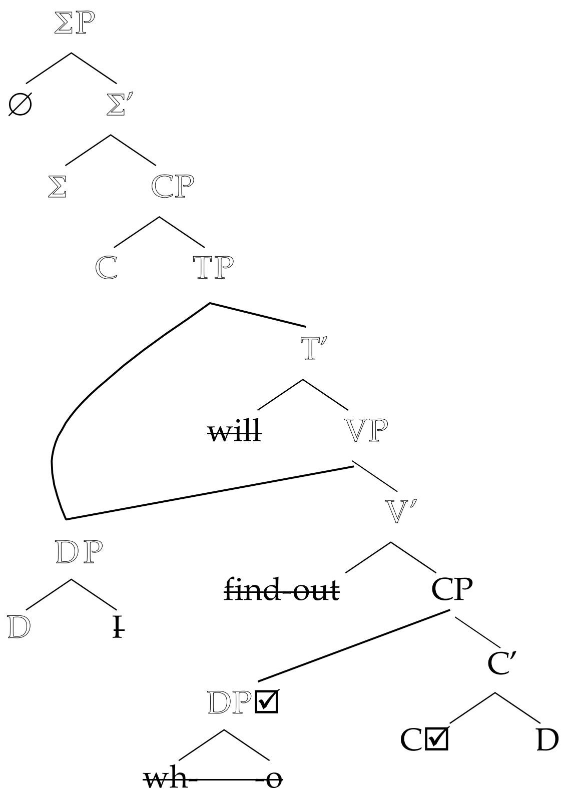
e:  (merge **C̲**)

f:



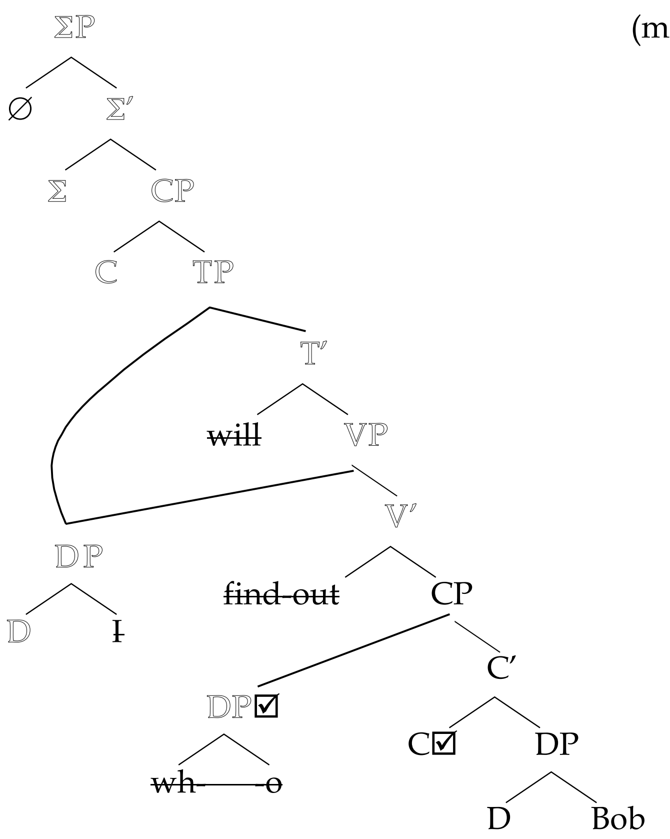
(feature checking)

g:

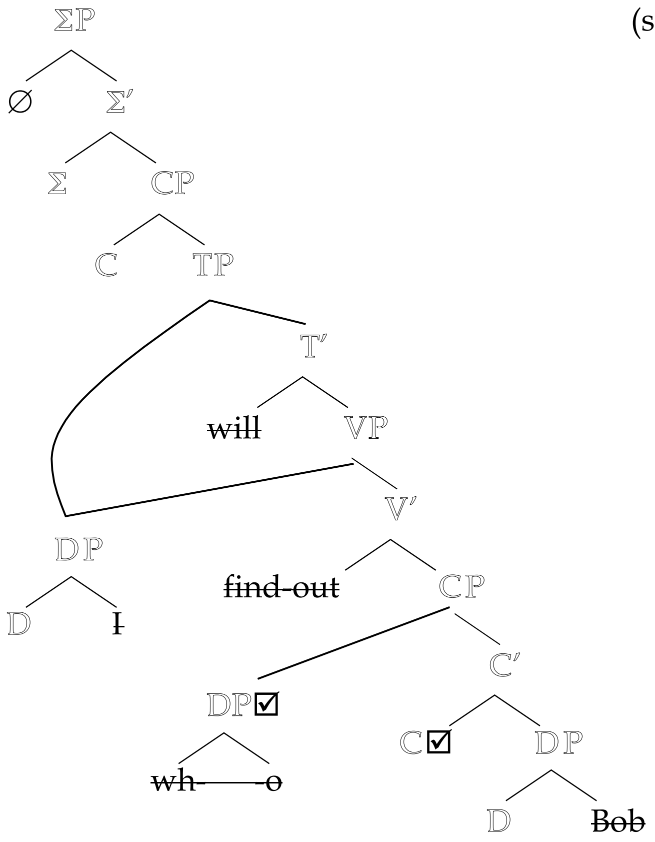


(merge **<u>D</u>**)

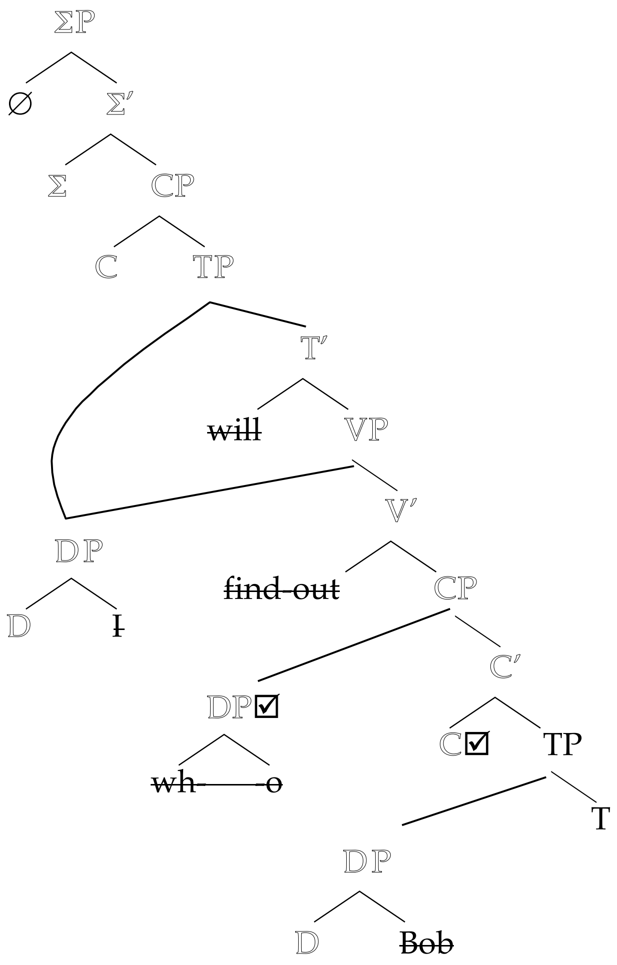
h: (merge **Bob**)



i: (spell-out)

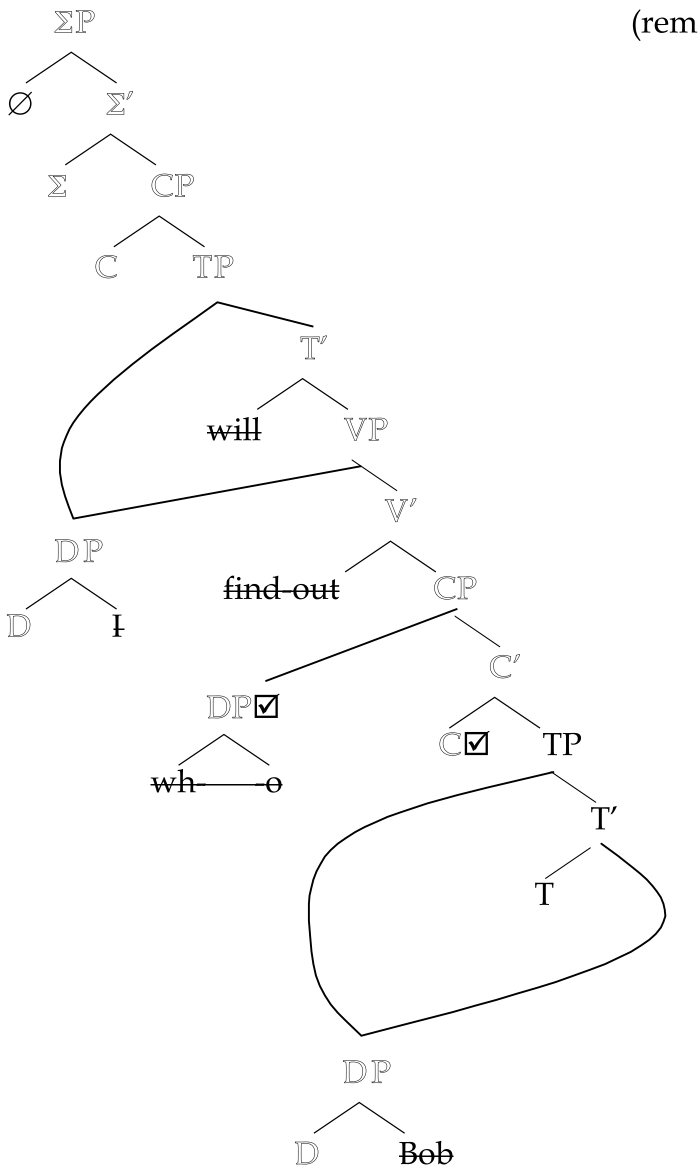


j:



(merge **T**)

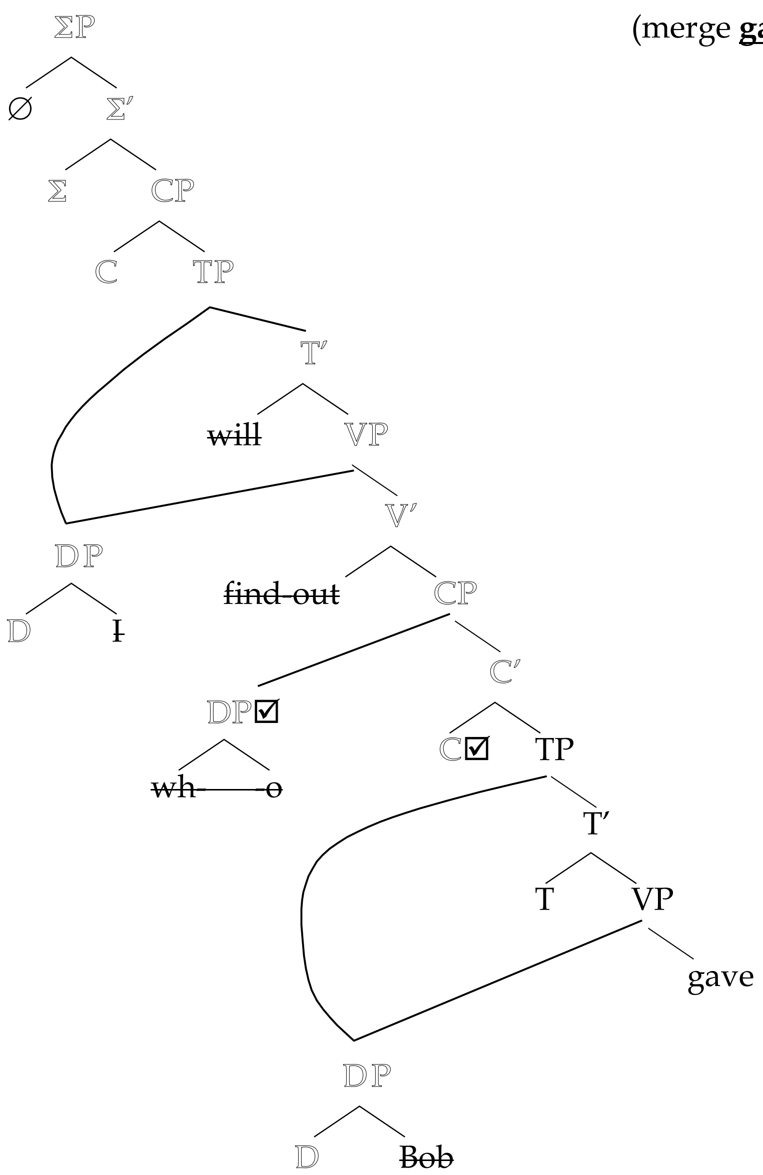
k:



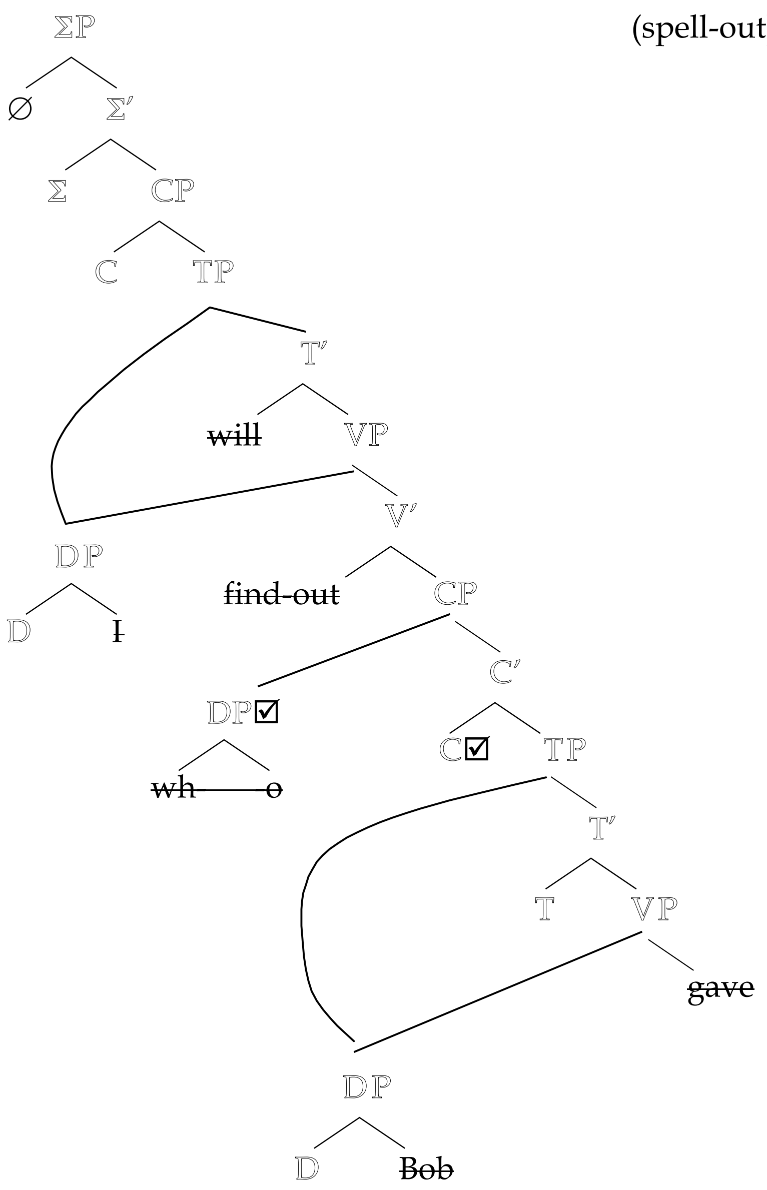
(remerge ***[^{DP} D Bob]***)

l:

(merge **<u>gave</u>**)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
DP☑
~~wh-~~
~~-o~~
C☑
TP
T′
T
VP
gave
DP
D
~~Bob~~

m:



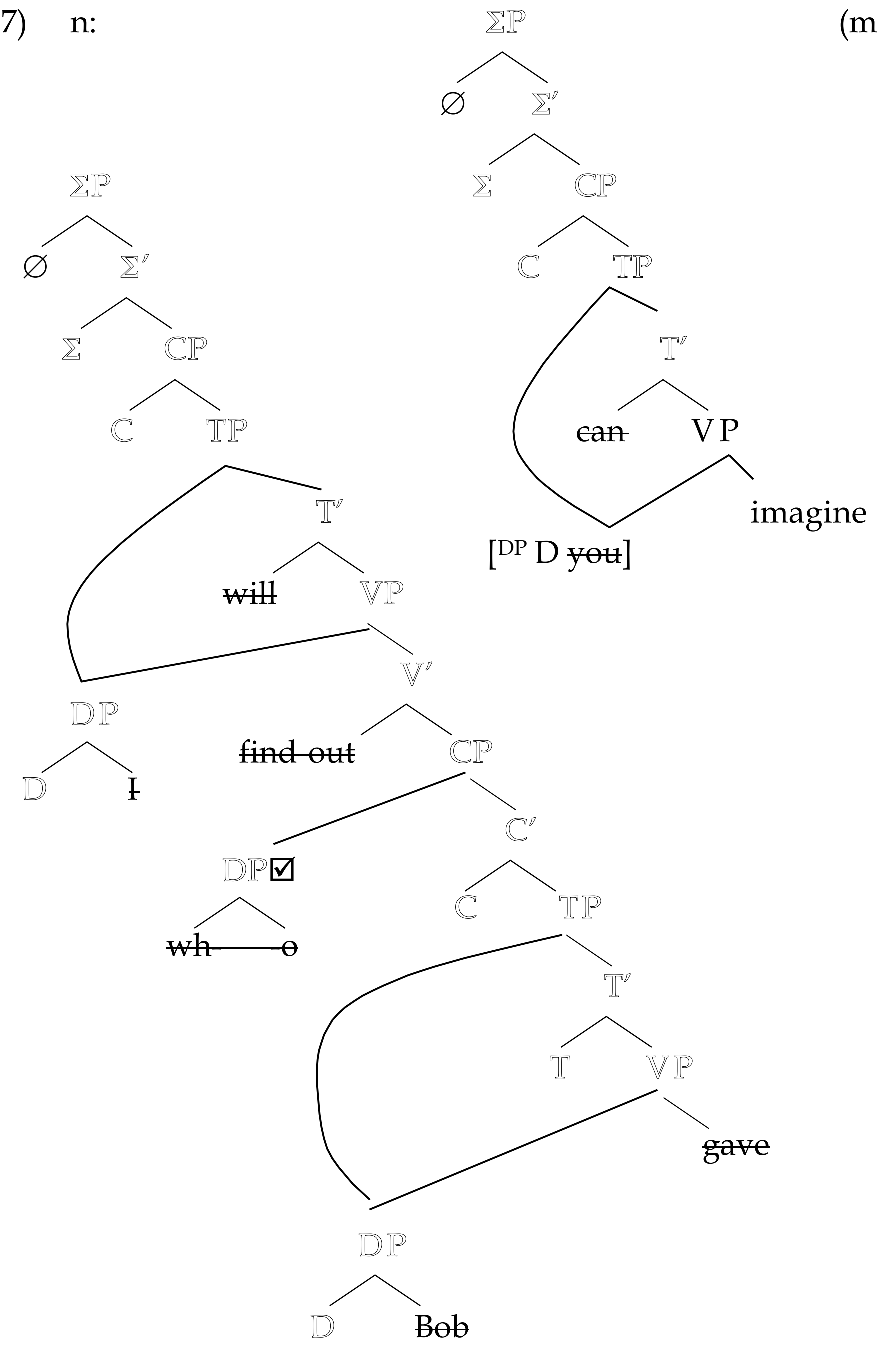
(spell-out)

Notice that spell-out has applied prematurely in step (87m), before the direct object is tucked in. As discussed before, it is crucial that the first derivational flow terminates at this point, otherwise the direct object will fail to c-command the target position in the subservient clause when it is time for it to remerge.

This marks the end of the first derivational flow. So far, the corresponding PF (super-)string is **[#I#]$^{\cap}$[#will#$^{\cap}$#find-out#$^{\cap}$who#]$^{\cap}$[#Bob#]$^{\cap}$[#gave#]**

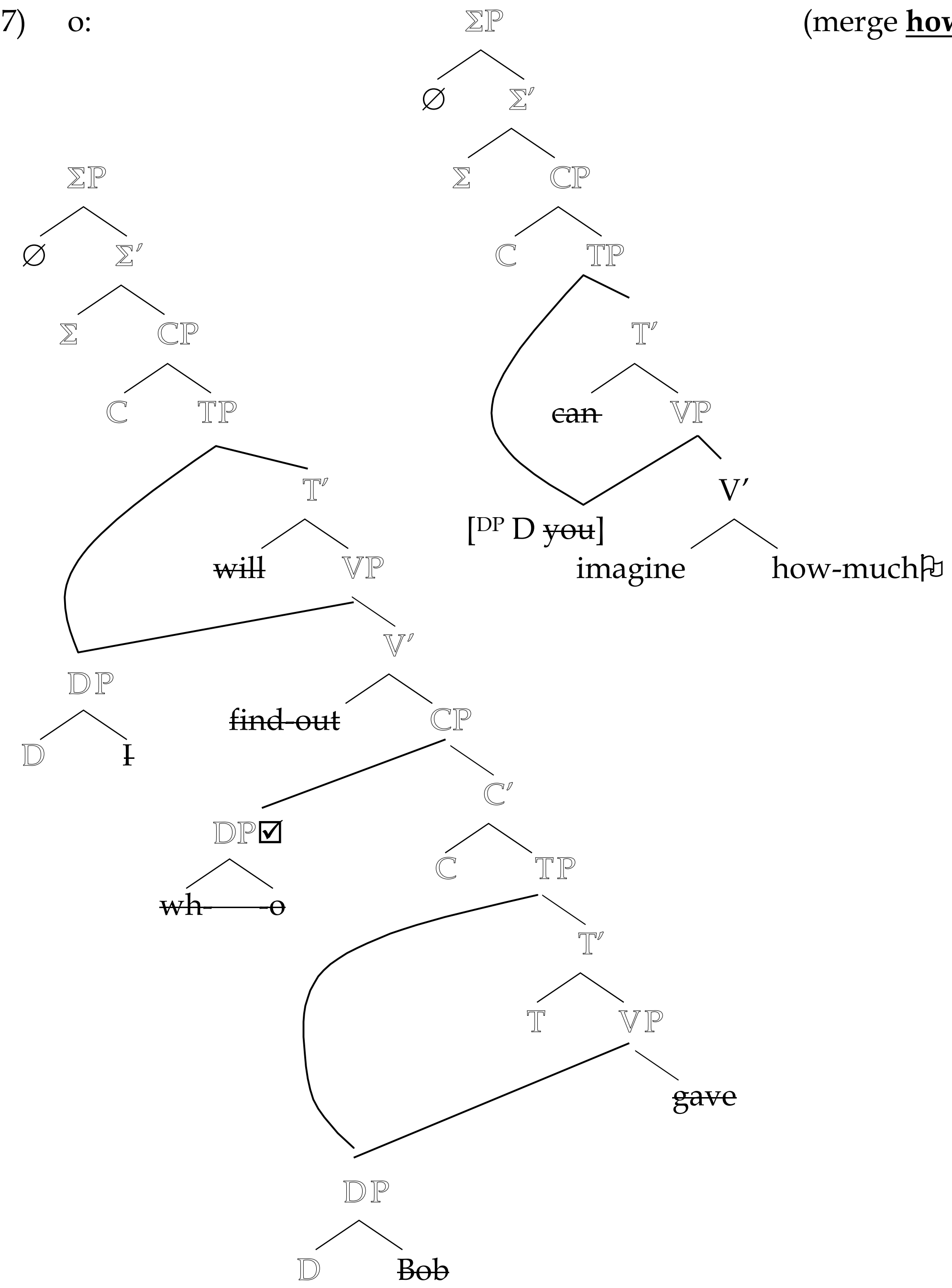
The computational system, then, shifts its attention to numeration Δ, and starts the second derivational flow, which proceeds in the usual fashion, from the top downwards, up to the point where the structure in (87n) obtains.

(87) n: (merge **imagine**)

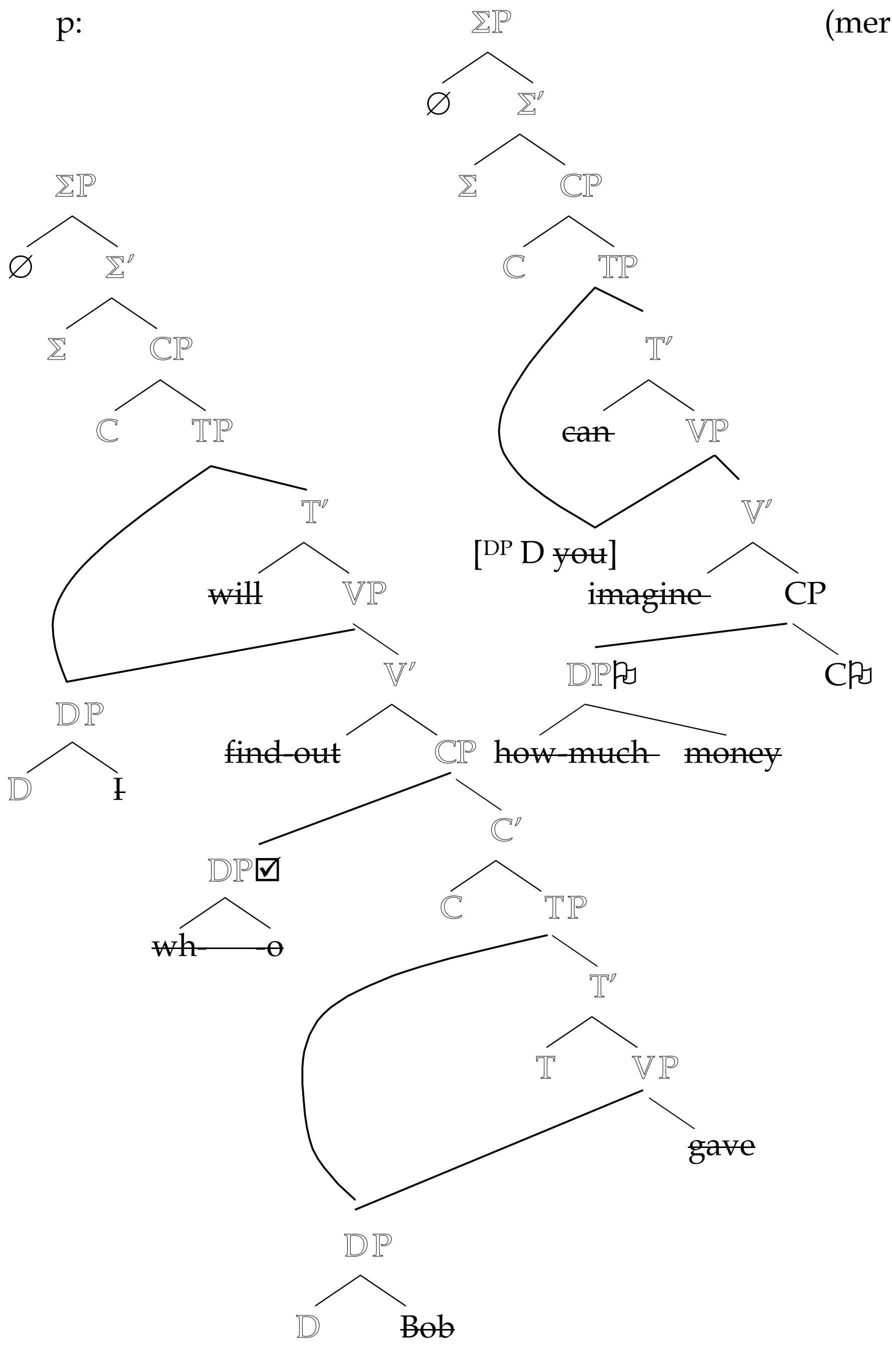


The remainder of the derivation is as follows.

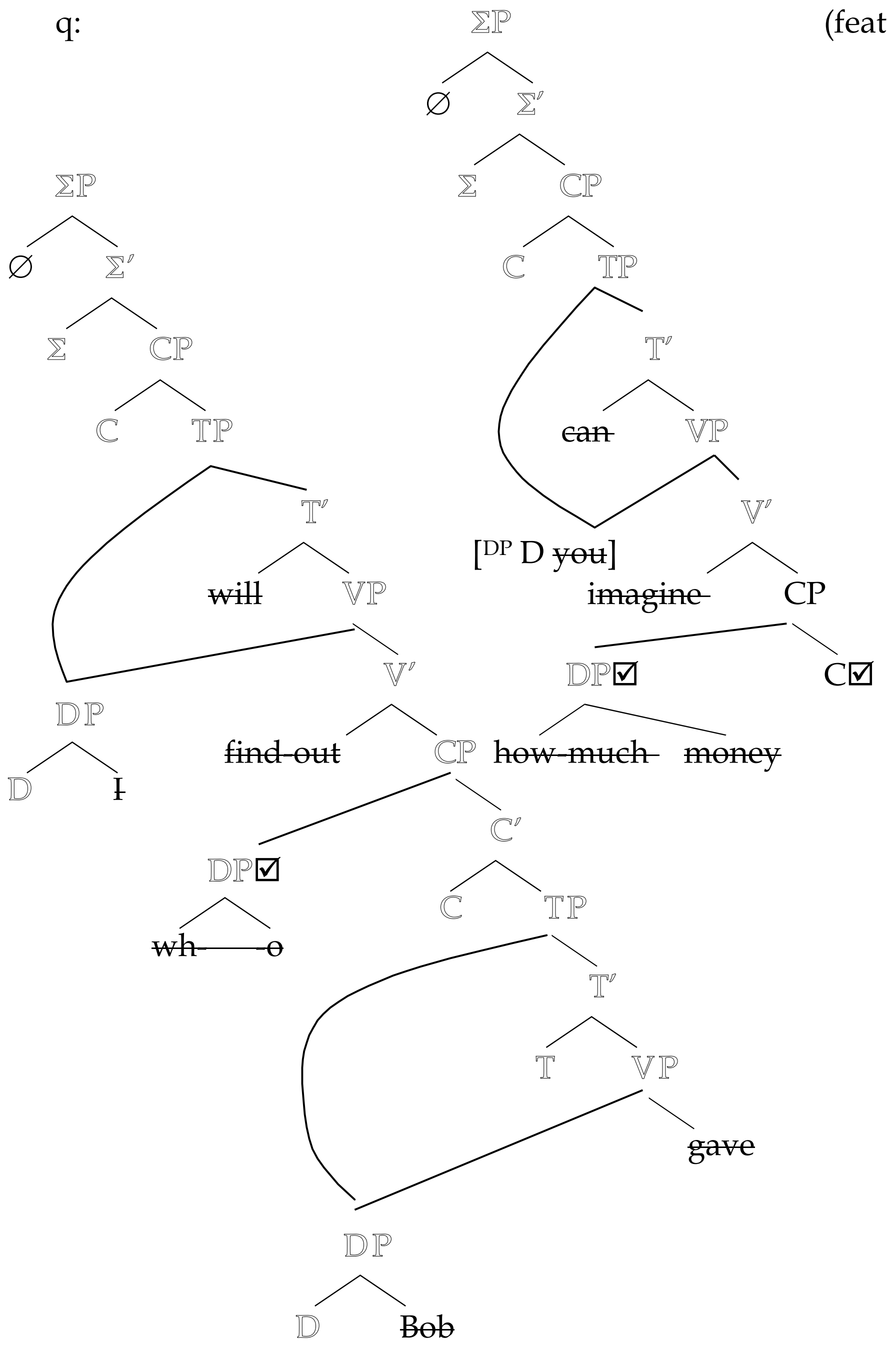
(87) o: (merge **how-much**)



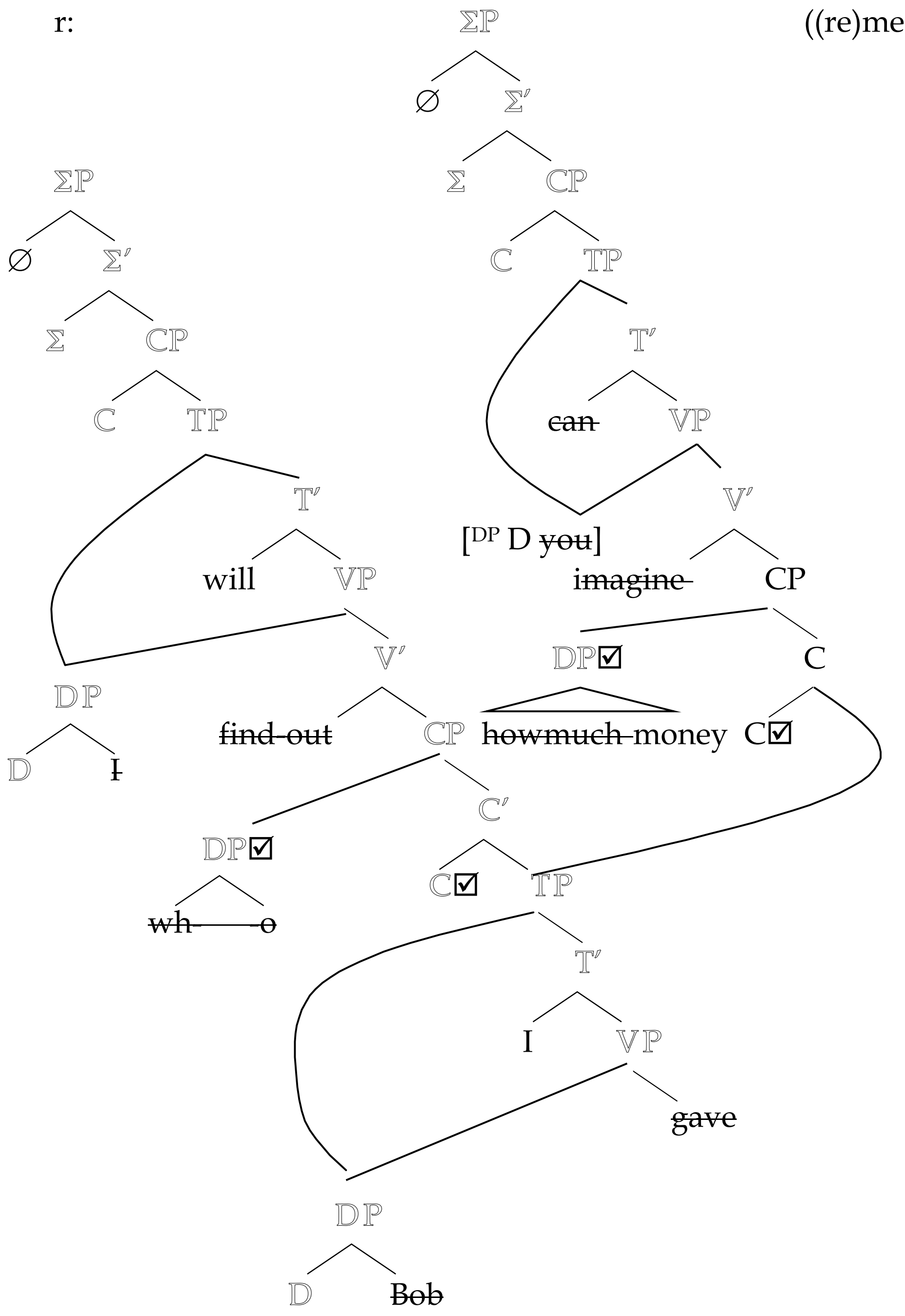
p: (merge **C**)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
can
VP
[DP D ~~you~~]
V′
~~imagine~~
CP
DP⚐
C⚐
~~how-much~~
~~money~~
ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~will~~
VP
V′
~~find-out~~
CP
DP
D
~~I~~
DP☑
~~wh-~~
~~-o~~
C′
C
TP
T′
T
VP
~~gave~~
DP
D
~~Bob~~

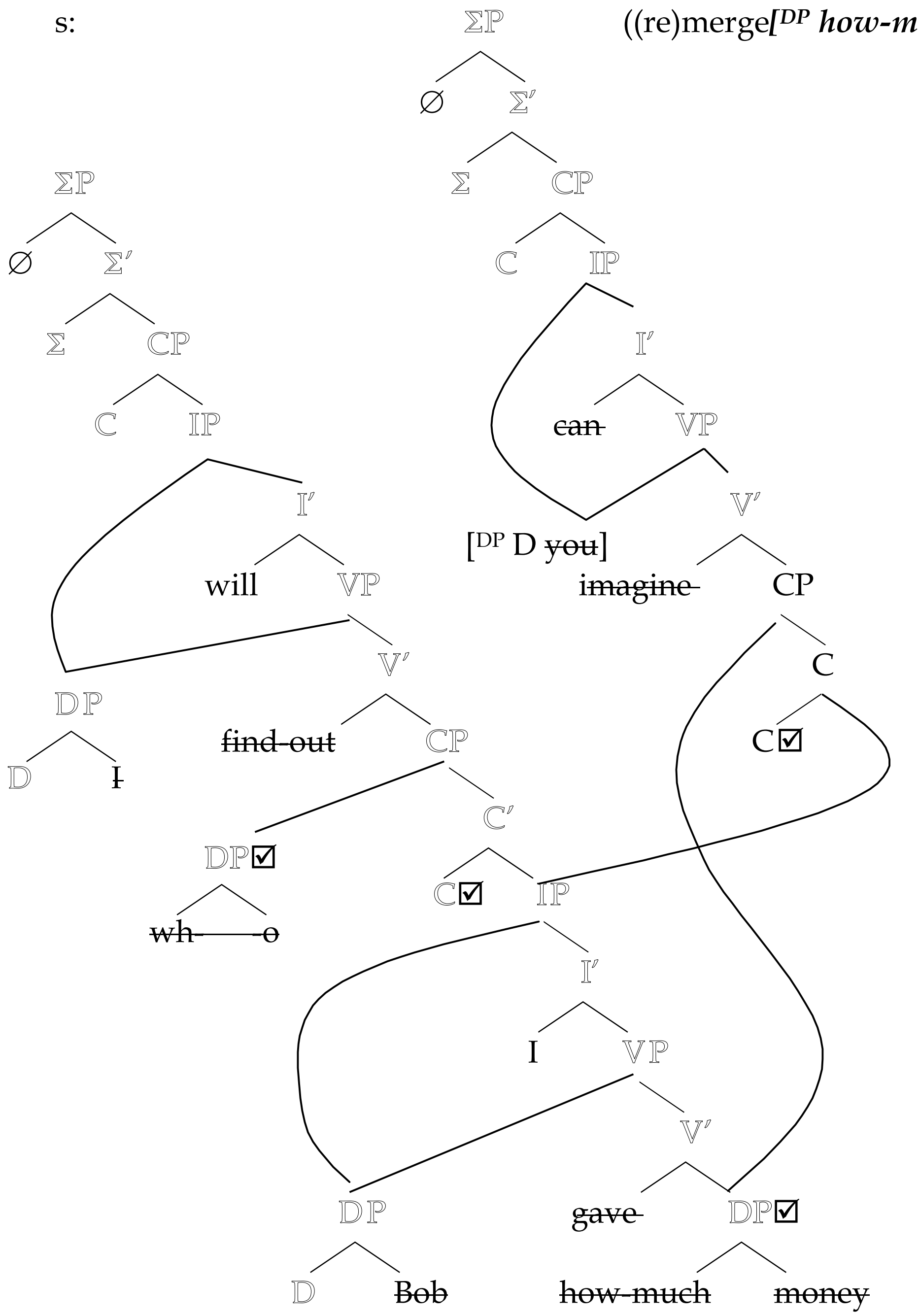
q: (feature checking)



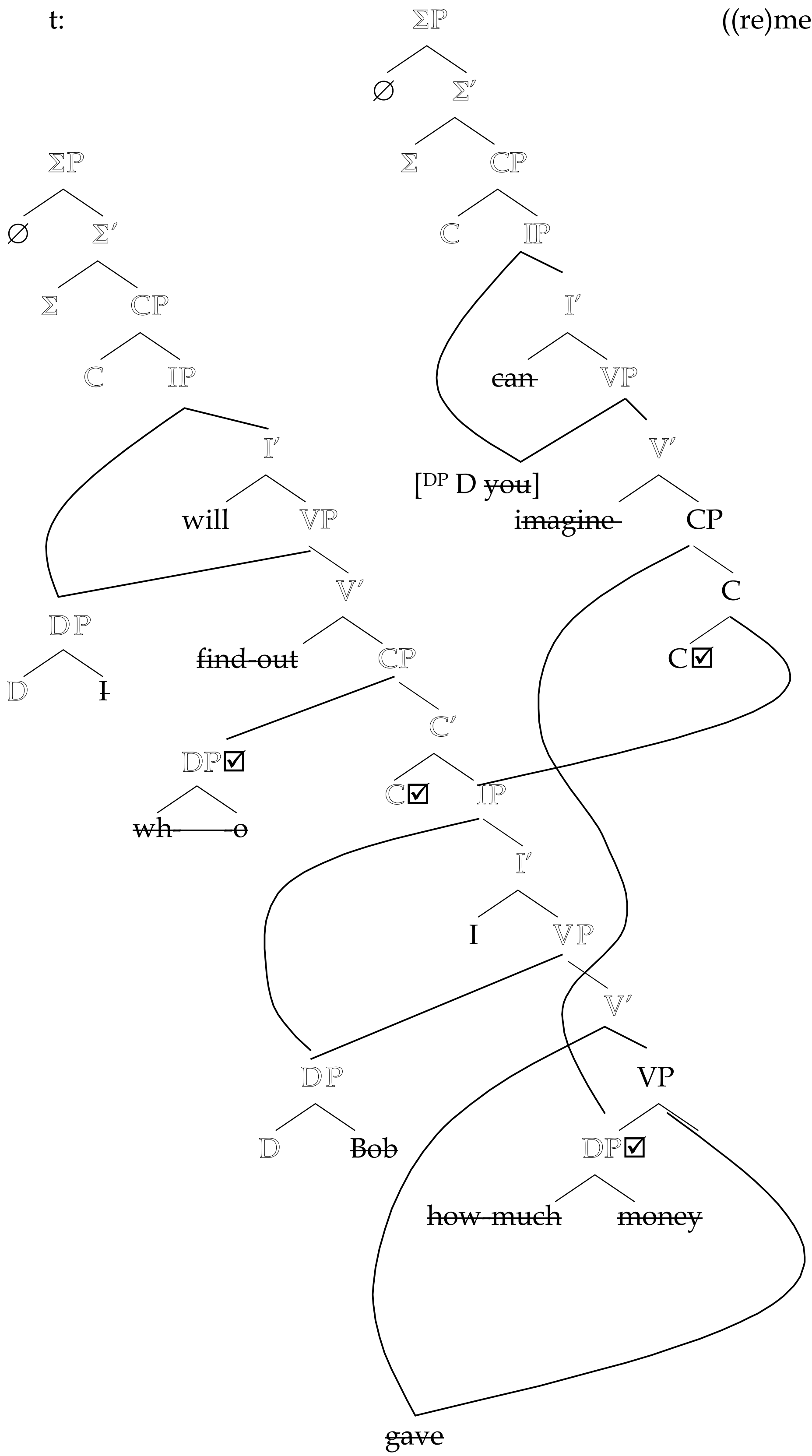
r: ((re)merge **TP**)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
~~can~~
VP
[DP D ~~you~~]
V′
~~imagine~~
CP
DP☑
C
~~howmuch~~ money
C☑
ΣP
∅
Σ′
Σ
CP
C
TP
T′
will
VP
V′
DP
D
~~I~~
~~find out~~
CP
C′
DP☑
~~wh-~~ ~~-o~~
C☑
TP
T′
I
VP
~~gave~~
DP
D
~~Bob~~

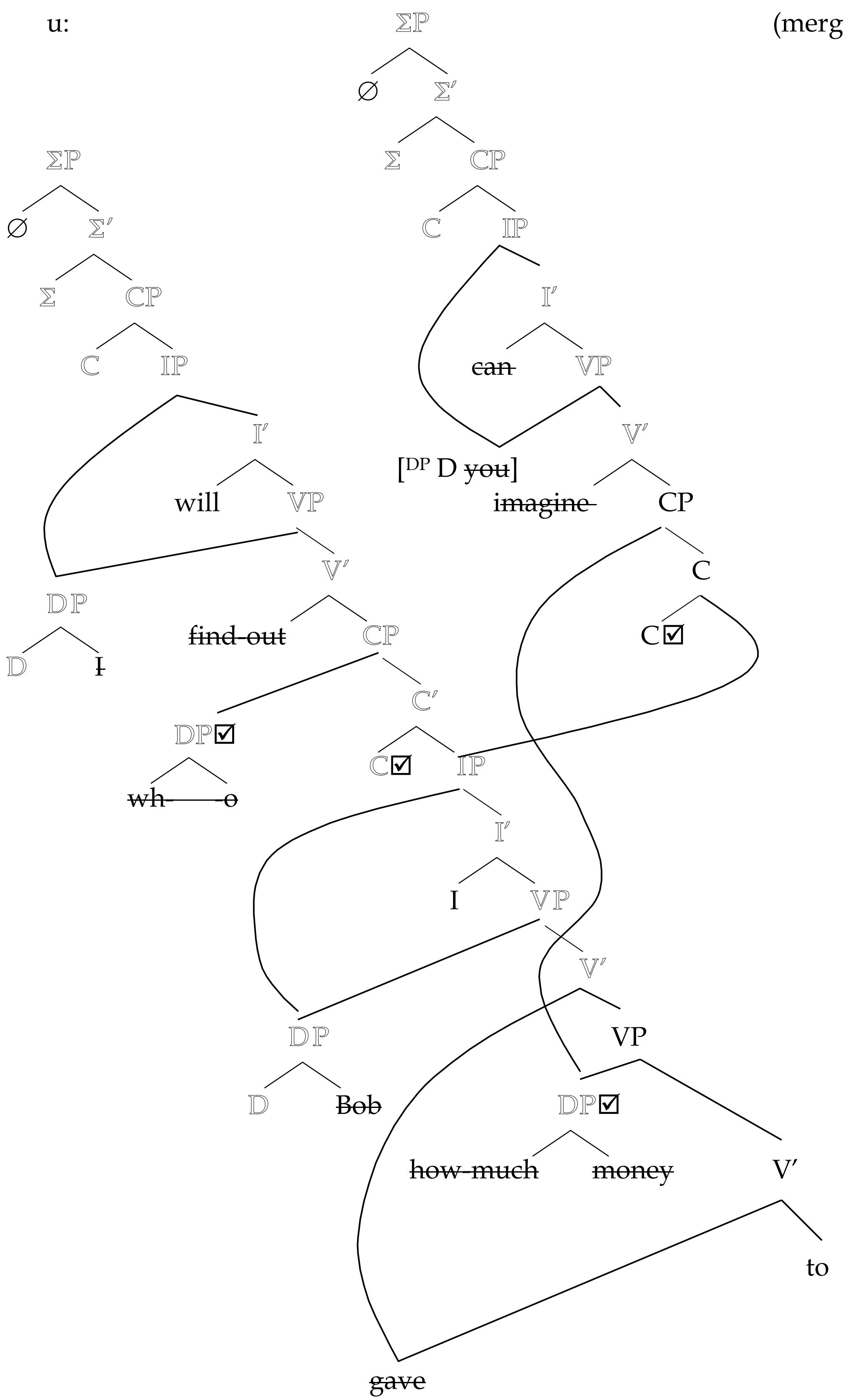
s: ((re)merge*[^{DP} how-much money]*)



t: ((re)merge **gave**)



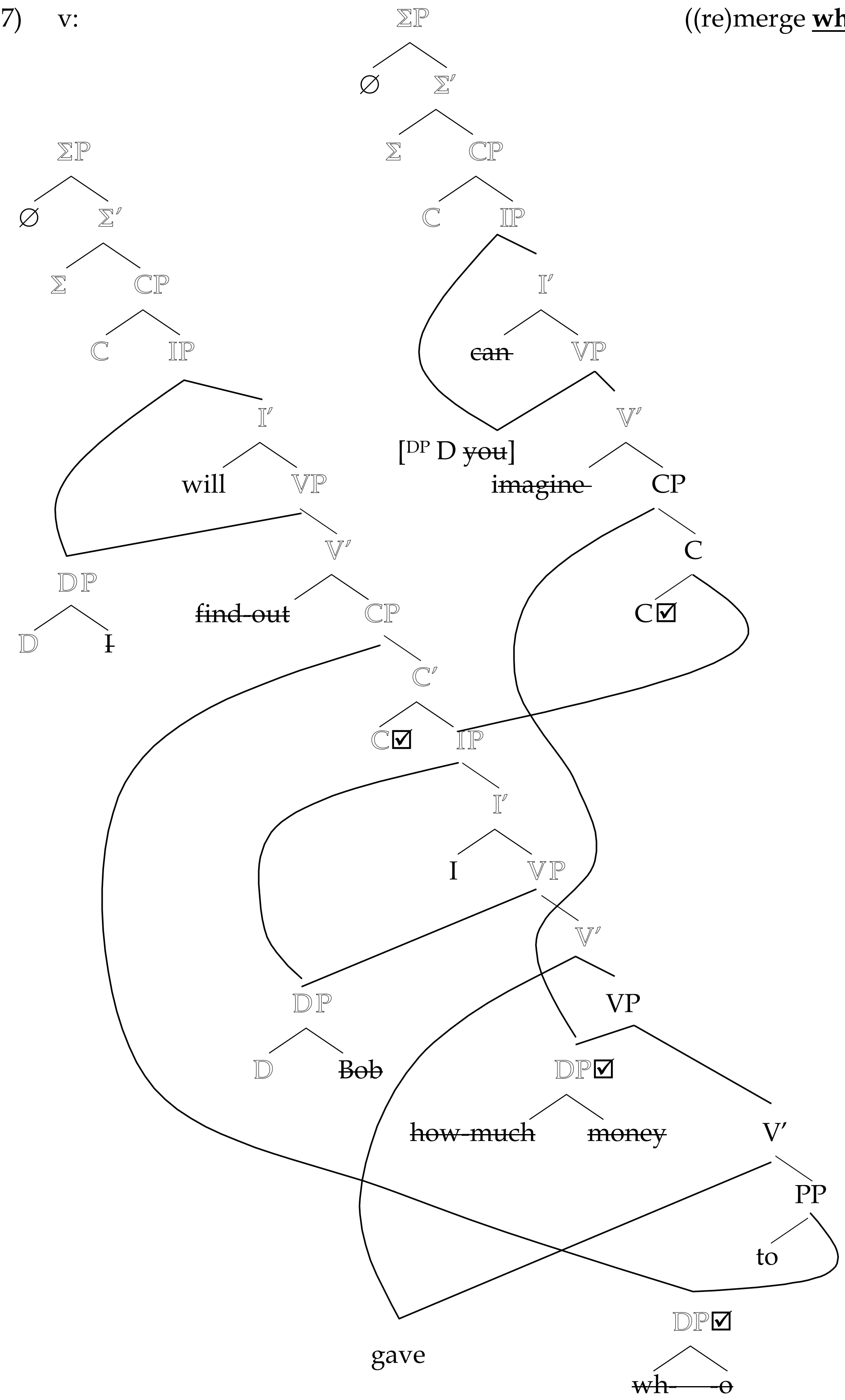
u: (merge **to**)

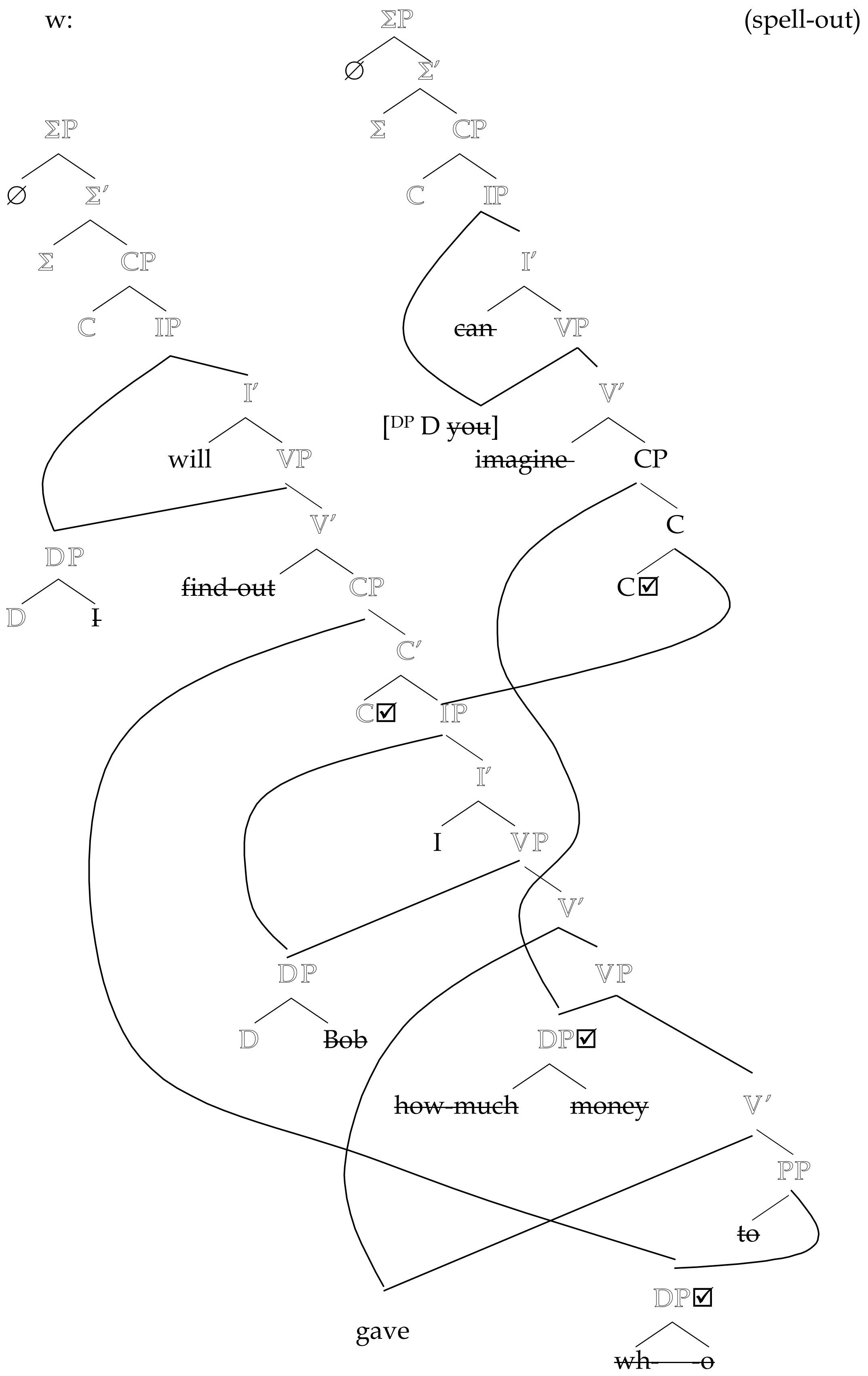

ΣP
∅
Σ′
Σ
CP
C
IP
I′
~~can~~
VP
[DP D ~~you~~]
V′
~~imagine~~
CP
C
C☑
ΣP
∅
Σ′
Σ
CP
C
IP
I′
will
VP
V′
DP
D
~~I~~
~~find-out~~
CP
DP☑
~~wh-~~
~~-o~~
C′
C☑
IP
I′
I
VP
V′
DP
D
~~Bob~~
VP
DP☑
~~how-much~~
~~money~~
V′
to
~~gave~~

The step in (87u) is a crucial one. The WH-phrase **how much money** is introduced in the c-command path of **who** before who lowers to its 'D-structure position' (so to speak). Notice, however, that, at this point, **how much money** is no longer active, as is [wh] has already been checked in a parallel domain outside the scope of **who**. Therefore, the MLC is satisfied.

The derivation continues as follows.

(87) v: ((re)merge **who**)




w:
(spell-out)
ΣP
∅
Σ′
Σ
CP
C
IP
I′
~~can~~
VP
[DP D ~~you~~]
V′
~~imagine~~
CP
C
C☑
ΣP
∅
Σ′
Σ
CP
C
IP
I′
will
VP
V′
DP
D
~~I~~
~~find-out~~
CP
C′
C☑
IP
I′
I
VP
V′
DP
D
~~Bob~~
VP
DP☑
~~how-much~~
~~money~~
V′
PP
~~to~~
DP☑
~~wh-~~
~~-o~~
gave

The corresponding PF (super-)string is **[#I#]⌒[#will#⌒#find-out#⌒#who#]⌒[#you#]⌒[#can#⌒#imagine#⌒how-much#⌒#money#]⌒[#to#]**.

Notice that, at no step in (87), the MLC (as defined in (82)) has been violated.

V.7. On the Restriction On Invasion at the Subject Position

As shown in **§II.7**, an important empirical generalization about syntactic amalgams is that invasive clauses can fit in the position of an object (cf. (88a) and (88b)) or an adjunct (cf. (88c)) of the master clause, but somehow they cannot fit in a subject position, as shown in (89).[154]

(88) a: Tom believes that Amy has been dating *I forget* ***who*** since last month.

b: Tom believes that Amy gave all her money to *I forget* ***who*** yesterday.

c: Tom said that Amy has been dating Bob since *I forget* ***when.***

(89) * Tom said that *I forgot* ***who*** is dating Amy**.**

[154] As previously mentioned in **§II.7**, the example in (89) is fully acceptable under the interpretation corresponding to the structures in (i). This reading is irrelevant for our purposes, as they are cases of ordinary embedding, rather then syntactic amalgamation.

(i) [^{CP} C [^{TP} Tom$_3$ T [^{VP} t$_3$ said [^{CP} that [^{IP} I$_2$ T [^{VP} t$_2$ forgot [^{CP} who$_1$ [^{IP} t$_1$ is [^{VP} t$_1$ dating Amy]]]]]]]]]

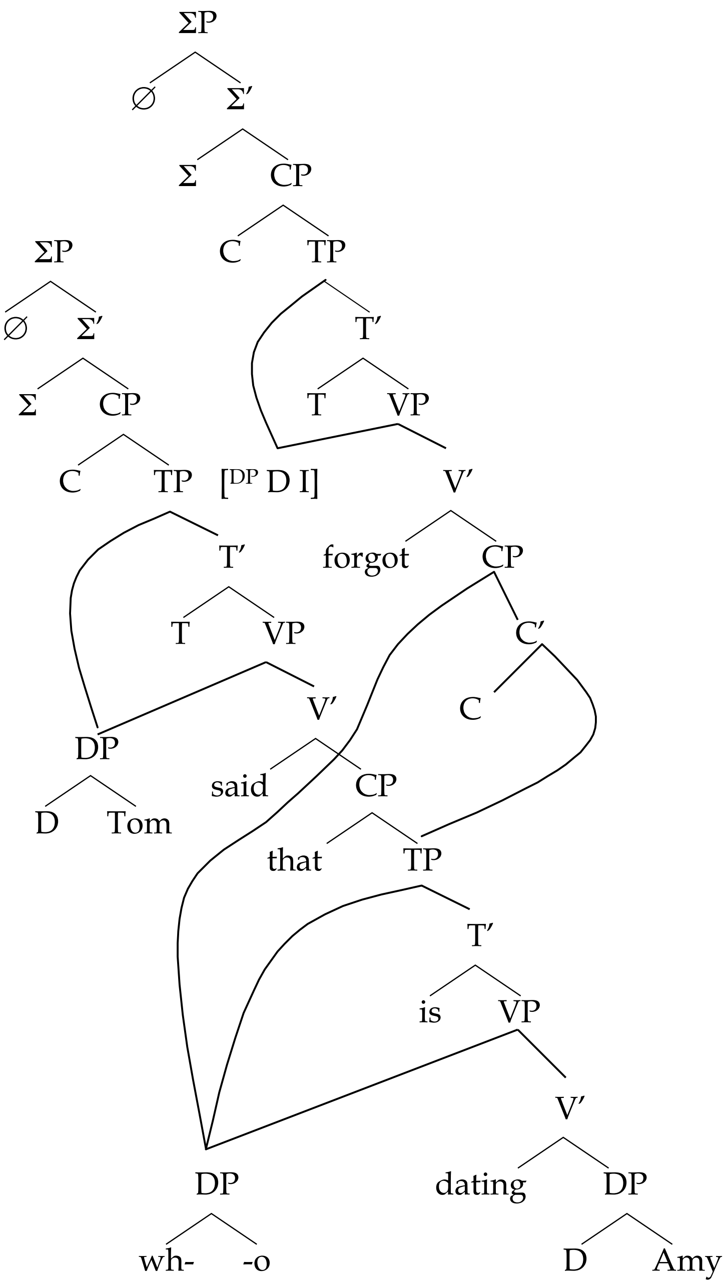
Thus, there seems to be a constraint on what counts as a legitimate 'invasion point'. Such a constraint can be stated along the lines of (91).

(91) A DP that occupies a spec/TP position in the master clause cannot simultaneously occupy a spec/CP position in a subservient clause.

Although descriptively adequate, this is obviously a mere stipulation. Given the assumptions about trans-sentential shared constituency that I have been assuming so far, it seems rather mysterious as to why the generalization behind the stipulation in (91) should hold.

From a representational point of view, without assuming the stipulation in (91), there is nothing wrong with the structure in (92), which would correspond to the unacceptable example in (89) above.

(92)



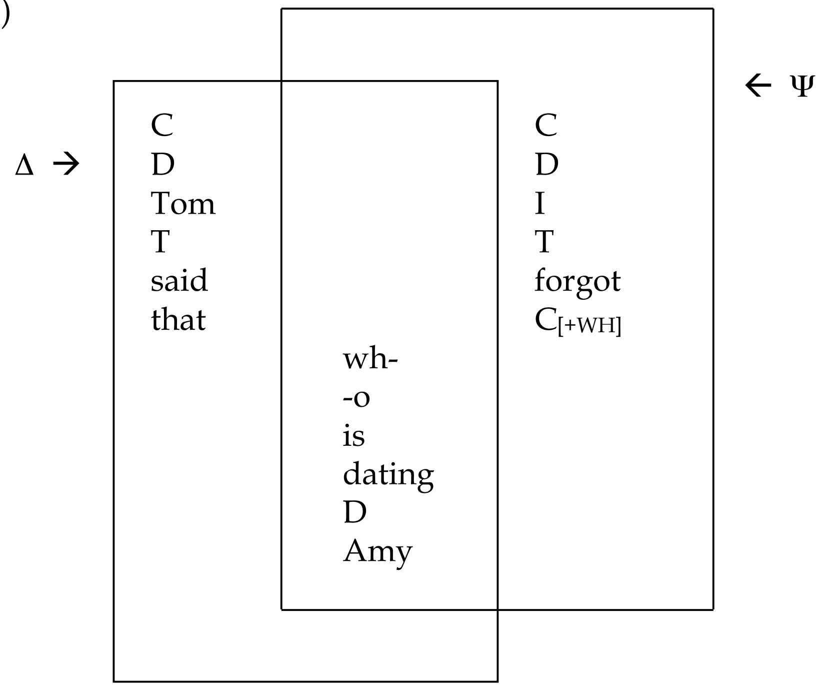
Thus, at first sight, it seems that such constraint on invasion cannot be straightforwardly reduced to deeper principles.

However, once we assume a derivational approach to syntax, and — crucially — once we implement such view in terms of a 'generalized tucking in' mechanics to phrase structure building (cf. **§IV.3**), then we can correctly predict (89) to be ungrammatical.

V.7.1 Finite Clauses.

The relevant (non-convergent) derivation for the structure in (92) would be as follows. The starting point is the input (93)

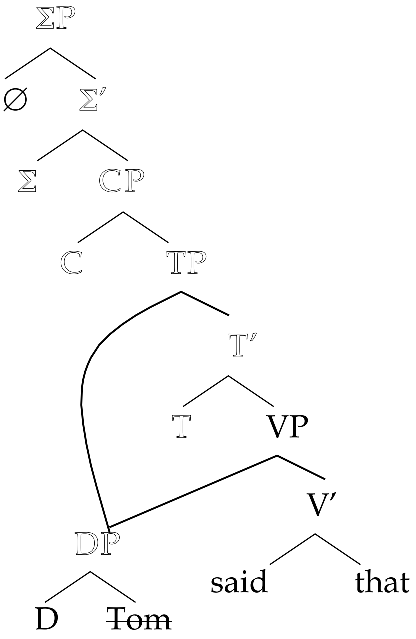
(93)



The computational system starts by randomly zooming into numeration Δ to begin the structure-building process. Given the conception about the nature of the grammar and the parser implied by the derivational system outlined in chapter **IV** (heavily inspired by Phillips' (1996, 2003) work), the choice of Δ automatically makes the sentence built from Δ the master clause, with the one built from Ψ being subservient to it. The PF representation of the whole Siamese-Tree structure is built incrementally, as the derivation proceeds, with smaller chunks of each subcomputation getting successively spelled-out and being pronounced with respect to each other in an order that directly reflects the order in which syntax delivers them to the A-P system.

First, the computational system builds the non-shared portion of the phrase-marker corresponding to Δ. The lexical tokens in the non-intersecting portion of Δ get combined one by one, in a top-to-bottom fashion, up to the point when the structure in (94) obtains.

(94)



At this step, the PF representation is as in (95).

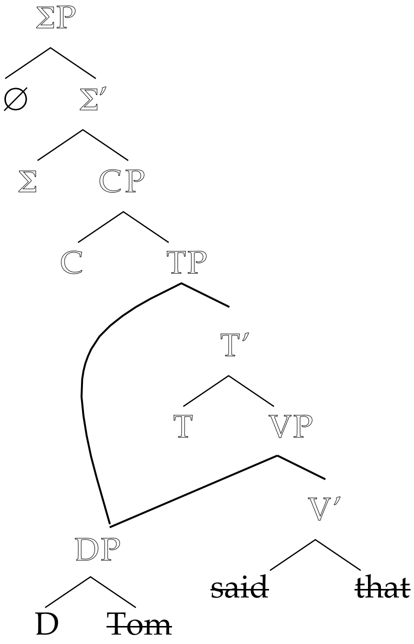
(95) [#Tom#]

The next natural step is to build the subject of the sentence introduced by the complementizer already present in the structure. However, at this point, the system can recognize that the subject that is about to be built would be a WH-phrase that has no matching C head within the domain established by Δ. Rather, its matching C is the [+WH] complementizer in Ψ. As will be discussed shortly, if the subject is build at this point, it will not be able to undergo a feature-checking operation later on.

The system is then forced to abandon the computation of the master clause, leaving the structure incomplete, to be completed by the same subcomputation that builds the subservient clause.

Right before one subcomputation hands the structure to the other, spell-out needs to apply to guarantee LCA satisfaction. This is indicated in (96), whose corresponding PF representation is as in (97).

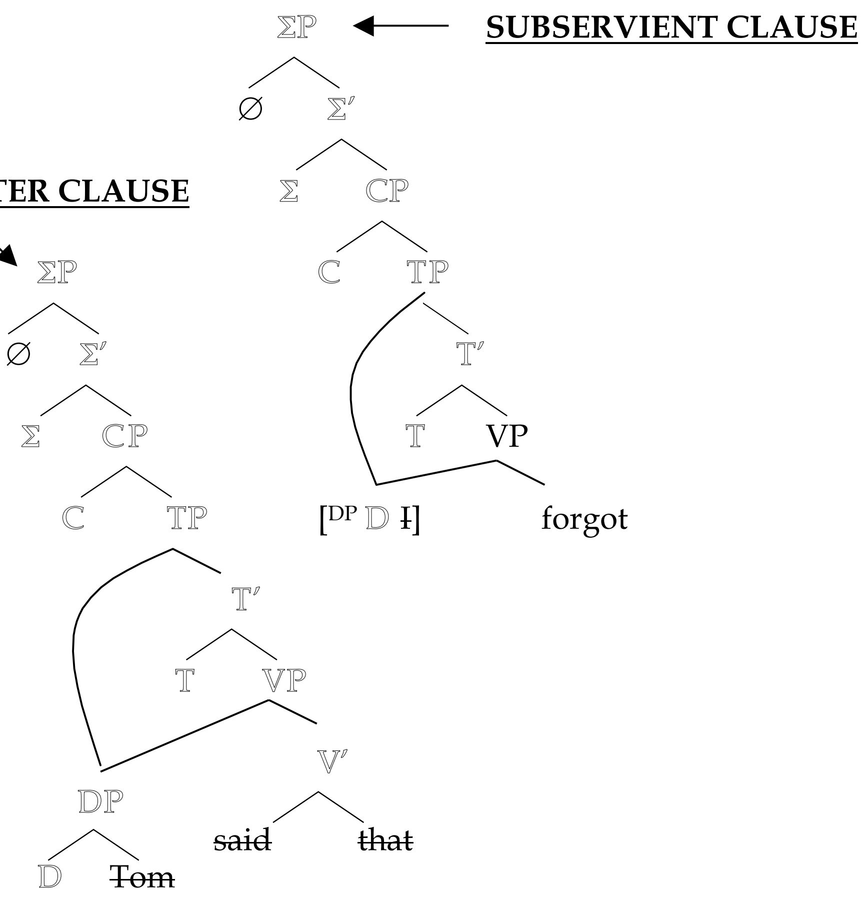
(96)



(97) [#Tom#]$^{\cap}$[#said#$^{\cap}$#that#]

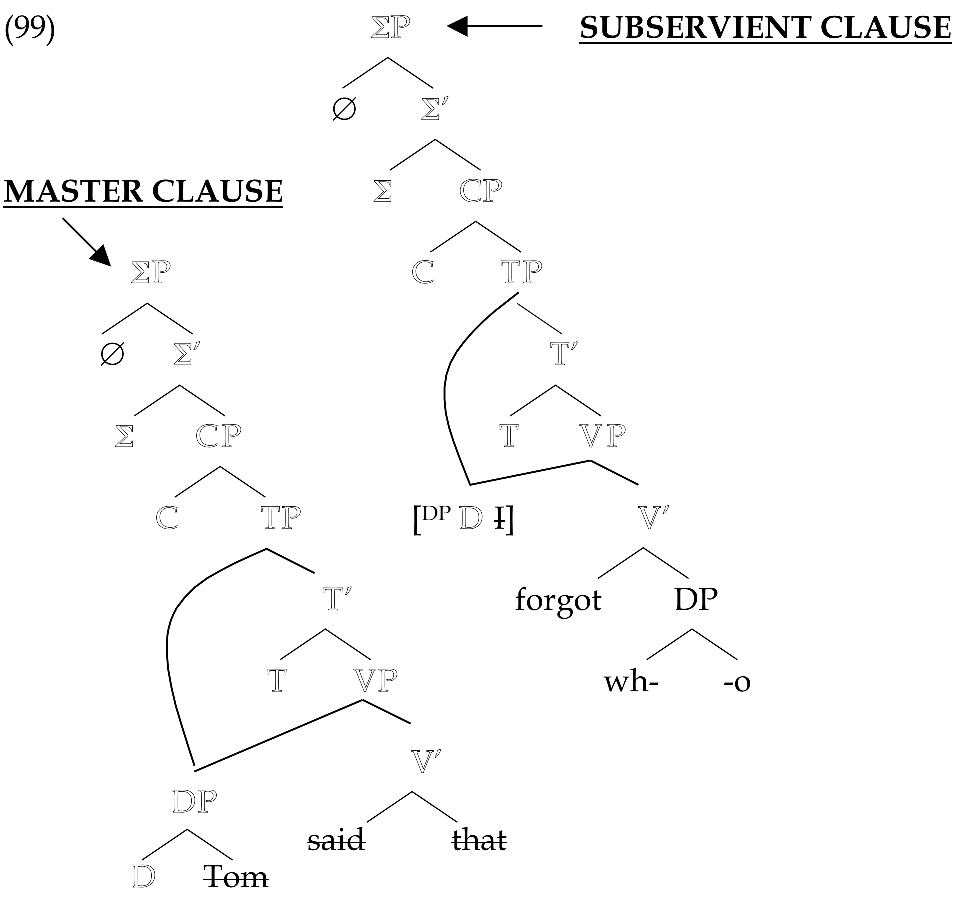
The computational system then zooms into Ψ (cf. (93)) and begins to build the subservient clause. The non-shared lexical tokens in Ψ are combined one by one, in a top-to-bottom fashion, up to the point illustrated in (98).

(98)



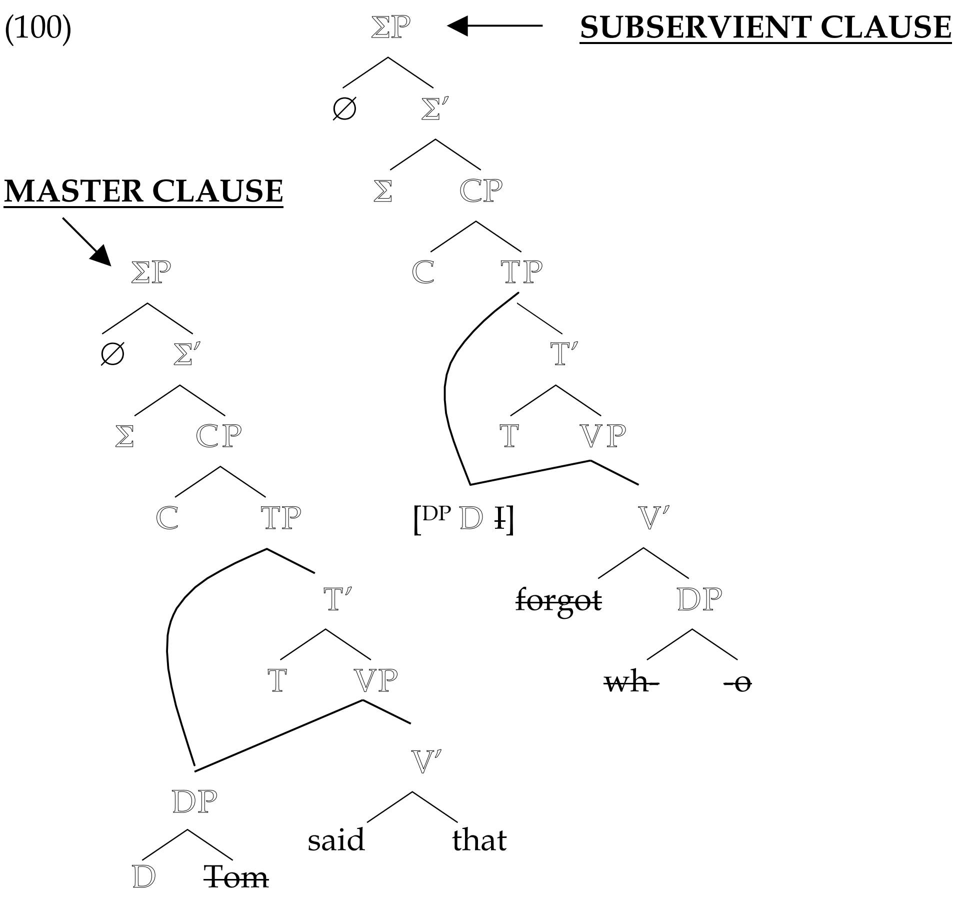
At this point, the WH-phase **who** is built from lexical tokens shared by Ψ and Δ, which are tucked in at the bottom of the subservient clause, giving rise to (99).

(99)



The next necessary step is to introduce $C_{[+WH]}$ and tuck it in at the bottom of the subservient clause, as a (temporary) sister to **who**, so that the relevant checking of (WH and EPP) features can take place. But, before that, spell-out needs to apply in order to guarantee LCA satisfaction. This is indicated in (100), whose corresponding PF representation is as in (101).

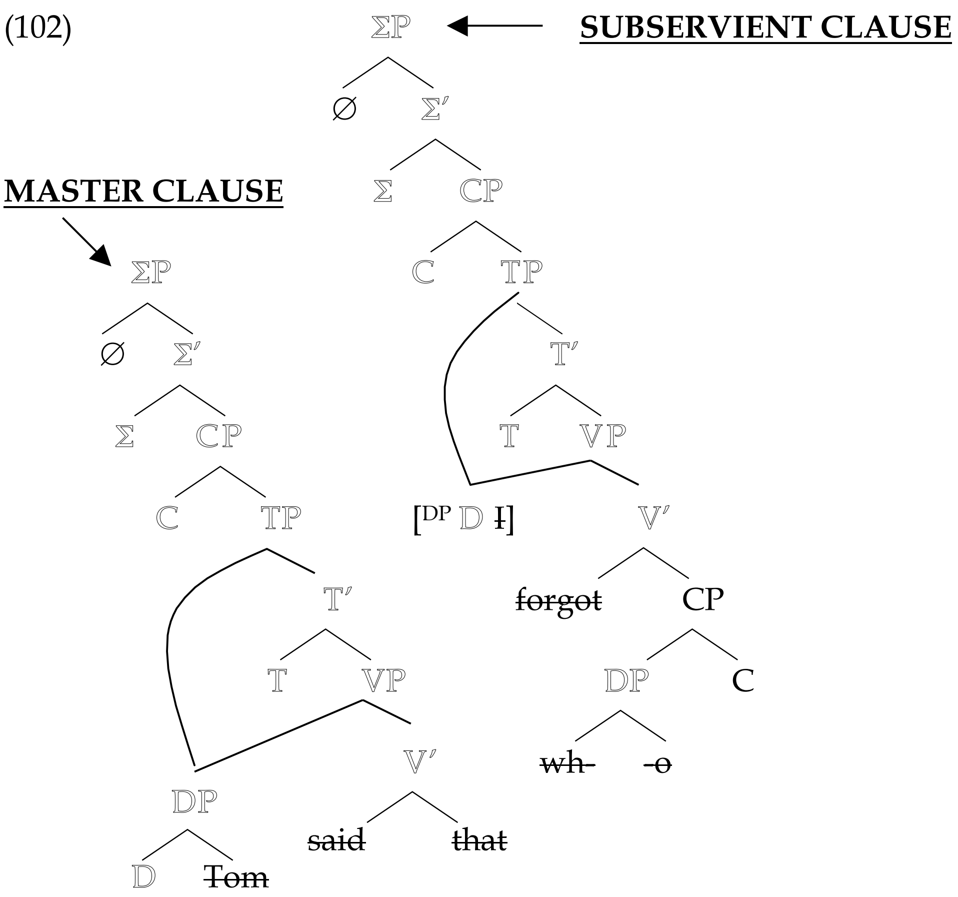
(100)



(101) [#Tom#]$^{\cap}$[#said#$^{\cap}$#that#]$^{\cap}$[#I#]$^{\cap}$[#forgot#$^{\cap}$#who#]

Once that is done, then the [+WH] complementizer is finally merged inside the **V′** in the subservient clause, as a sister to the WH-phrase, so that all relevant feature checking operations can take place. This is indicated in (102).

(102)

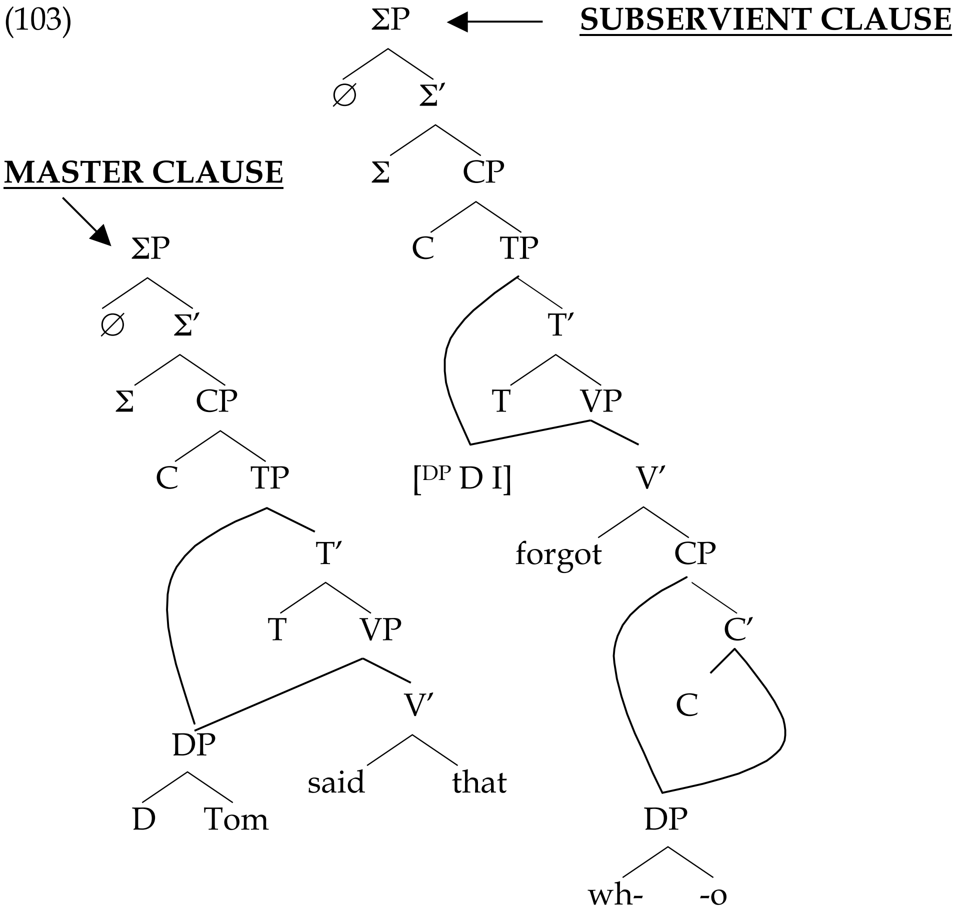


At this point, the lowest complementizer of the subservient clause is supposed to take the embedded TP of the master clause as its complement. However, there is no such TP there yet.

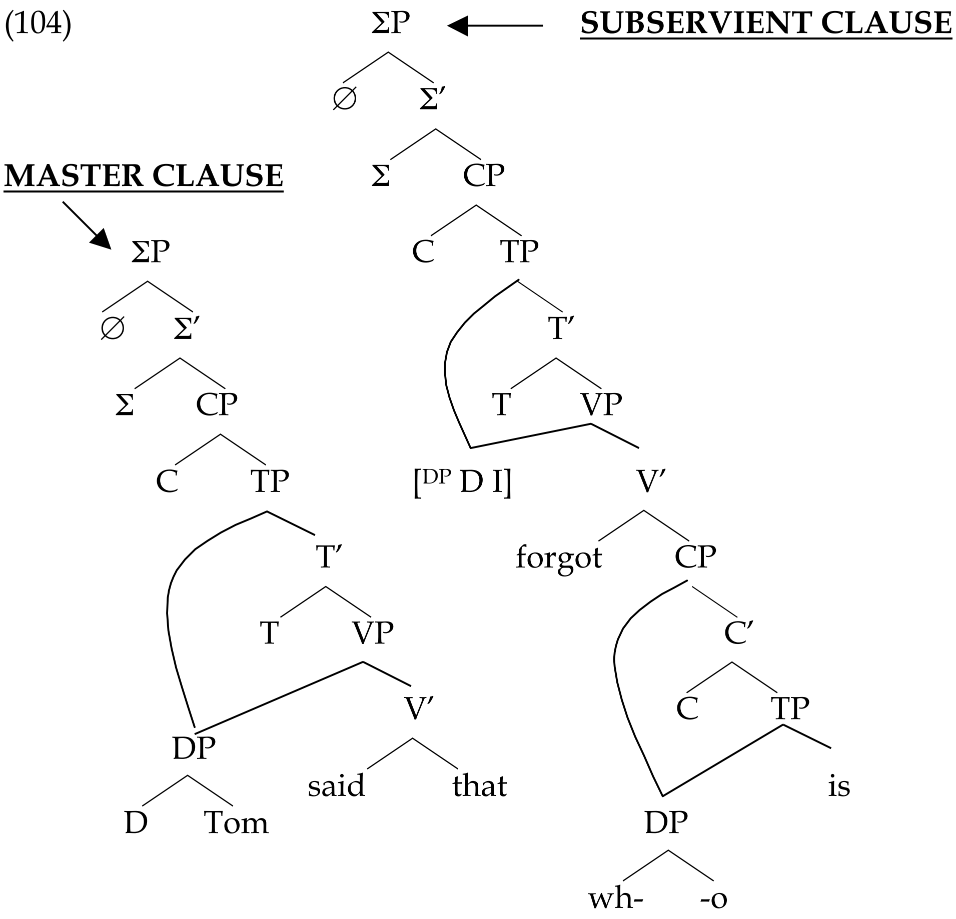
The only alternative left is to build the embedded TP from scratch, from the lexical tokens present in the numeration Ψ (precisely the same lexical tokens shared by numeration Δ). This would be done in the usual fashion, tucking in the relevant lexical tokens one by one at the bottom of the spine of the tree, going from the top downwards and applying spell-out whenever (and only when) necessary for PF reasons.

The reminder of the derivation would be as shown in the steps from (103) to (110) below. For expository reasons, indicators of spell-out are omitted from the notation.

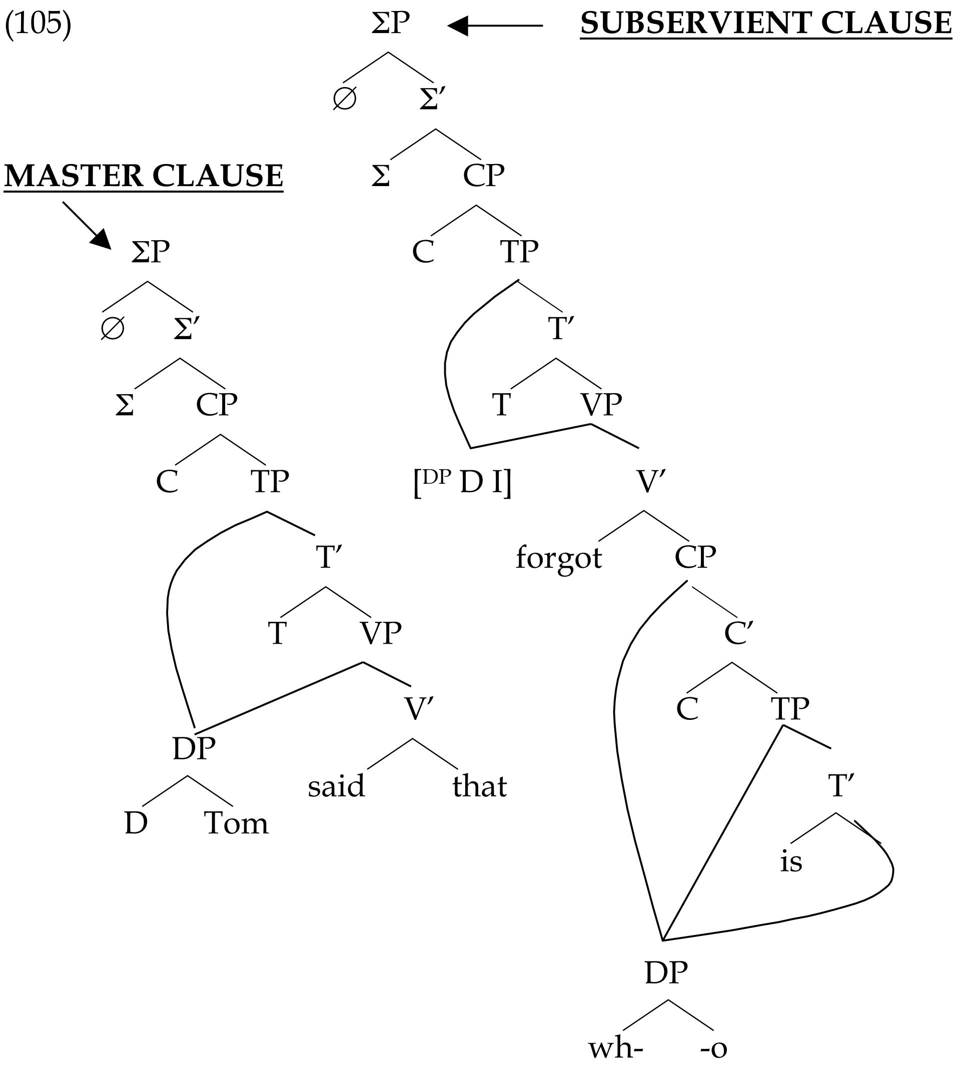
(103)



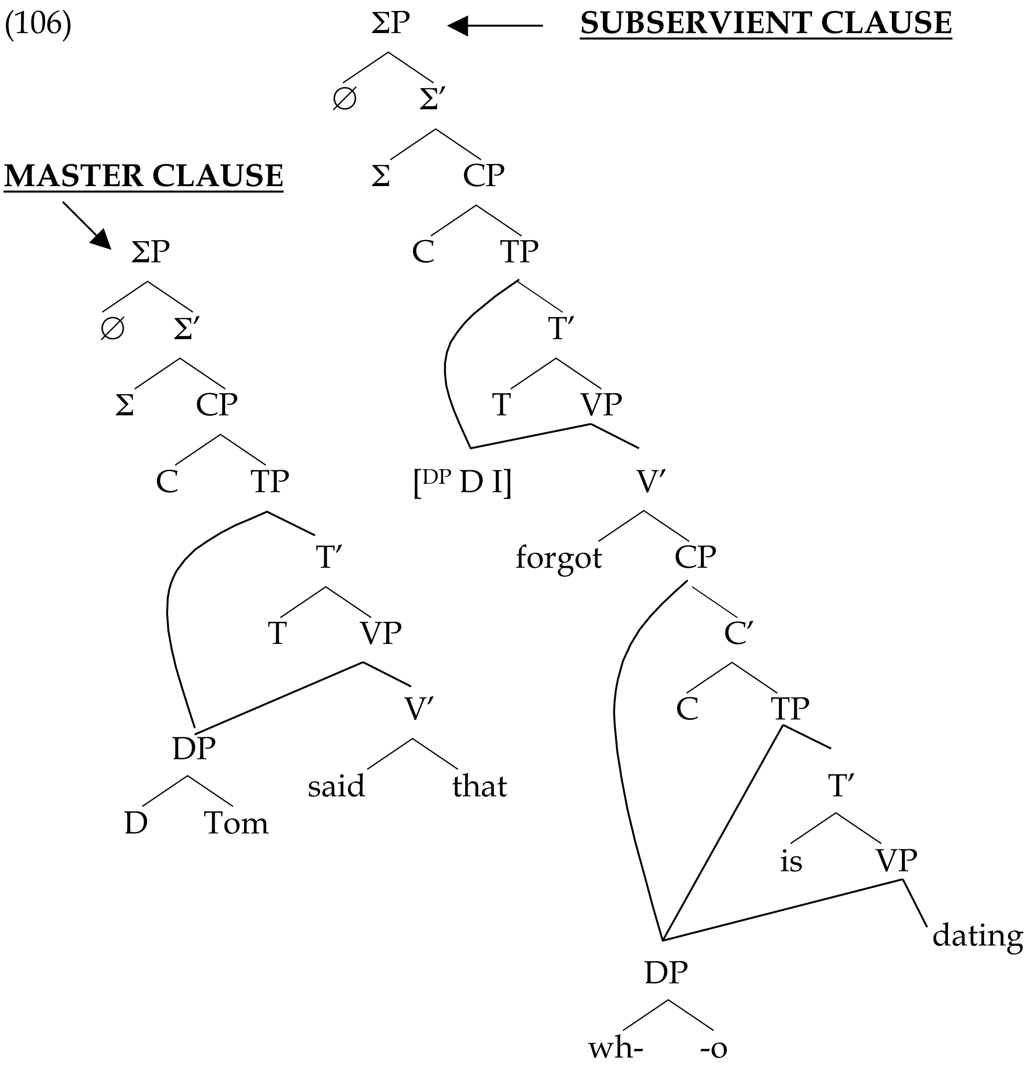
(104)


ΣP
SUBSERVIENT CLAUSE
∅
Σ′
Σ
CP
C
TP
T′
T
VP
[DP D I]
V′
forgot
CP
C′
C
TP
is
DP
wh-
-o
MASTER CLAUSE
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
said
that
DP
D
Tom

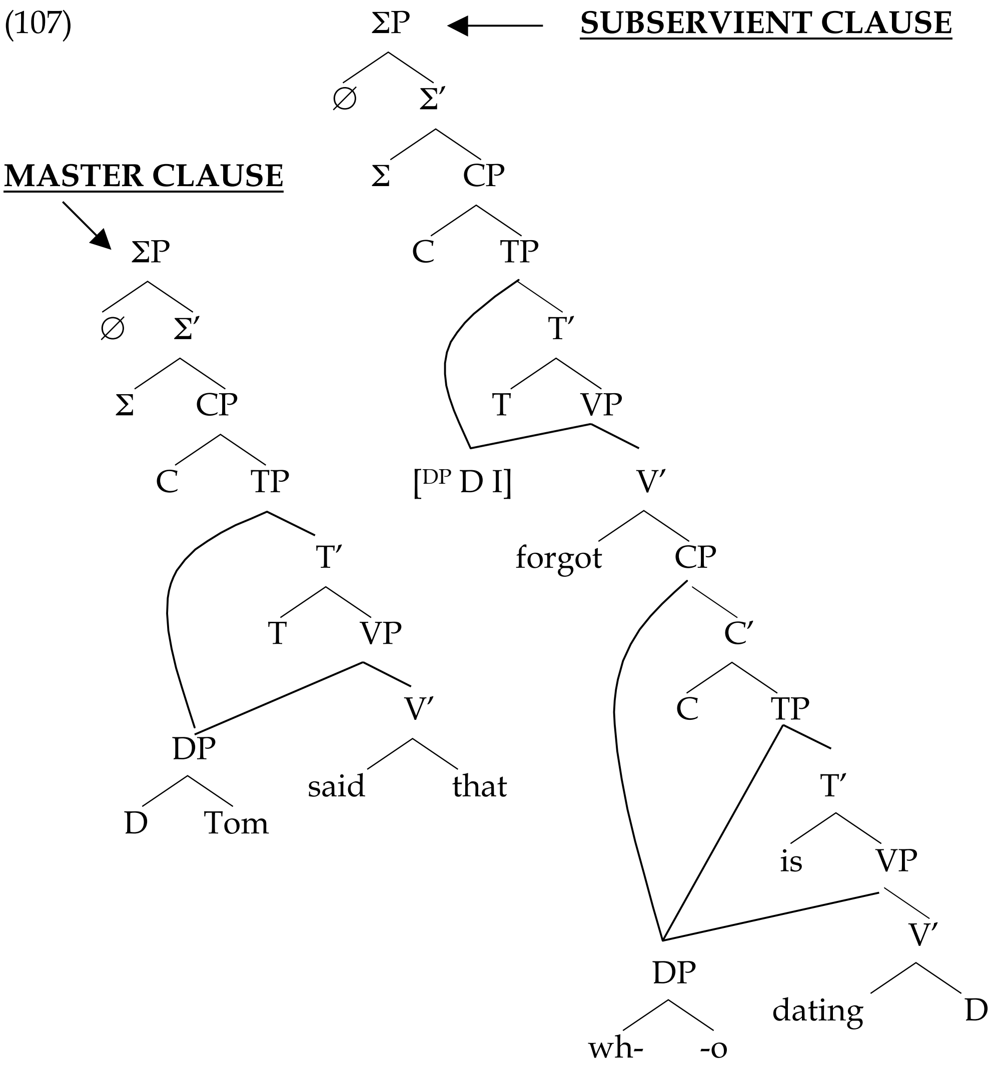
(105)


ΣP
SUBSERVIENT CLAUSE
∅
Σ′
MASTER CLAUSE
Σ
CP
ΣP
C
TP
∅
Σ′
T′
Σ
CP
T
VP
C
TP
[DP D I]
V′
T′
forgot
CP
T
VP
C′
V′
C
TP
DP
said
that
T′
D
Tom
is
DP
wh-
-o

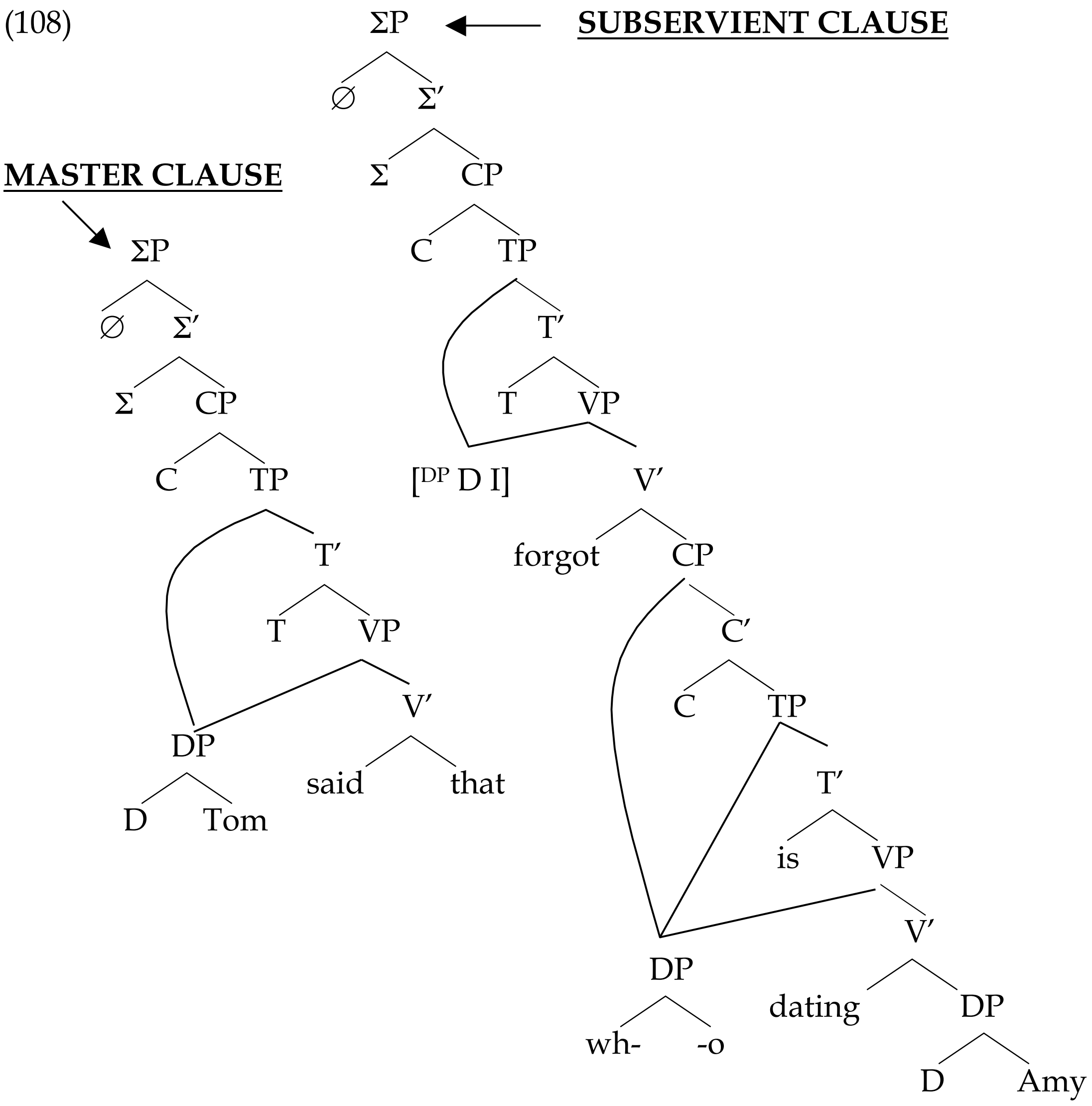
(106)


ΣP
SUBSERVIENT CLAUSE
∅
Σ′
Σ
CP
MASTER CLAUSE
C
TP
ΣP
∅
Σ′
T′
Σ
CP
T
VP
C
TP
[DP D I]
V′
T′
forgot
CP
T
VP
C′
V′
C
TP
DP
said
that
T′
D
Tom
is
VP
dating
DP
wh-
-o

(107)


ΣP
SUBSERVIENT CLAUSE
∅
Σ′
Σ
CP
C
TP
T′
T
VP
[DP D I]
V′
forgot
CP
C′
C
TP
T′
is
VP
V′
DP
wh-
-o
dating
D
MASTER CLAUSE
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
Tom
said
that

(108)



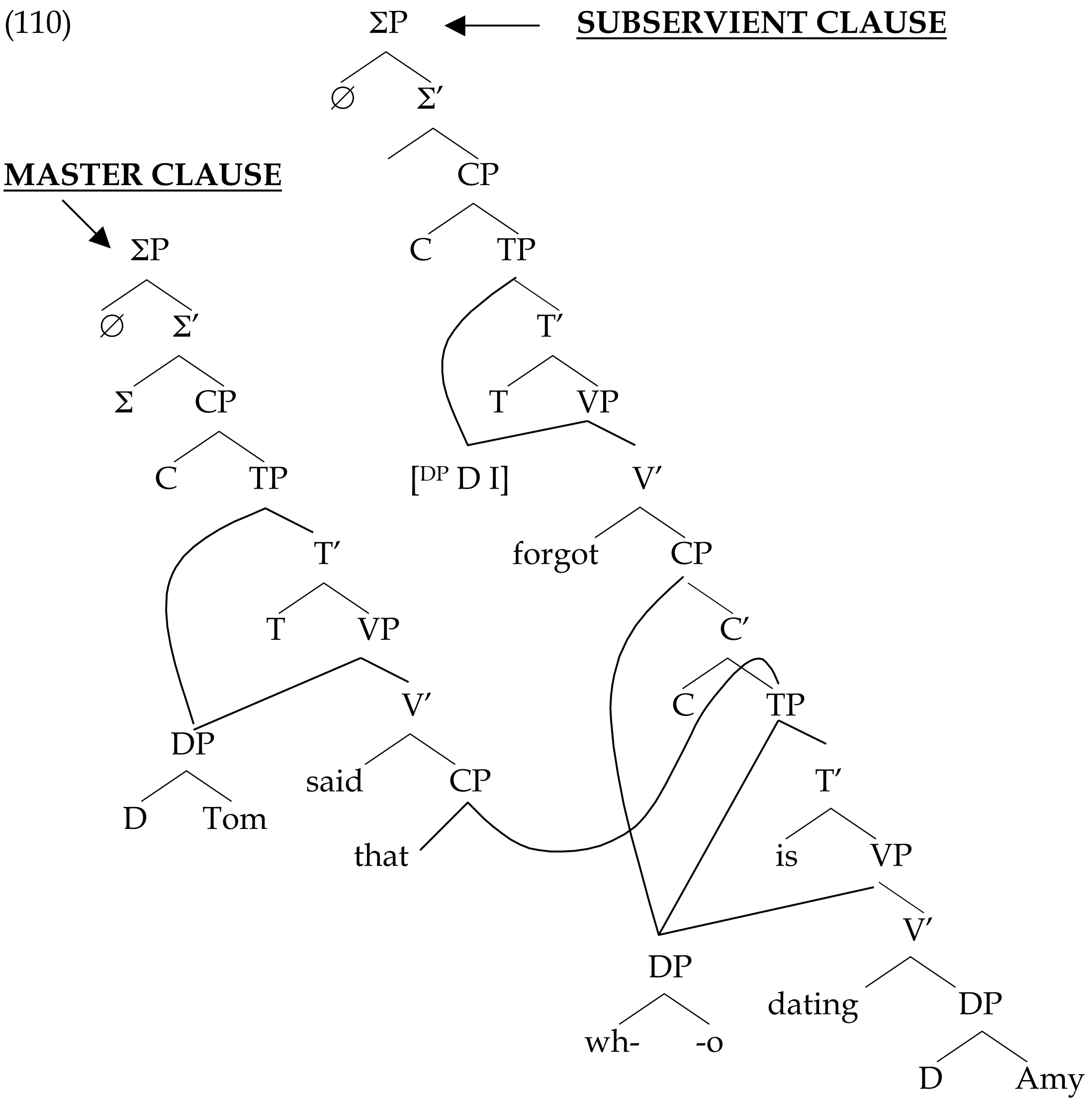
At this point, the construction of the subservient clause is over. The corresponding PF representation so far is as in (109).

(109) [#Tom#]$^{\cap}$[#said#$^{\cap}$#that#]$^{\cap}$[#I#]$^{\cap}$[#forgot#$^{\cap}$#who#]$^{\cap}$[#is#$^{\cap}$#dating#$^{\cap}$#Amy#]

Notice, however, that the master clause still lacks an embedded clause, which, given Δ, is supposed to be a shared constituent, namely: the lower TP embedded inside the subservient clause.

As it stands, the master clause in (108) is not convergent. What needs to be done in order to fix that structure is to make the lower TP of the subservient clause a shared constituent, so that, aside from being a sister to the lowest C of the subservient clause, it becomes also a sister to the lowest C of the master clause. At some point during the computation of the subservient clause, the system has to somehow take that TP (whether it is complete or not) and tuck it in at the bottom of the master clause, as a sister to the lowest [-WH] complementizer **that**. Eventually, the resulting global structure would be the Siamese Tree configuration in (110).

(110)



However, taking such derivational step is impossible. In order to do that, the system would have to be able to go back and fourth between subcomputations. As discussed in **§IV.3.4**, there is an inherent asymmetry in the way overlapping computations interact. Once a given derivational round is over, there is no way back. A constituent built in a terminated derivational round can still be remerged inside a phrase in the active derivational workspace (as long as the former (vacuously) c-commands the later), but cannot have anything being remerged inside it while it still remains in the inactive derivational workspace.

In the hypothetical computation above, the problem lies in the higher V′ constituent of the master clause. At the point when the derivational round for the master clause terminates, the daughters of that V′ are **said** and **that**. Later on, the lower TP of the subservient clause is remerged inside that same V′ as the new sister to **that** (thereby creating a CP that is the new mother of the shared TP and the new daughter of that V′ in question). When that happens, the V′ was crucially not in the active derivational workspace.

Therefore, in the end, there is no constraint on invasion at subject position as a primitive notion. In a heavily dynamic system where derivations proceed in a 'generalized tucking-in' fashion, every subject is introduced before its corresponding T. It follows, then, that the corresponding TP cannot possibly be built early enough for it to be shared, since structure-sharing is inherently asymmetric, with master clauses feeding subservient clauses, but not the other way around.

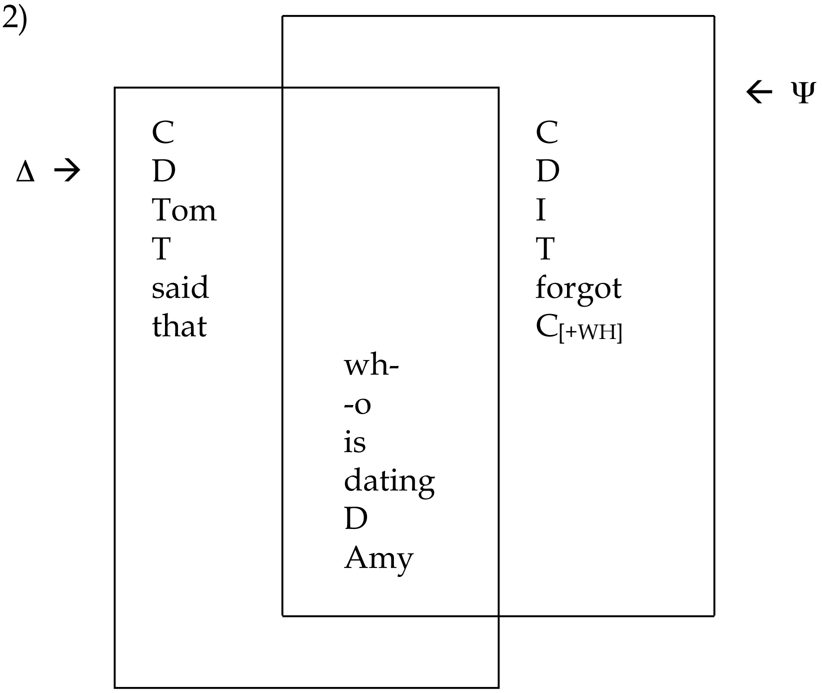
Consider now the example in (111), which is not a possible syntactic amalgam.

(111) * Tom said that who I forgot is dating Amy.

At first sight, it may seem that the system proposed here could potentially overgenerate cases like this, where the WH-phrase is introduced early, still in the derivational round corresponding to the master clause, and then shared later on. Let us take a closer look at the relevant derivations, then, and appreciate how the ungrammaticality of (111) is indeed predicted without further stipulation.

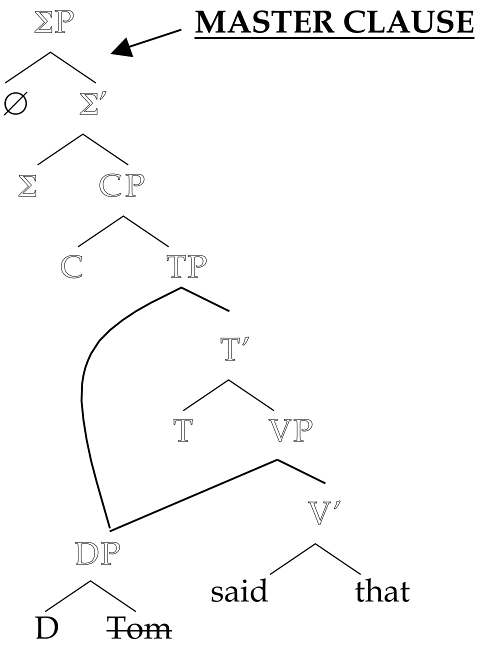
Starting from the same intersecting numerations — repeated below as (112) —, consider the following derivation for (111).

(112)



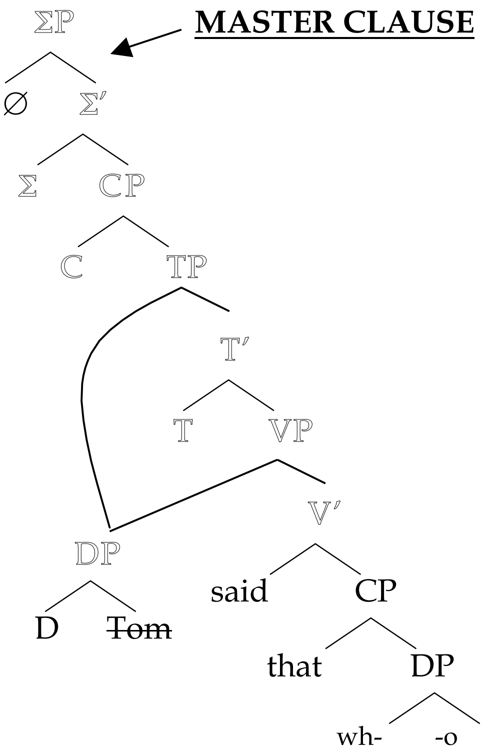
First, the computational system begin to build the master clause. The lexical tokens in the non-intersecting portion of Δ get combined one by one, in a top-to-bottom fashion, up to the point when the structure in (113) obtains.

(113)



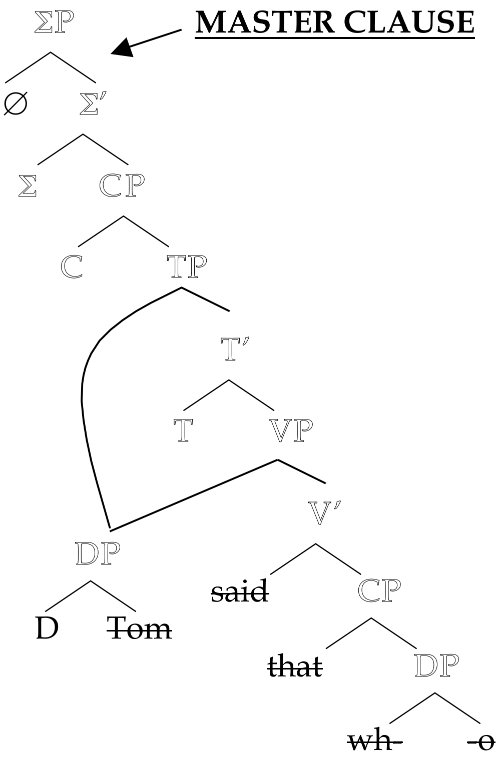
Then the WH-phase **who** is built (in a step-by-step fashion) at the very bottom of the spine, as a temporary sister to the complementizer **that**, yielding (114).

(114)



Then spell-out applies, and the structure in (115) obtains. The corresponding PF representation so far is as in (116).

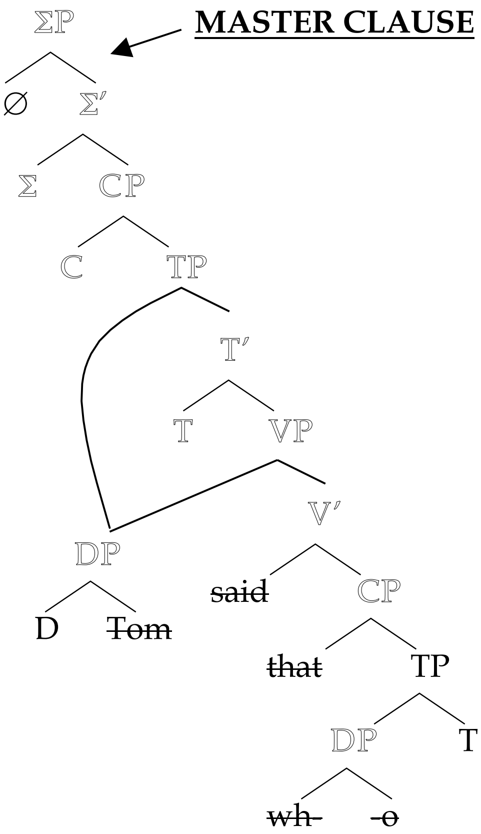
(115)



(116) [#Tom#]$^\cap$[#said#$^\cap$#that#$^\cap$#who#]

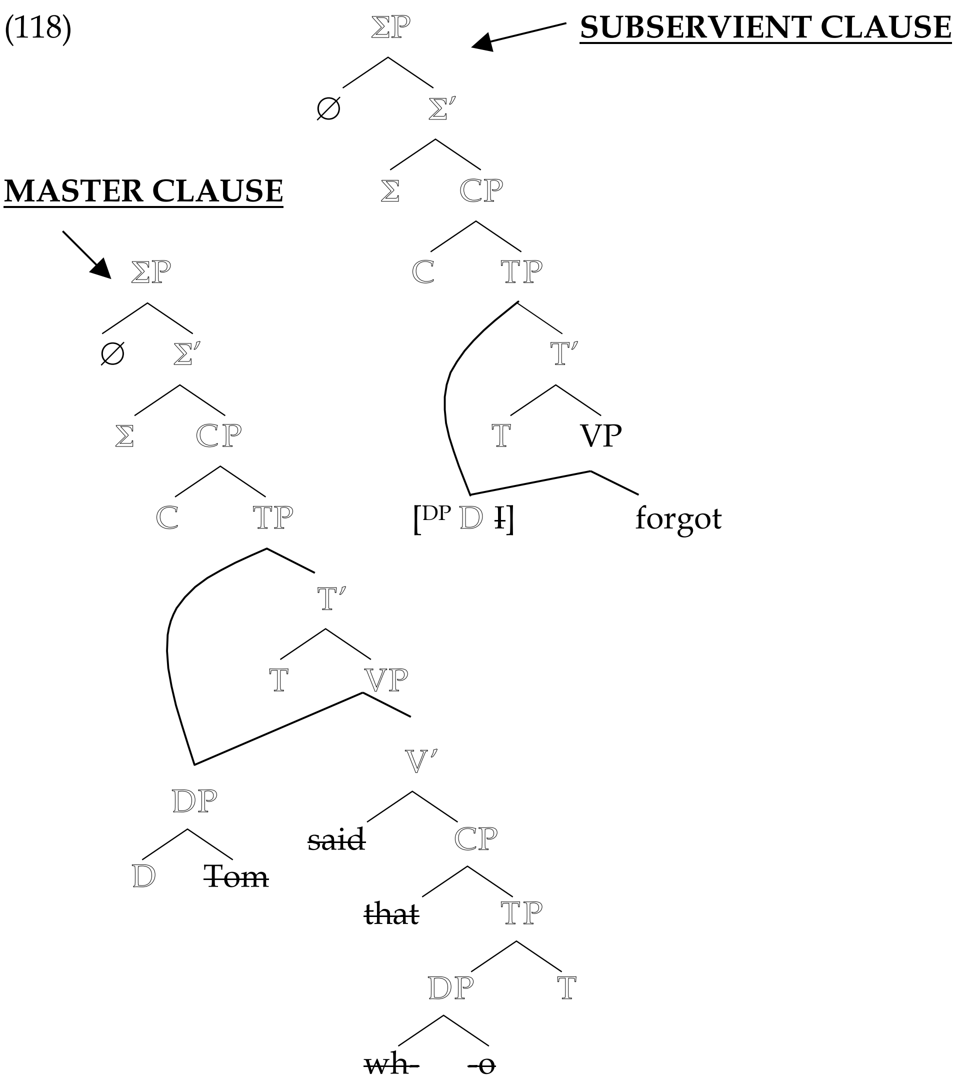
At the next step, the T head of the embedded clause is tucked in at the bottom of the phrase marker as a sister to **<u>who</u>**, as in (117), which guarantees that there will be a TP to be shared in the next derivational round.

(117)



This structure is then spelled-out, and the derivational round of the master clause terminates. The derivational round of the subservient clause begins, and computational system then builds another phrase marker in parallel, combining the lexical tokens of Ψ in the usual 'generalized tucking-in' fashion, starting from the non-shared portion of the numeration, up to the point where (118) obtains.

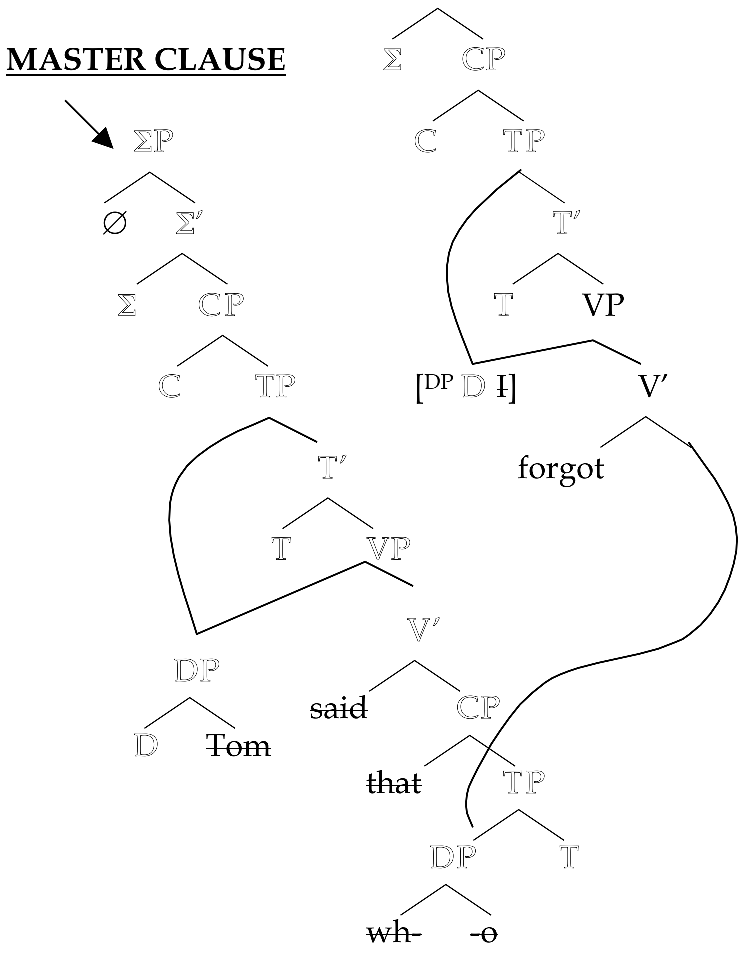
(118)



Still within the derivational round of the subservient clause, the system can take the WH-phrase **<u>who</u>** and tuck it in at the bottom of the spine of the subservient clause, as a temporary complement to the verb **<u>forgot</u>** (cf. (119)), so that it can subsequently become the specifier of the [+WH] complementizer that is about to be introduced in the following step.

(119)

ΣP
SUBSERVIENT CLAUSE
∅
Σ′



This is an illegitimate step, however. In order for any element to be (re)merged inside a phrase, it must c-command its future sister (cf. **§IV.3**). Notice that, in the input structure in (118) above, the WH-phase **<u>who</u>** is dominated by a TP whose head is a member of the reference set (i.e. Ψ) for that derivational round. Therefore, that TP is visible for the purposes of figuring out whether **<u>who</u>** (vacuously) c-commands **<u>forgot</u>**. Since that TP dominates **<u>who</u>** but does not dominate **<u>forgot</u>**, it follows that **<u>who</u>** fails to (vacuously) c-command **<u>forgot</u>**.

Therefore, the derivation is ruled out at this point, even before the [+WH] complementizer is introduced so that the TP gets the chance to be shared.

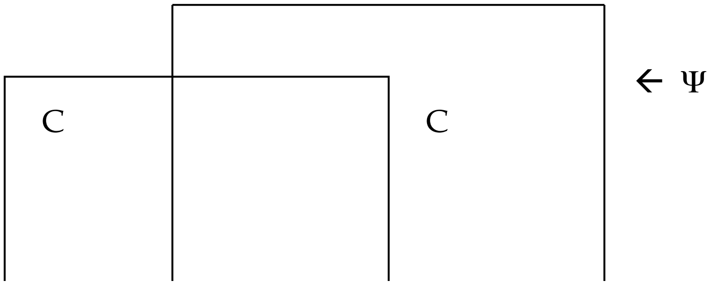
V.7.2 Non-Finite Clauses.

However, as previously mentioned in **§II.7**, it is possible for clause invasion to target a subject position in ECM constructions, as shown in (120).

(120) The boss wants *you'll never guess* ***which employee*** to do that job.

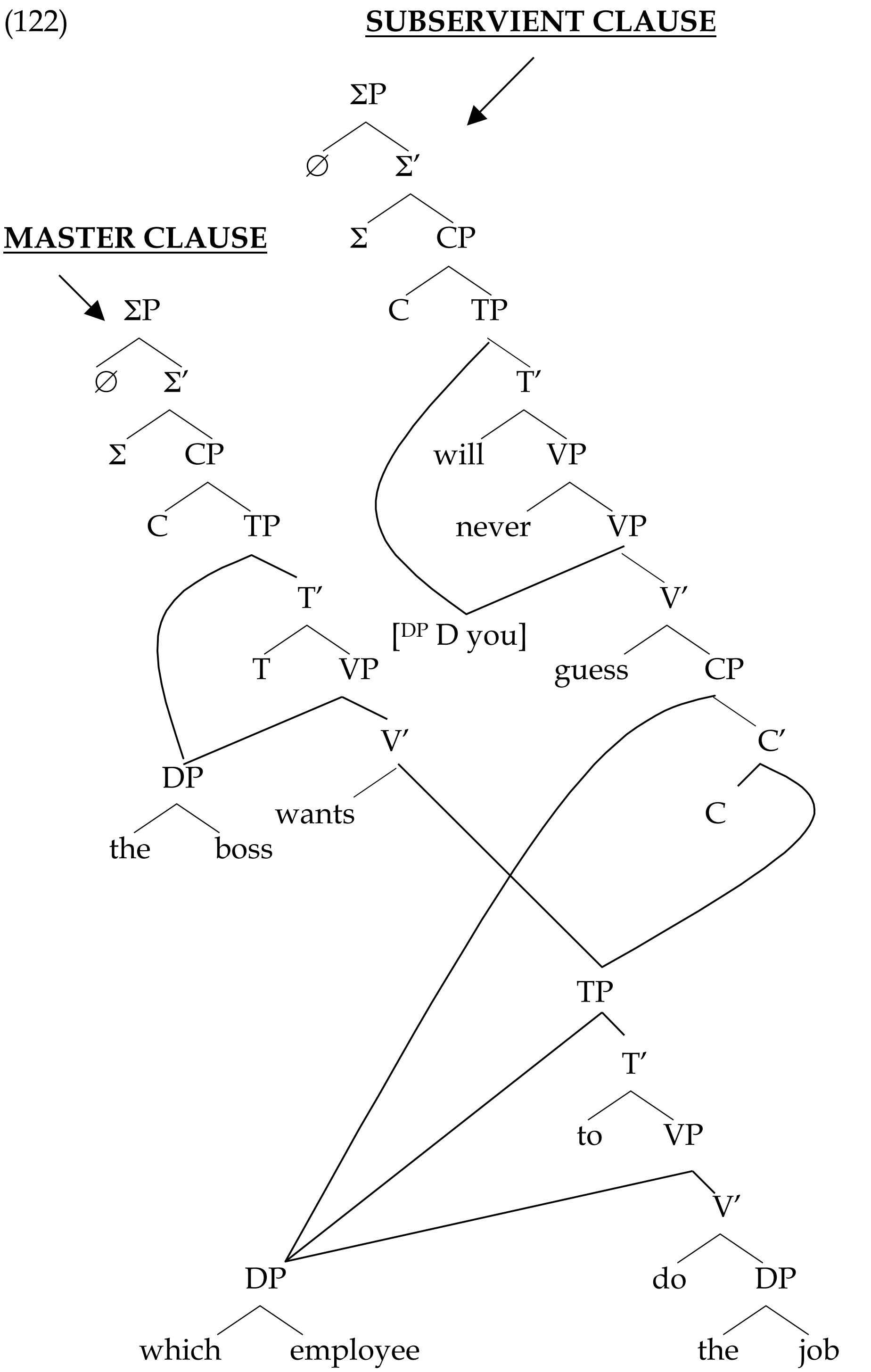
These cases seem to pose a problem for the analyses just presented. By standard assumptions, the subject position at issue would be the spec/TP of the (shared) embedded clause. The corresponding input would be the intersecting numerations in (121), and corresponding representation would be as in (122).

(121)



Δ →	the		D
	boss		you
	T		will
	wants		forgot
			never
		which	$\mathrm{C}_{[+\mathrm{WH}]}$
		employee	
		to	
		do	
		the	
		job	

(122)



If this is the case, we would expect examples like (120) to be as unacceptable as the ones like (89). After all, the relevant parts of the structure are identical.[155] The representation in (122) should be impossible to be derived for the same reasons that the one in (110) is. If the head of the shared TP (i.e. **to**) is introduced early, still in the derivational round where the master clause is built, then the WH subject **which employee** would later fail to c-command **guess** (by virtue of being dominated by the shared TP), so that it could be tucked in at a position where it would eventually end up as the spec/CP and undergo the relevant feature-checking process associated with WH-phrases and [+WH] complementizers. On the other hand, if the introduction of the head of the shared TP (i.e. **to**) is delayed until the derivational round where the subservient clause is built, then it would be too late for the embedded TP to be shared, given the asymmetry inherent to overlapping computations. Thus, it seems that the analysis so far undergenerates, making the wrong prediction that examples like (120) should not be possible.[156]

[155] One could say that the Siamese-Tree structure in (122) is expected to be ungrammatical even on purely representational grounds. If we focus on the subservient clause (cf. (i) below), we detect that it has a WH-chain without case, due to the fact that neither is the matrix verb (i.e. **guess**) associated with the relevant structure that would assign accusative case to **which employee**, nor is the embedded non-finite T able to assign nominative case to **which employee**.

(i) [^{CP} [^{IP} [^{DP} you]$_2$ will [^{VP} never t$_2$ guess [^{CP} [^{DP} which employee]$_1$ [^{IP} t$_1$ to [^{VP} do [^{DP} the job]]]]]]]

That being the case, the problem mentioned above becomes even worse, as there would be one other thing forcing us to wrongly predict accepted examples to be ungrammatical. Notice however, that **which employee** arguably does get case in the domain of the master clause by whatever mechanism is ultimately responsible to assigning case to embedded subjects in ECM constructions. Since there is nothing wrong with the DP **which employee** itself (its case feature is indeed checked somewhere), and since it is the very same token of the DP **which employee** that participates in both sides of the Siamese-Tree structure, there is no *a priori* reason why there should be any case-related problem with the WH-chain in the subservient clause.

[156] In principle, one could hypothesize that what makes invasion at the subject position in ECM constructions possible is something related to the fact that those subjects have a special status with regards to case, as they are related to a case assigner in the matrix domain of the master clause (arguably the head of a specific functional projection (e.g. AgrOP, *v*P, AccP) right above VP, ommited from the notation in (122) for expository reasons). Thus, there is a real difference that the system could, in principle, piggyback on in order to derive representations like (122).

Interestingly, however, the meaning of (120) — repeated below as (123) — is not compatible with the representation in (122). For instance, (124a) is a possible paraphrase for it, but (124b) is not.

(123) The boss wants *you'll never guess* ***which employee*** to do that job.

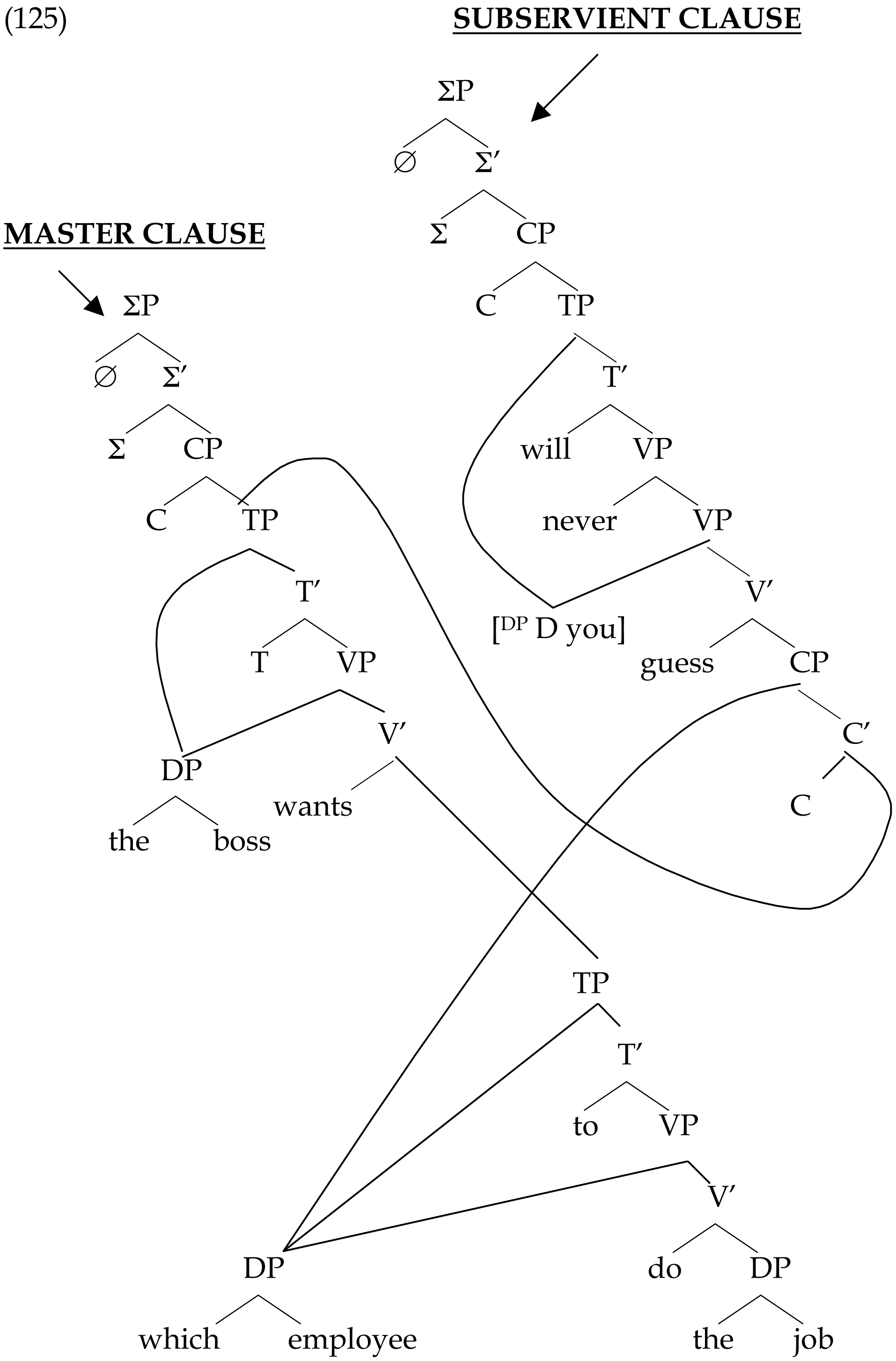
(124) a: You will never guess [which employee]$_1$ the boss wants **t$_1$** to **t$_1$** do the job.

b: # You will never guess [which employee]$_1$ **t$_1$** is the one doing the job.

In other words, what the listener will never guess is the identity of the employee ***x*** such that the boss wants ***x*** to do the job. Therefore, the actual LF representation of (123) should be as in (125), where the shared TP is the highest TP at the master clause.

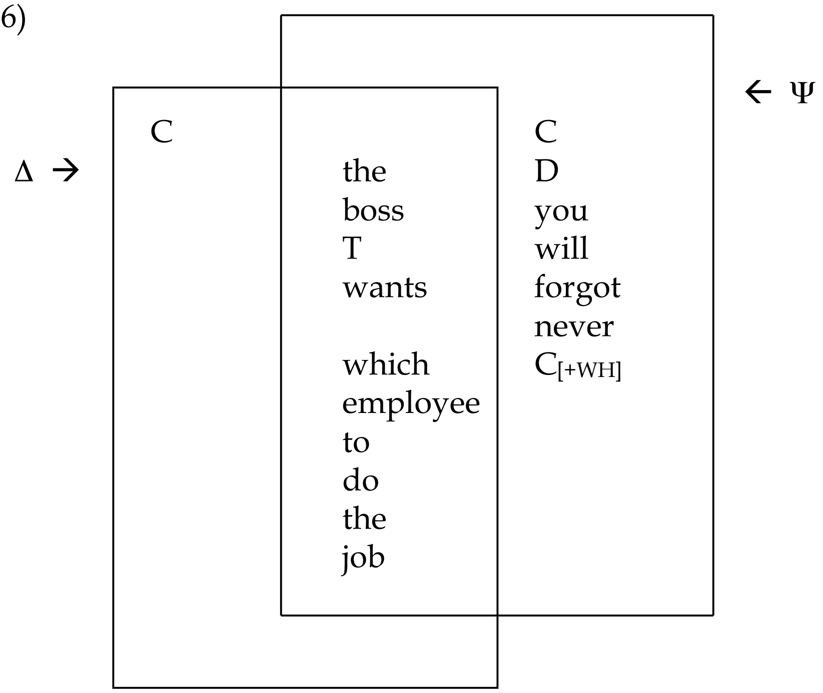
Nevertheless, even if we factor that in, and even if we assume that ECM subjects do occupy the specifier of that relevant functional projection in overt syntax (following Lasnik 1999, 2001; Boskovic 2001), it does not immediately follow that amalgamation at that position should be possible, unless we propose major changes in the theory. This is so because the timing issues discussed above would remain unaltered. It would be still the case that the head of the non-finite T would going to be introduced either too early or too late.

(125)



This representation can indeed be generated by the system proposed here. The starting point would be the intersecting numerations in (126).

(126)

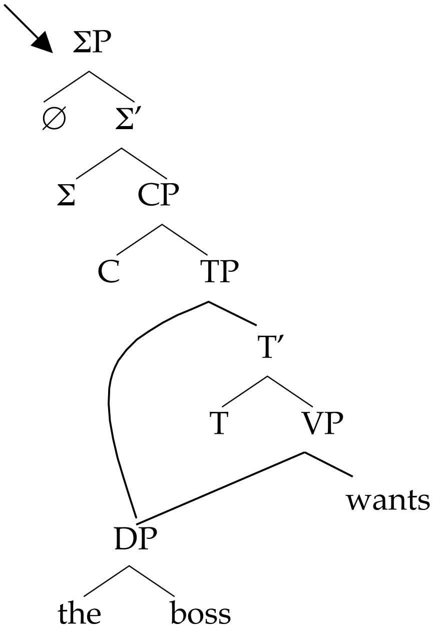


The relevant derivational steps would be as follows (once again, indicators of spell-out have been omitted from the notation for expository reasons).

In the first derivational round, the master clause begins to be built, down to the point where the verb **want** is introduced, as in (127).

(127)

MASTER CLAUSE

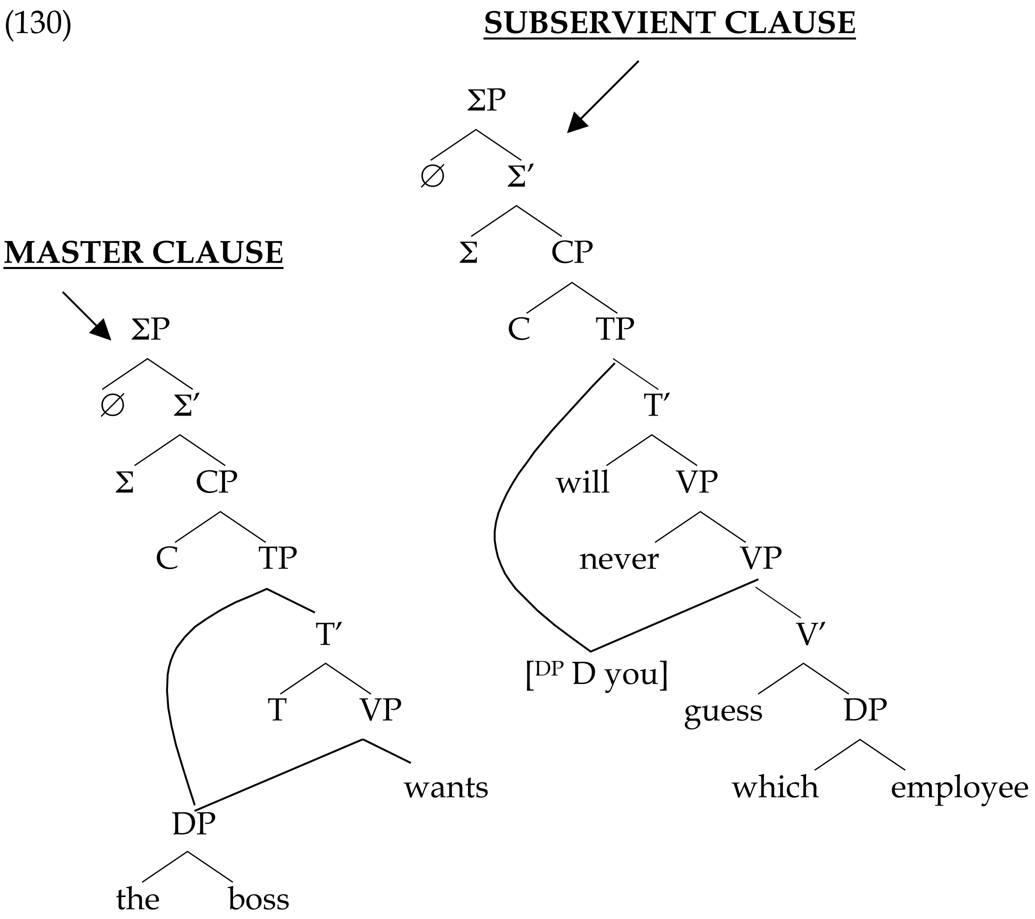


The first derivational round terminates here, with the structure left incomplete, to be finished in the next round. The corresponding PF-string so far is as in (128).

(128) [#the#$^\cap$#boss#]$^\cap$[#wants#]

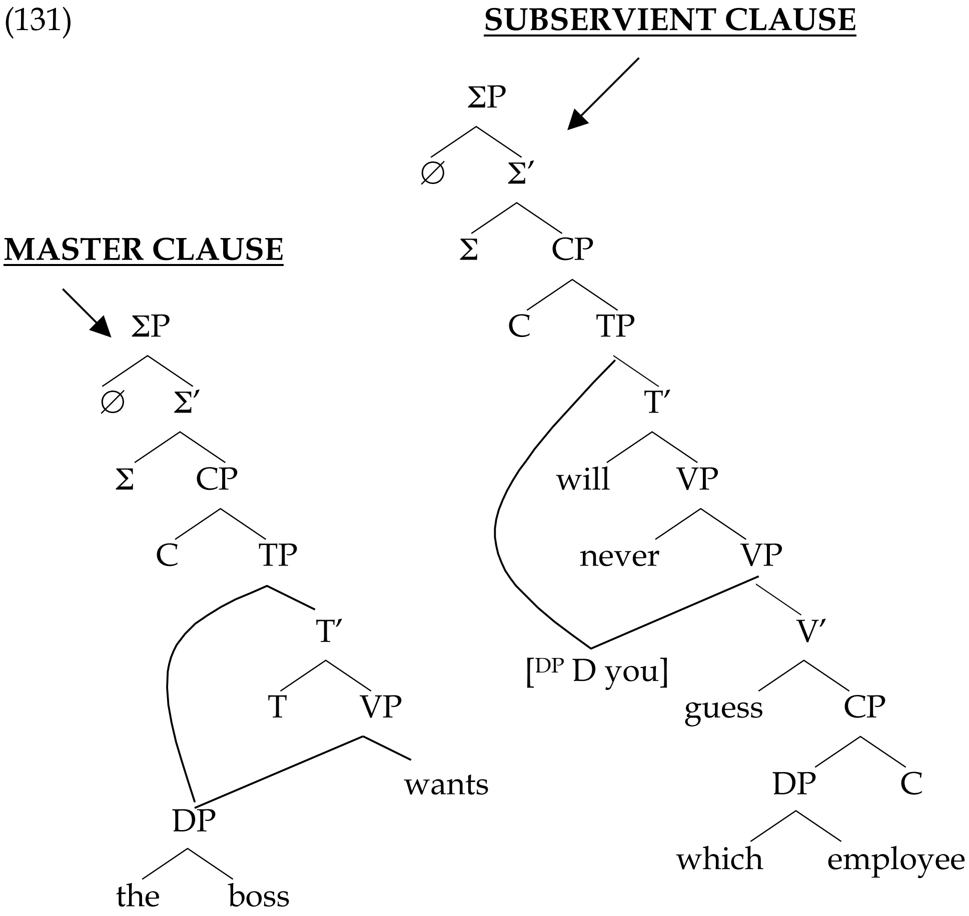
The second derivational round begins, and the subservient clause starts to be built. The first relevant step is the one right after the WH-phrase **which employee** is built as a temporary complement to the matrix verb **guess**, as in (129).

(130)



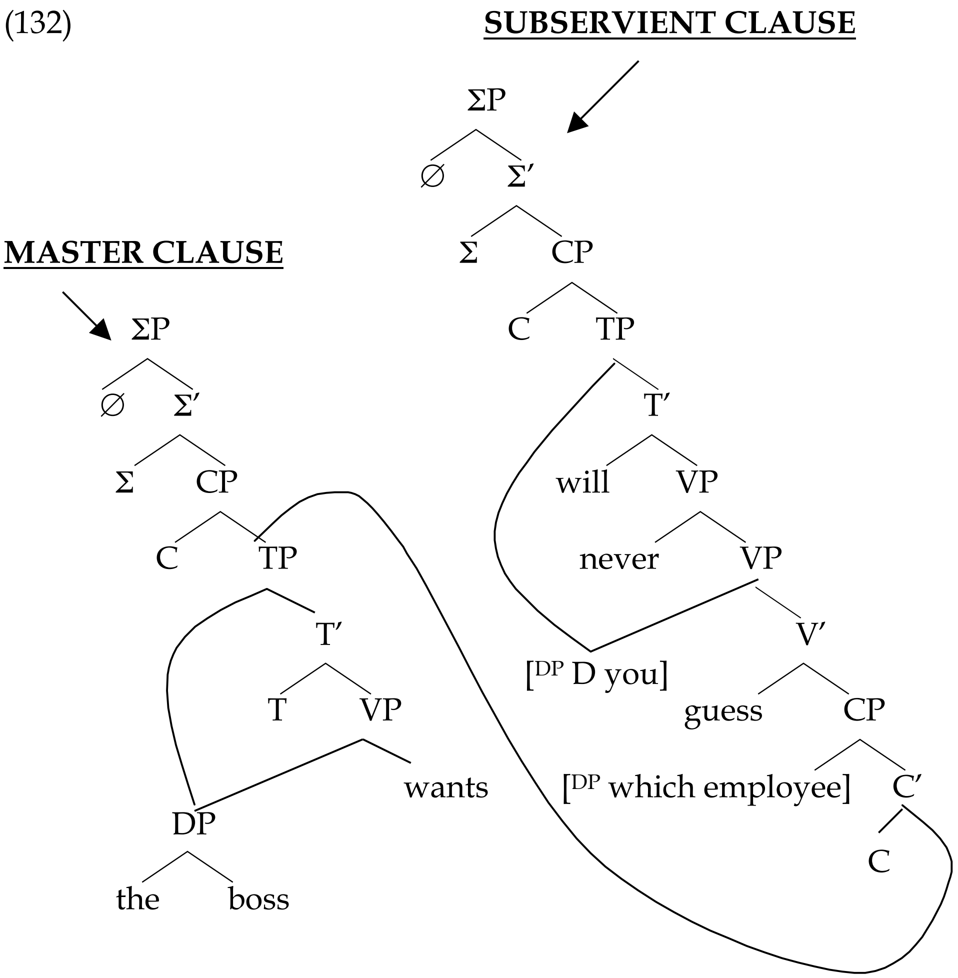
Then, the [+WH] complementizer is merged inside V′ as a temporary sister to **which employee**, as in (131).

(131)



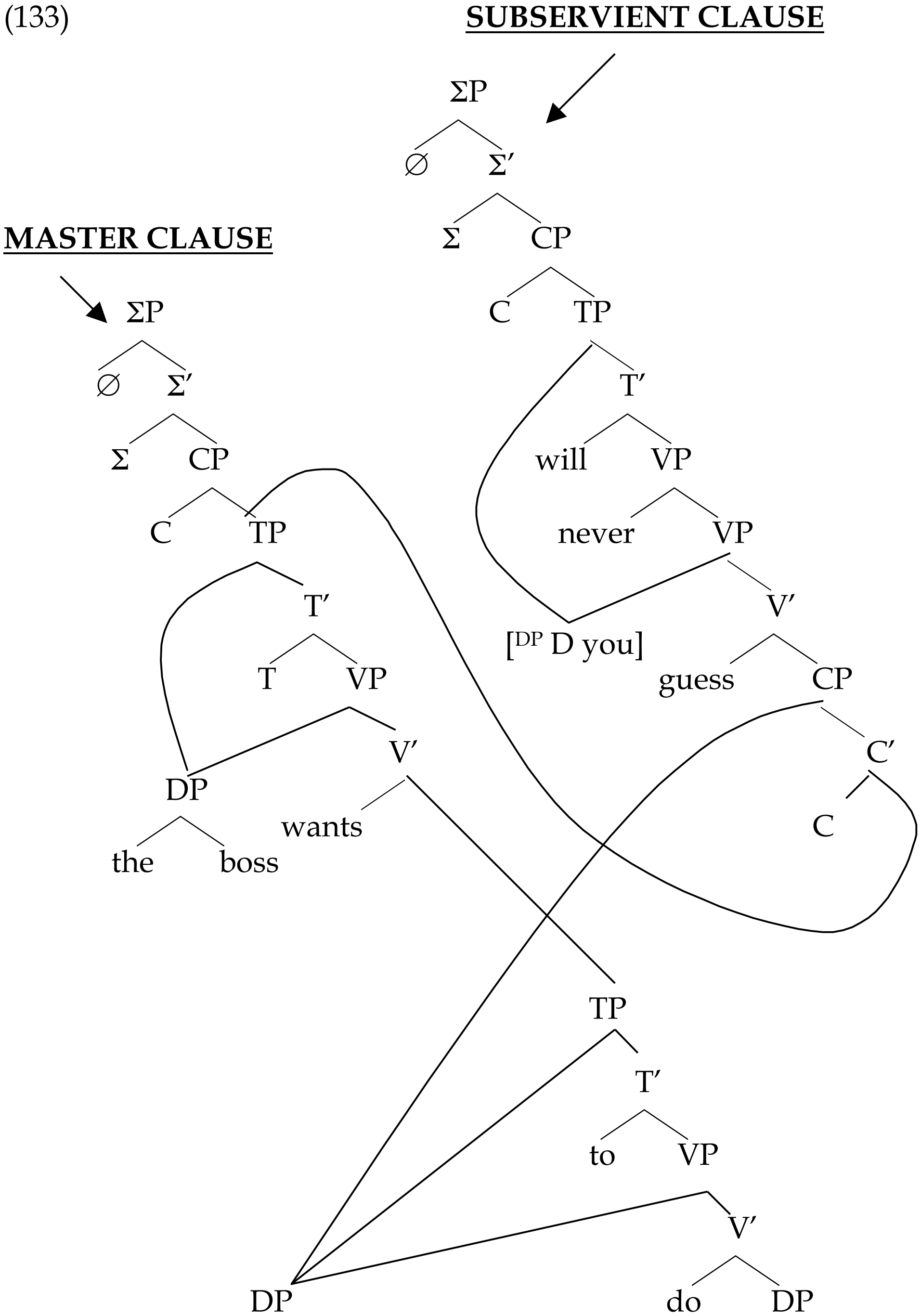
Now comes the crucial step. The matrix TP of the master clause is remerged inside the embedded CP of the subservient clause, as in (132). Notice that, right before that, the TP that is about to be shared (vacuously) c-commands the lowest C of the subservient clause. This is so because nothing that dominates that TP is visible to the subcomputation that builds the subservient clause. So, it is as if there were nothing dominating that TP.

(132)



After that, the derivation proceeds in the usual fashion, from the top downwards, until the whole shared TP is fully built, as in (133), whose corresponding PF-string is as in (134).

(133)



(134)[#the#$^\cap$#boss#]$^\cap$[#wants#]$^\cap$[#which#$^\cap$#employee#]$^\cap$[#to#$^\cap$#do#$^\cap$#the#$^\cap$#job#]

V.7.3. Back to Finite Clauses

The analysis given in the previous section for cases of clause invasion at the subject position in ECM constructions resolved a tension with regards to what seemed to be a contradiction in the paradigm.

However, for better or for worse, that same analysis makes the prediction that the possibility of sharing the highest TP and clause invasion at the position of the subject of the lower TP should in principle be available across the board. Consequently, examples like (89) —repeated below as (135) — are predicted to be derivable by the system here proposed, being possible under the reading corresponding to the paraphrase in (136).

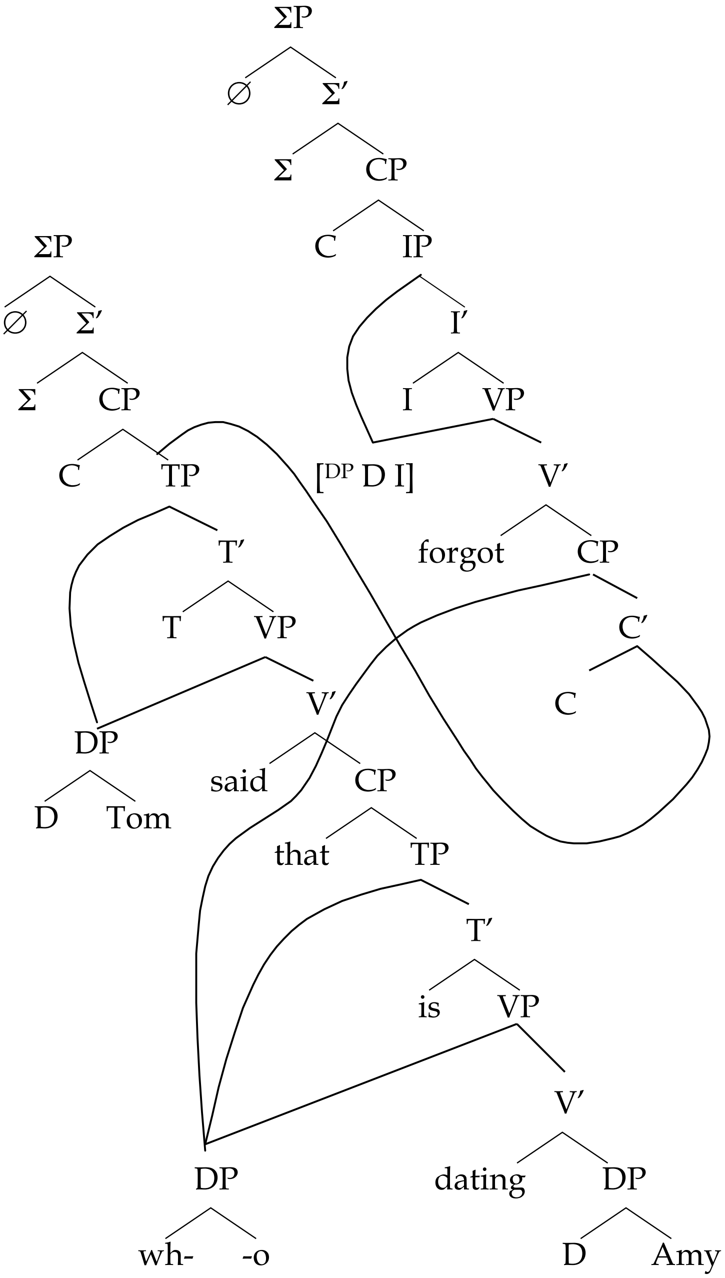
(135) * Tom said that *I forgot **who*** is dating Amy**.**

(136) I forgot ***who***$_1$ Tom said **t$_1$** is dating Amy.

The corresponding LF representation would be as in (137), where it is the higher TP of the master clause (rather than the lower one) that is under the scope of the verb **forgot**. In other words, what is being forgotten is not the identity of the person that is dating Amy. Rather, it is the identity of the person ***x***, such that Tom said that ***x*** is dating Amy.[157]

[157] In such case, this person ***x*** may or may not be the one actually dating Amy.

(137)



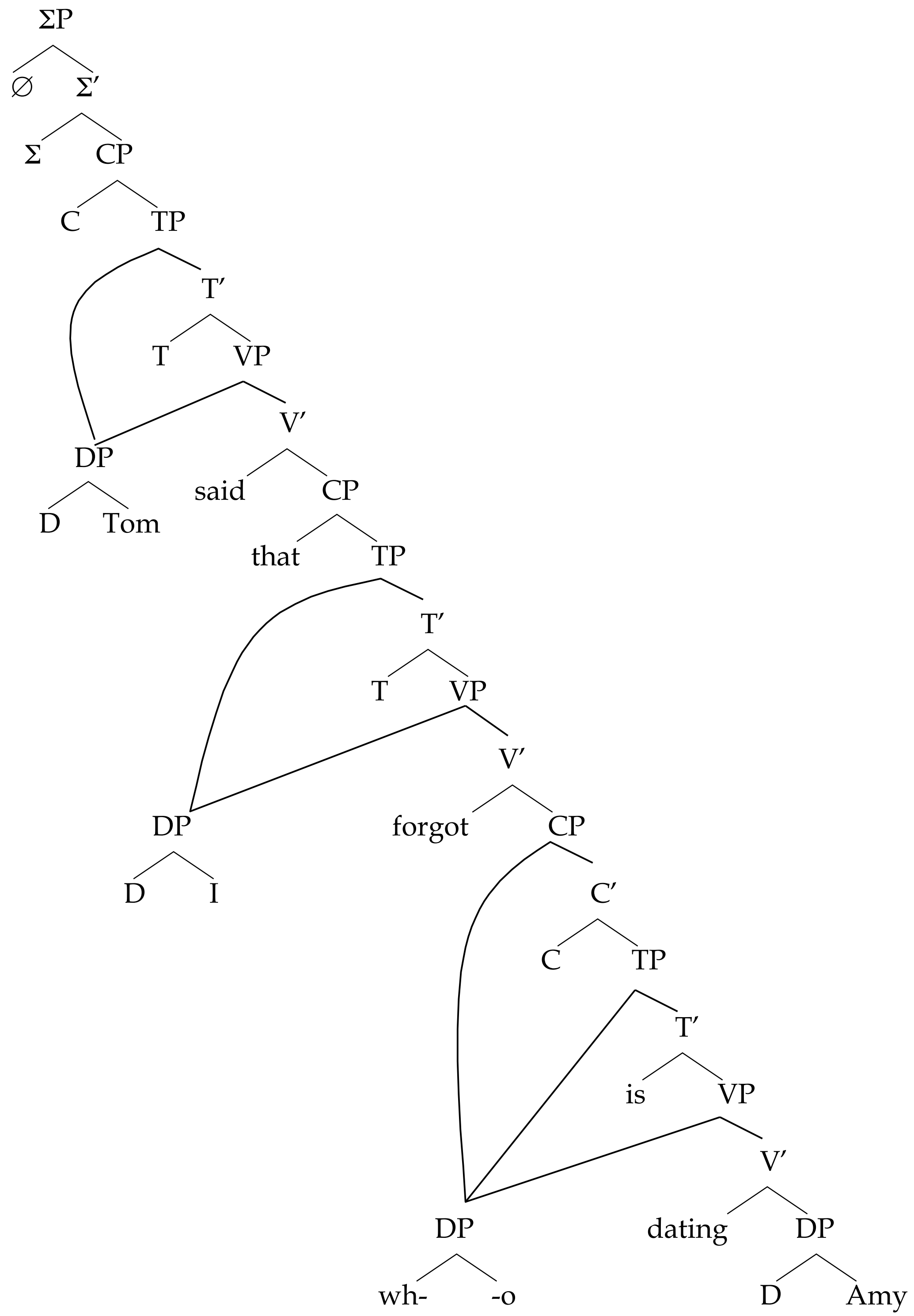
Thus, the fact that the degree of acceptance of examples like (135) is very low is indeed a potential problem for my analysis.

I do not claim to have an ultimate solution to this problem, but the facts seem to strongly indicate that the very low degree of acceptance of examples like (135), under the relevant reading (i.e. (137)) may be an artifact of parsing limitations, something like a garden-path effect.

It is not so much that the string of words in question is not acceptable. The problem is that that very string is fully acceptable under a non-amalgam reading, which would correspond to the structure in (138).[158]

[158] Therefore, in the end, the meaning that I have been taken to be irrelevant turned out to be indirectly relevant, once we performance variables are factored in.

(138)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
Tom
said
CP
that
TP
T′
T
VP
V′
DP
D
I
forgot
CP
C′
C
TP
T′
is
VP
V′
DP
wh-
-o
dating
DP
D
Amy

The strong preference for parsing the string of words in (135) as in (138), rather than as in (137), is not surprising under standard assumptions about sentence processing on the perception side (i.e. notions derivative of *minimal attachment* and *late closure*), especially if we endorse assume a Minimalist approach to sentence processing, such as the one proposed by Weinberg (1999). From that perspective, the task of mapping the string in (135) onto the structure in (138) would involve way less computational complexity than mapping it onto (137), both globally (only one derivational round rather than two) and locally (minimization of spell-out applications at the decision points, crucially, after **said that**[159].

From that perspective, the reason why amalgamated subjects of ECM constructions have a much higher degree of acceptability would be the fact that the point of invasion in those cases is right after an ECM verb, which gives a big hint to the listener that what the finite clause that follows it cannot possibly be its complement, which pretty much reduces the logical possibilities down to analyzing that finite clause as a subservient matrix clause in a Siamase-Tree configuration.

[159] It is standardly assumed in mainstream Minimalism that issues of computational complexity and derivational economy are tied to the notion of 'reference set'. A structure x can only win over a structure y if both x and y are derivable from the same numeration. The two structures in question come each from a distinct input (intersection of numerations). However, on the perception side, there is no *a priori* numeration to begin with. The decisions have to be made locally in terms of the 'current substring', which gets constantly updated leading to constant changes with respect to the logically possible numerations behind that structure being parsed.

Interestingly, some speakers find a slight contrast in acceptability between *bona fide* ECM verbs (e.g. **want**) and hybrid verbs that can be select either a finite or a non-finite complement clause clause (e.g. **believe**).

(139) The boss wants you'll never guess which employee to do the job.

(140) ? The boss believes you'll never guess which employee to be the best.

Moreover, some speakers report an amelioration effect with invasive at the subject position of a finite clause if the complementizer (**that**) selected by the verb of the master clause is not pronounced, as in (141b).[160]

(141) a: * Tom said that *I forgot* ***who*** is dating Amy**.**

b: ? Tom said *I forgot* ***who*** is dating Amy**.**

If that is the case, one could hypothesize that, given the structure in (137) above, the issue could potentially be partially reduced to the that-trace effect observed in the non-amalgamated versions of the examples.

(142) a: * I forgot $\mathbf{who_1}$ Tom said that $\mathbf{t_1}$ is $\mathbf{t_1}$ dating Amy.

b: I forgot $\mathbf{who_1}$ Tom said $\mathbf{t_1}$ is $\mathbf{t_1}$ dating Amy.

[160] I am thankful to Norbert Hornstein for discussion on this matter.

V.8 Dynamic Interpretation and Relativized 'Matrixhood'

As shown in **§II.9**, in any syntactic amalgam, the co-reference possibilities among pronouns and R-expressions that are distributed one in the 'invasive clause' and the other in the 'invaded clause' are exactly the ones in the corresponding paratactic paraphrase, rather than the readings available in the corresponding hypotactic paraphrase, as shown below.

(143) a: **[Homer]**$_{1}$ gave **[he]**$_{1/2}$ doesn't even remember how much money to Lisa.

b: **[He]**$_{*1/2}$ doesn't even remember how much money **[Homer]**$_{1}$ gave to Lisa.

c: **[Homer]**$_{1}$ gave money to Lisa. **[He]**$_{1/2}$ doesn't even remember how much.

(144) a: **[He]**$_{*1/2}$ gave **[Homer]**$_{1}$ doesn't even remember how much money to Lisa.

b: **[Homer]**$_{1}$ doesn't even remember how much money **[he]**$_{1/2}$ gave to Lisa.

c: **[He]**$_{*1/2}$ gave money to Lisa. **[Homer]**$_{1}$ doesn't even remember how much.

(145) a: **[Homer]**$_{1}$ gave **[the idiot]**$_{1/2}$ doesn't even remember how much money to Lisa.

b: **[The idiot]**$_{*1/2}$ doesn't even remember how much money **[Homer]**$_{1}$ gave to Lisa.

c: **[Homer]**$_{1}$ gave money to Lisa. **[The idiot]**$_{1/2}$ doesn't even remember how much.

(146) a: **[The idiot]**$_{*1/2}$ gave **[Homer]**$_{1}$ doesn't even remember how much money to Lisa.

b: **[Homer]**$_{1}$ doesn't even remember how much money **[the idiot]**$*_{1/2}$ gave to Lisa.

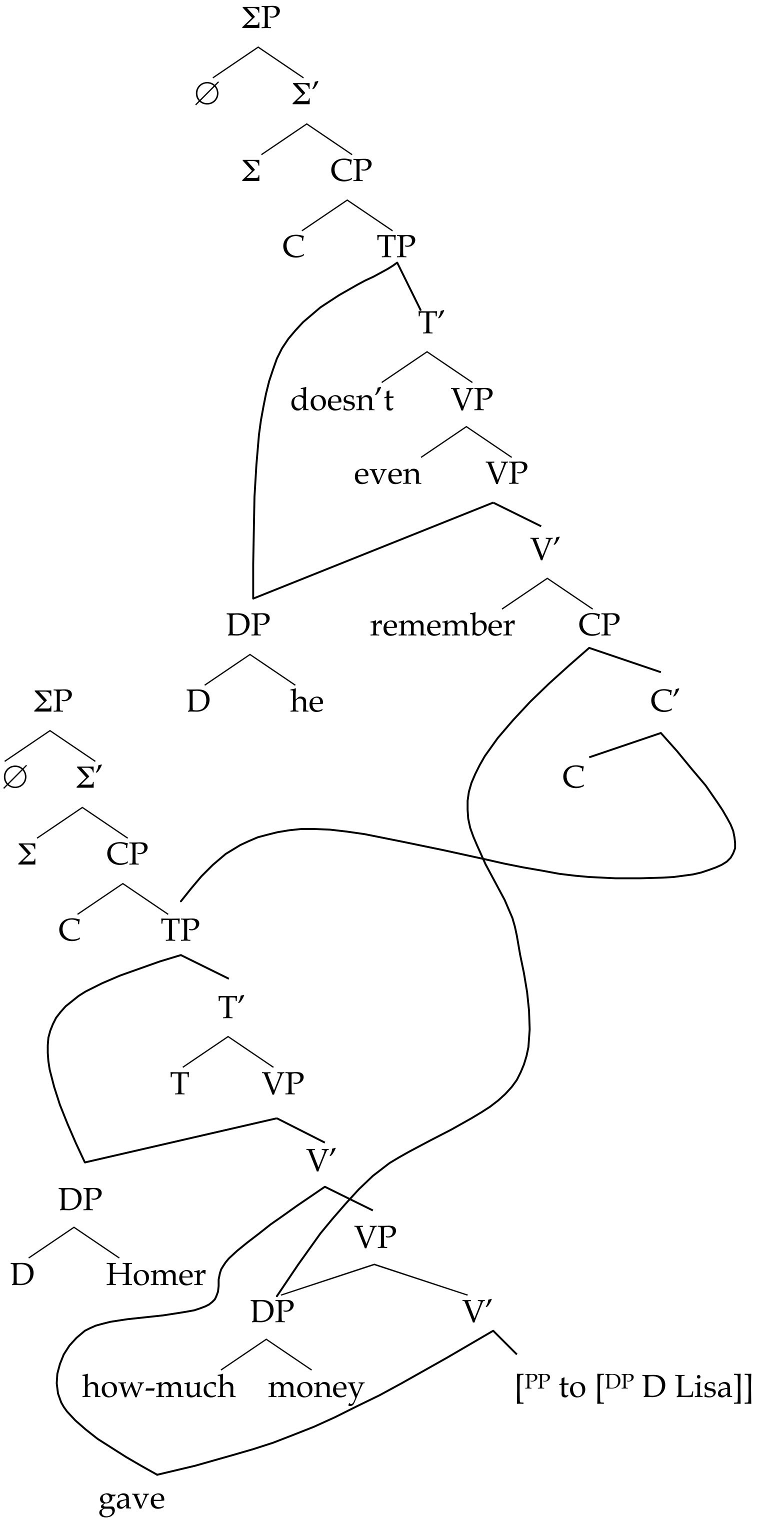
c: **[He]**$_{*1/2}$ gave money to Lisa. **[Homer]**$_{1}$ doesn't even remember how much.

Then, towards the end of **§III.2.3**, I have shown that the sluicing-based approach to amalgamation makes wrong predictions with regards to the facts above, as it presupposes a structure where there first of the two DPs in question c-commands the second one, which would lead to violation of Principle C of Binding Theory in cases like (146).

In principle, a similar problem seems to arise for the approach to amalgamation proposed in this dissertation. This is so because, given the multiply rooted representations assumed, the first of the two relevant DPs would be in the shared embedded clause, in a position where it is c-commanded by the second relevant DP, which would be in the spine of the subservient clause.

This is illustrated in (147), which corresponds to (143a).

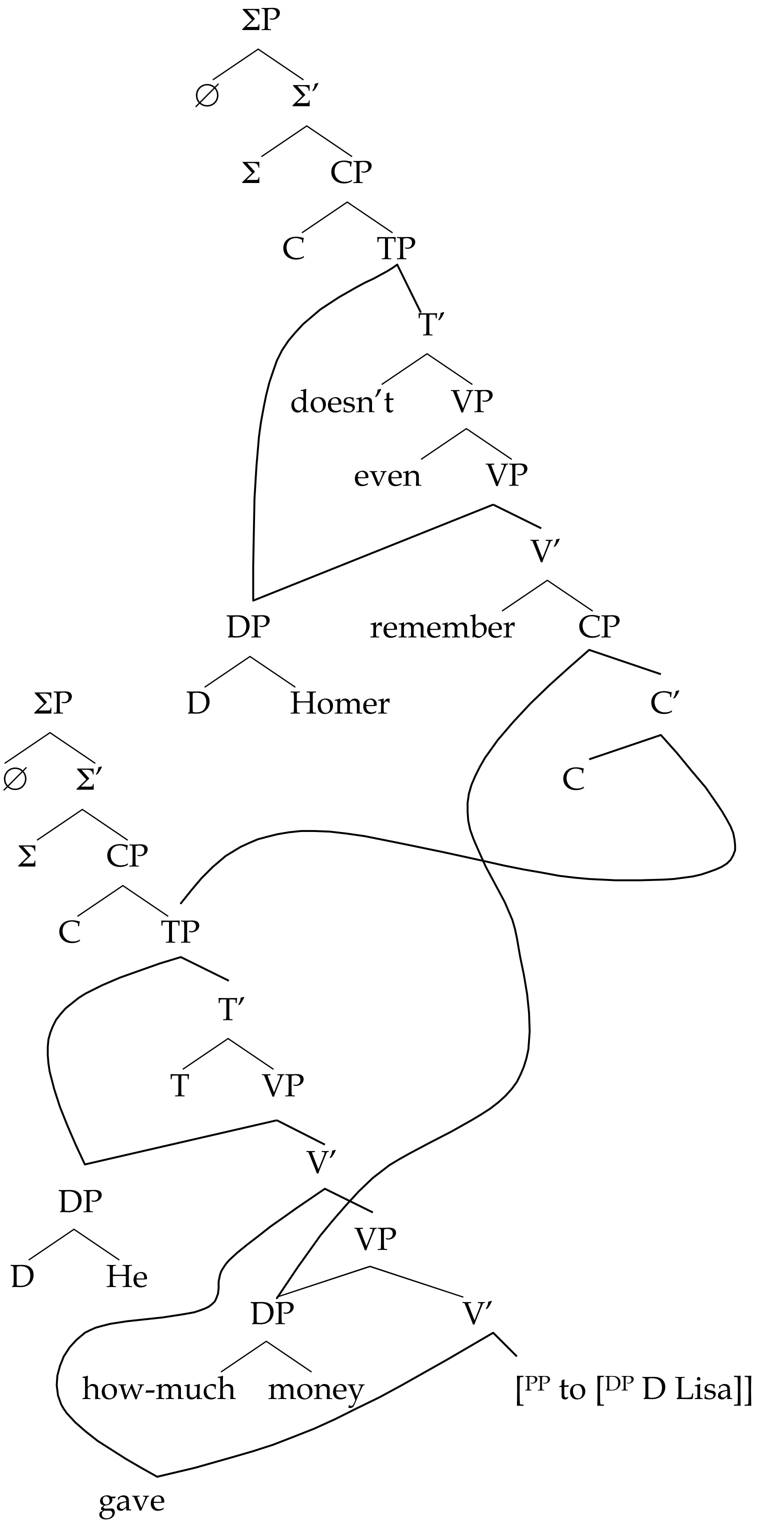
(147)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
doesn′t
VP
even
VP
V′
DP
remember
CP
D
he
C′
ΣP
C
∅
Σ′
Σ
CP
C
TP
T′
T
VP
V′
DP
D
Homer
VP
DP
V′
how-much
money
[PP to [DP D Lisa]]
gave

Notice that **he** c-commands **Homer** in (147), even though co-reference between the two is indeed possible. This is, in principle, problematic, to the extent that it is incompatible with Principle C of Binding Theory, which is strongly supported by a huge body of cross-linguistic data.

A similar problem concerns the impossibility of co-reference between **he** and **Homer** in (144a), which would be analyzed as in (148) below, where **Homer** is not c-commanded by **he**. Thus, *modulo* Principle C, co-reference is predicted to be possible, but it is not.

(148)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
doesn′t
VP
even
VP
V′
DP
remember
CP
D
Homer
ΣP
C′
∅
Σ′
C
Σ
CP
C
TP
T′
T
VP
V′
DP
VP
D
He
DP
V′
how-much
money
[PP to [DP D Lisa]]
gave

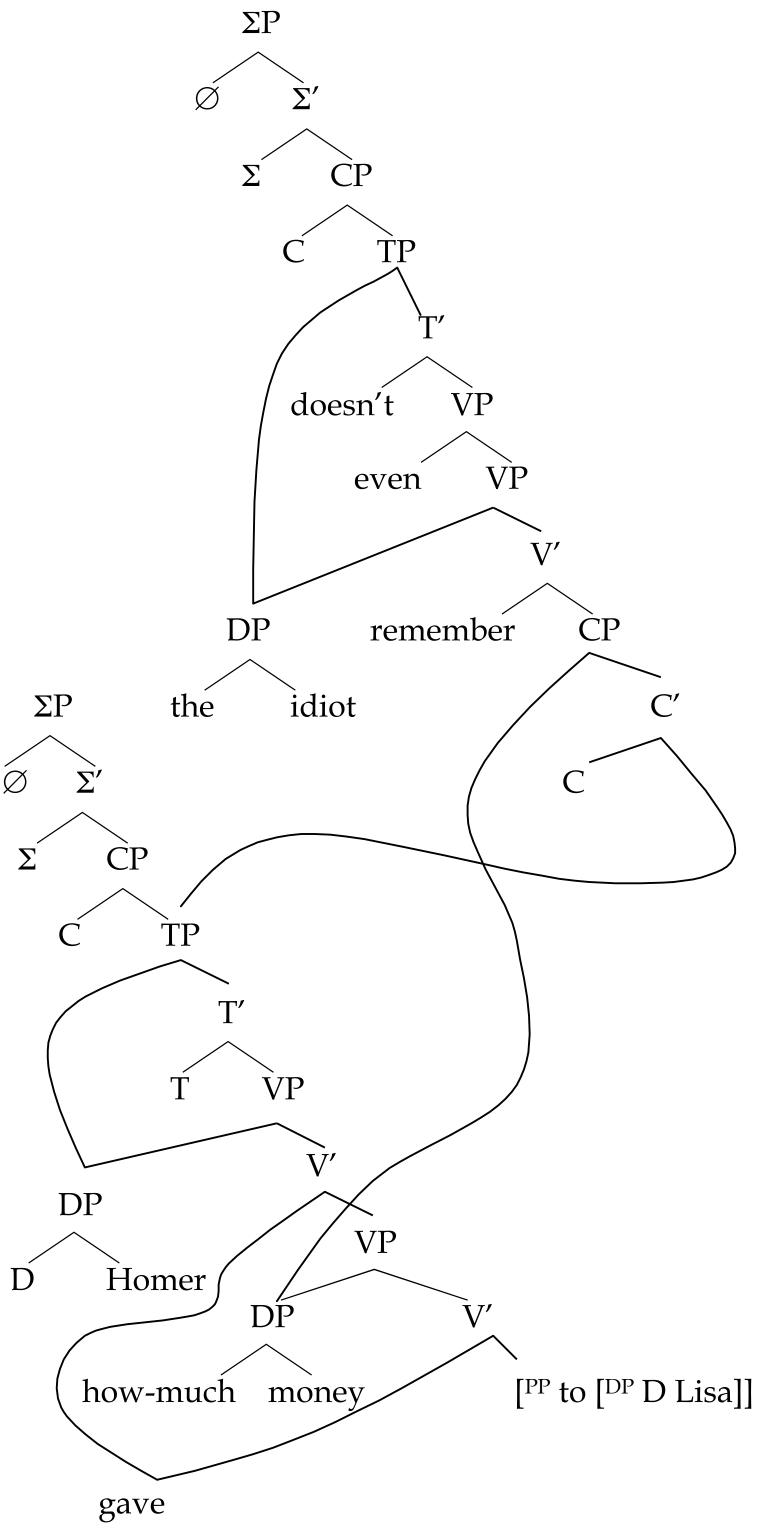
Now, take the case of potential co-reference between an epithet and a proper name. The relevant cases are (145a) and (146a), repeated below as (149) and (150).

(149) **[Homer]**$_{1}$ gave **[the idiot]**$_{1/2}$ doesn’t even remember how much money to Lisa.

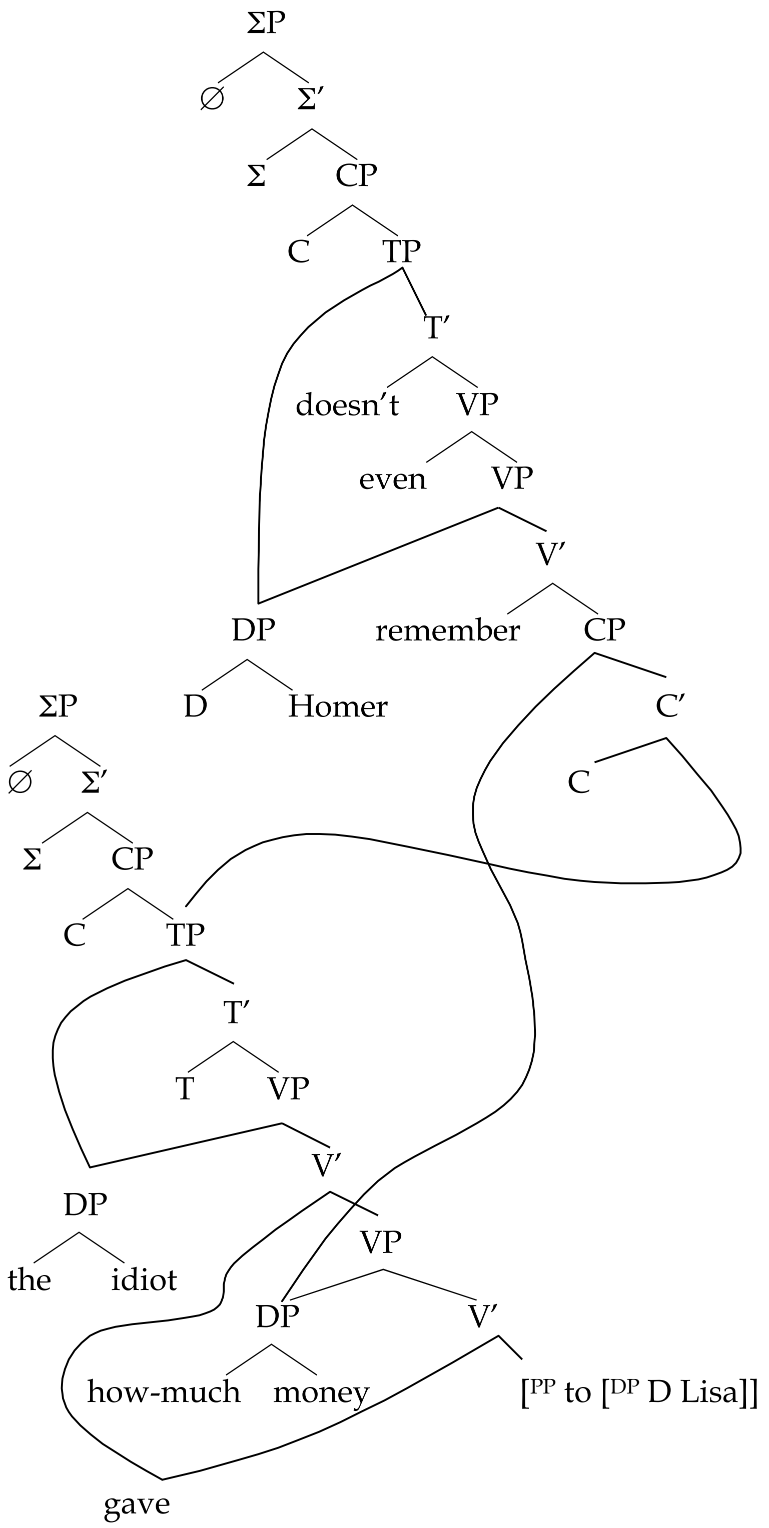
(150) **[The idiot]**$_{*1/2}$ gave **[Homer]**$_{1}$ doesn’t even remember how much money to Lisa.

Co-reference is possible in the first case but not in the second case. This contrast seems rather surprising, as the corresponding structures would be as in (151) and (152) respectively.

(151)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
doesn′t
VP
even
VP
V′
DP
remember
CP
the
idiot
ΣP
C′
∅
Σ′
C
Σ
CP
C
TP
T′
T
VP
V′
DP
VP
D
Homer
DP
V′
how-much
money
[PP to [DP D Lisa]]
gave

(152)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
doesn′t
VP
even
VP
V′
DP
remember
CP
D
Homer
ΣP
C′
∅
Σ′
C
Σ
CP
C
TP
T′
T
VP
V′
DP
the
idiot
VP
DP
V′
how-much
money
[PP to [DP D Lisa]]
gave

Notice that, in both structures above, there is an R-expression c-commanding another R-expression. In (151), **the idiot** c-commands **Homer**. In (152), **Homer** c-commands **the idiot**.

In (152), co-reference between **the idiot** and **Homer** is correctly predicted to impossible, modulo Principle C. In (151), however, the same logic leads to the prediction that co-reference between **Homer** and **the idiot** should be impossible as well. But such co-reference is possible.

In a nutshell, in all cases above, all relevant c-command relations in the Siamase-Tree configurations correspond exactly to the c-command relations in the corresponding 'hypotactic paraphrases'. However, the patterns of co-reference match the ones in the corresponding 'paratactic paraphrases', where the two DPs in question belong to two distinct unconnected sentences, therefore not standing in c-command relation with each other.

I do not claim to have an ultimate analysis for this phenomenon, but I like to point out that there should not underestimate the fact that the patterns exhibited by syntactic amalgams are identical to the ones found in the corresponding 'paratactic paraphrases'. I propose that such similarity is the key notion.

Let us focus on the 'paratactic paraphrases' now.

(153) **[Homer]**$_{1}$ gave money to Lisa. **[He]**$_{1/2}$ doesn't even remember how much.

(154) **[He]**$_{*1/2}$ gave money to Lisa. **[Homer]**$_{1}$ doesn't even remember how much.

(155) **[Homer]**$_{1}$ gave money to Lisa. **[The idiot]**$_{1/2}$ doesn't even remember how much.

(156) **[He]**$_{*1/2}$ gave money to Lisa. **[Homer]**$_{1}$ doesn't even remember how much.

In all cases, each of the two relevant DPs belongs to a distinct sentence. Therefore, neither DP c-commands the other. Consequently, whatever is ultimately responsible for the co-reference patterns above, it certainly has nothing to do with Binding Theory, which is dependent on the notion of c-command.

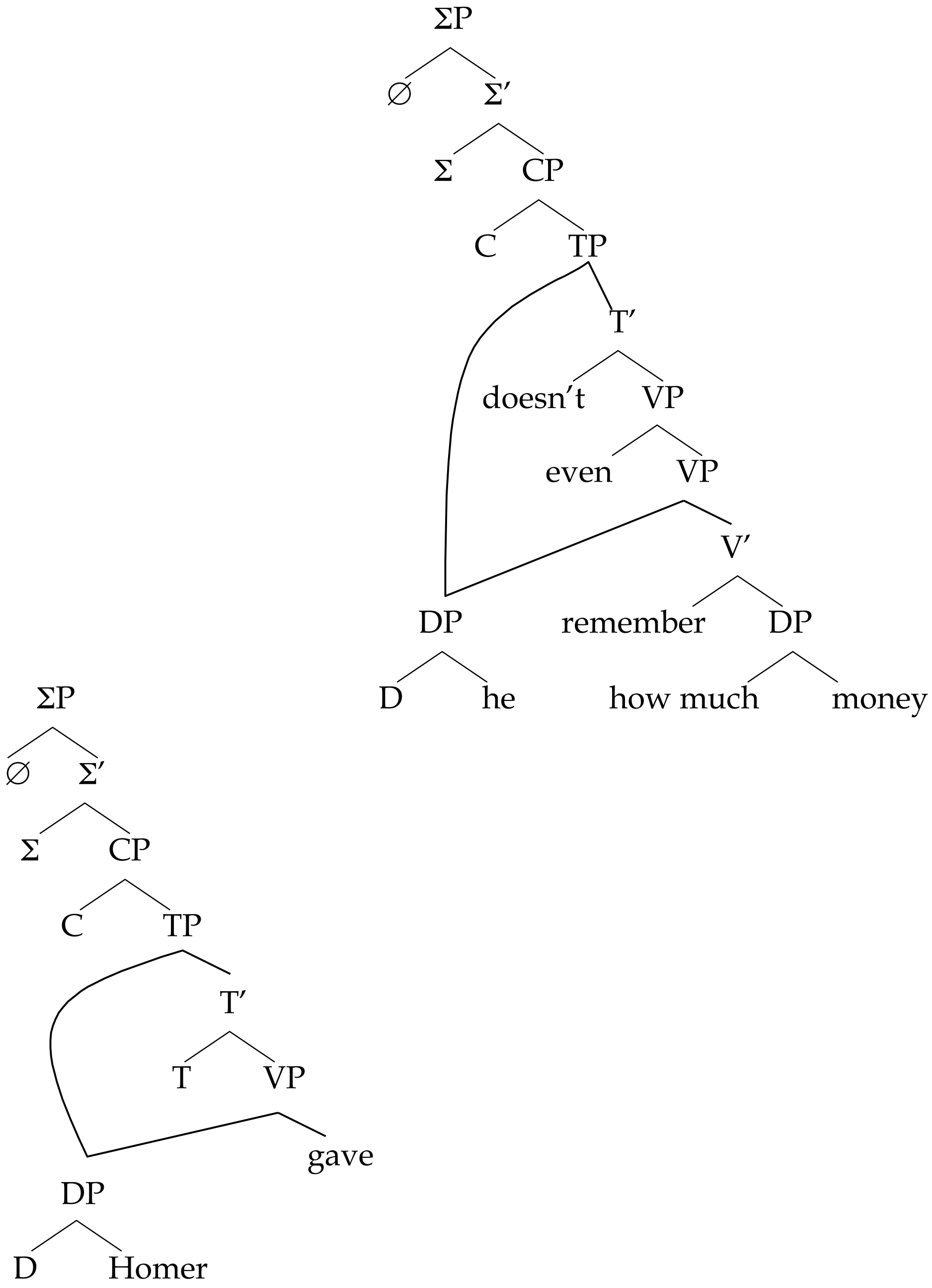
In this dissertation, I will not even speculate about what could be the cause of the co-reference patterns above. I will simply take it for granted that there are discursive-pragmatic principles of some sort, which derive the facts.

That being the case, I propose that the very same principles are responsible for the facts in syntactic amalgams.

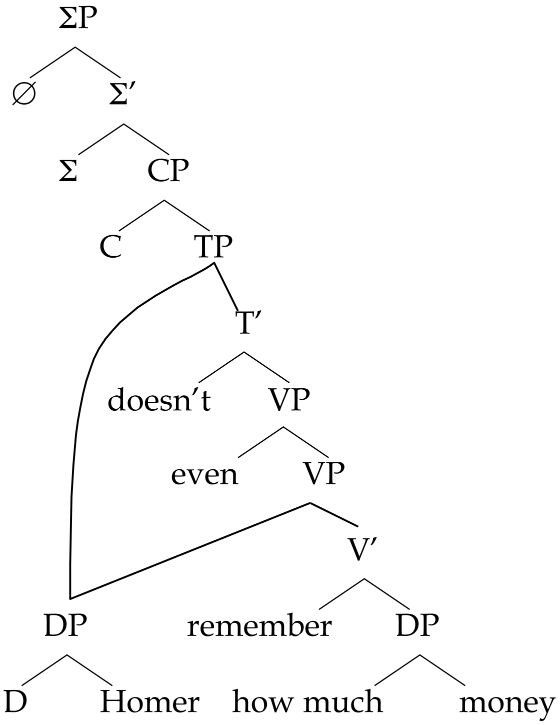
Notice that, although, in every case of amalgamation, it is the case that one of the relevant DPs c-commands the other in the final LF-representation, there is a moment in the derivation of the Siamse-Tree when the master clause and the subservient clause are not connected yet.

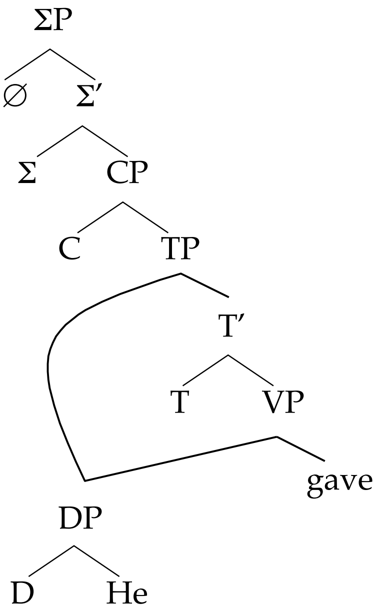
This is illustrated below for all four cases of amalgamation discussed in this section.

(157)

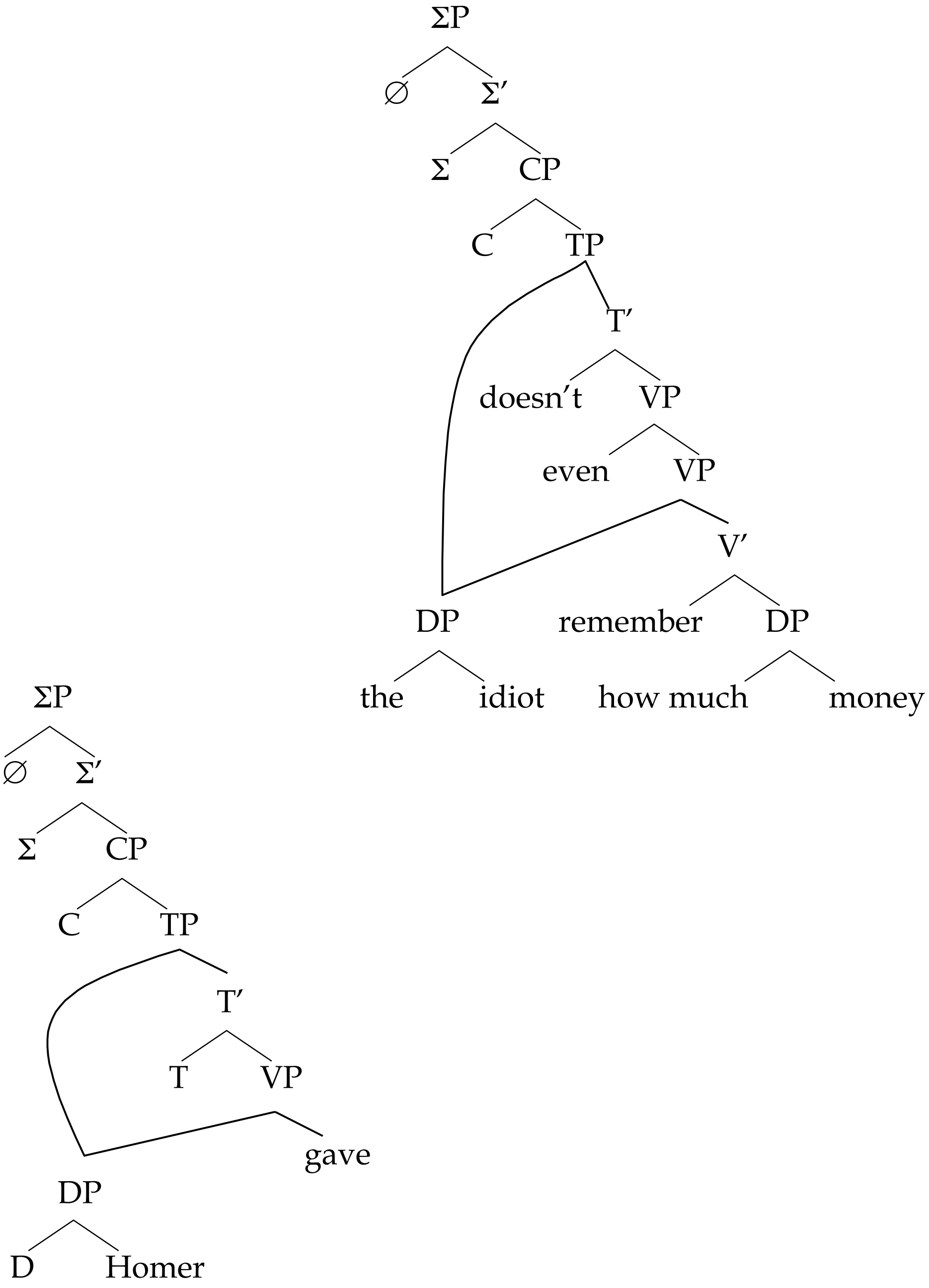


(158)

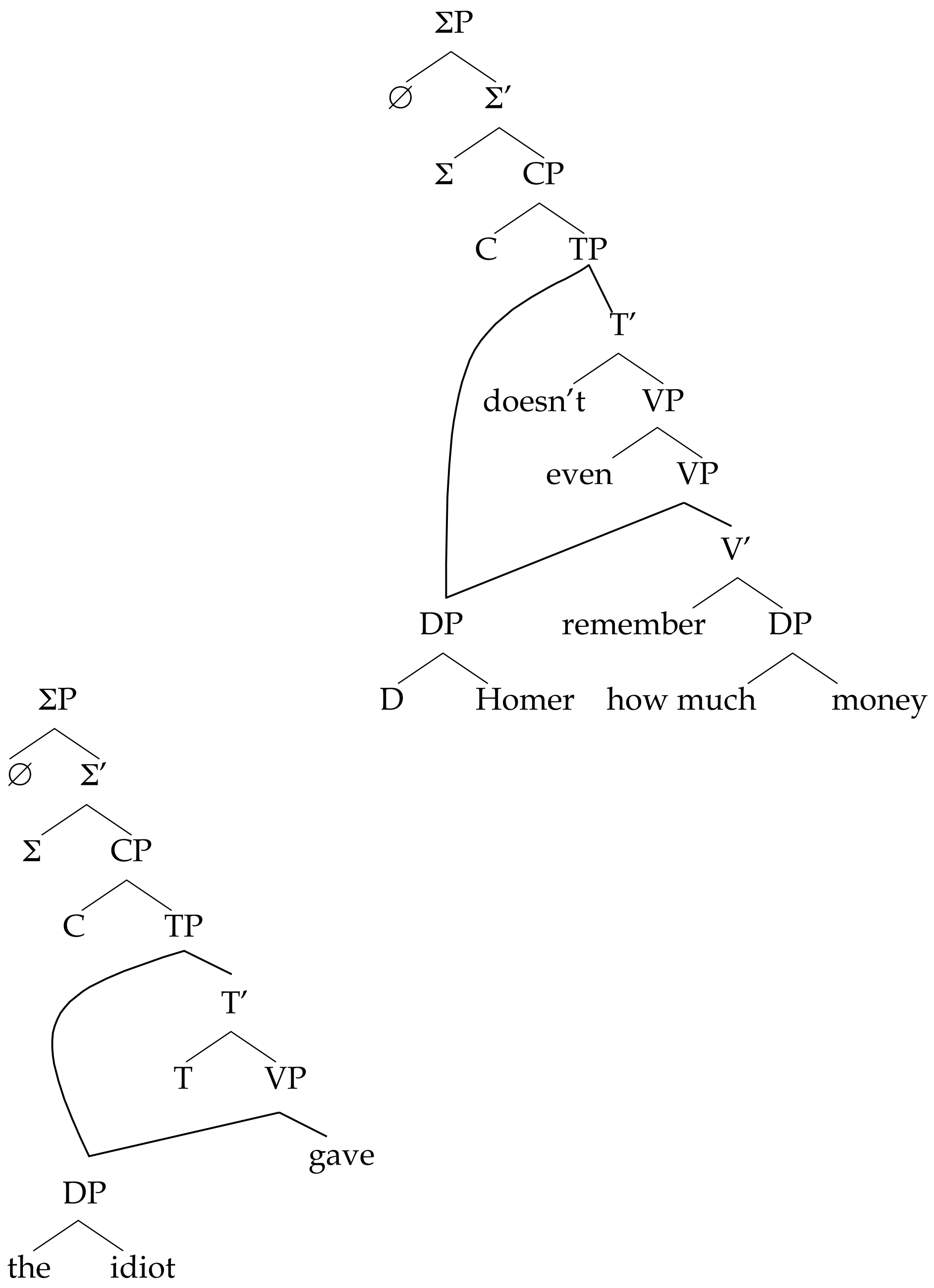

ΣP
∅
Σ′
Σ
CP
C
TP
T′
doesn’t
VP
even
VP
V′
DP
remember
DP
D
Homer
how much
money


ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
gave
DP
D
He

(159)


ΣP
∅
Σ′
Σ
CP
C
TP
T′
doesn′t
VP
even
VP
V′
DP
remember
DP
the
idiot
how much
money
ΣP
∅
Σ′
Σ
CP
C
TP
T′
T
VP
gave
DP
D
Homer

(160)



In such configurations, there is no c-command relation between the two relevant DPs, just like what happens with the 'paratactic paraphrases'. At that specific point, the subservient clause is not yet 'behind' the 'master clause', but just 'after' it, since the lowest TP has not been shared yet. Therefore, it is as if there were two independent (incomplete) sentences one after the other, as shown below.

(161) **[Homer]**$_{1}$ gave...

[He]$_{1/2}$ doesn't even remember how much money...

(162) **[He]**$_{*1/2}$ gave...

[Homer]$_{1}$ doesn't even remember how much money...

(163) **[Homer]**$_{1}$ gave...

[the idiot]$_{1/2}$ doesn't even remember how much money...

(164) **[The idiot]**$_{*1/2}$ gave...

[Homer]$_{1}$ doesn't even remember how much money...

My suggestion, then, is that the system proposed in chapter IV should be combined with some modified version of Lebeaux's (1995) and Epstein, Groat, Kawashima & Kitahara's (1998) theories, where interpretation of DPs, and satisfaction of Binding Principles, is done in a dynamic fashion, as the LF-representation is built.

I will leave to future research the task of formalizing the details of this intuition, such as how exactly such dynamic interpretive devices would be sensitive relates to the notions of ‘derivational round’ and ‘behindness’. The basic idea is that, at the relevant point, the there are two parallel (incomplete) sentences still unconnected, which would give rise to the same results found in ‘paratactic paraphrases’.

VI

Concluding Remarks

Having described and analyzed the phenomenon of amalgamation in the previous chapters, now I make my final remarks, first summarizing what I consider to be the main conclusions about the theory of UG which can be drawn from the study of amalgamation (cf. **VI.1**), and pointing out issues to be addressed in future research (cf. **VI.2**).

VI.1. Conclusions

The main analytical points made in this dissertation are as follows:

(i) Syntactic amalgams are not the same thing as parenthetical constructions, as amalgamation involves the sharing of some syntactic material between the invasive and the invaded clauses in a way that parentheticalization does not; and this has major consequences for the establishment of syntactic relations across these domains (binding, movement, etc).

(ii) Syntactic amalgamation does not involve a combination of sluicing and DP-ellipsis (contra Lakoff 1974, Tsubomoto & Whitman 2000). Such an approach fails to account for all the empirical generalizations in chapter **II**.

(iii) Syntactic amalgamation does not involve topicalization of an embedded TP through a remnant movement mechanics. Such an approach also fails to account for all the empirical generalizations in chapter **II**.

(iv) Syntactic amalgams involve multiple matrix clauses that share the same embedded clause (which is why some constraints on long-distance dependencies (e.g. superiority, island) get obfuscated in such constructions, given the existence of *quasi*-parallel domains where the relevant chain link may 'escape' the effects of the relevant constraint).

The main <u>theoretical points</u> made in this dissertation are as follows:

(v) Amalgamation requires a derivational approach to syntax.

(vi) Phrase-structure building uniformly involves tucking-in, so that derivations proceed in a top-to-bottom fashion.

(vii) Constituency is dynamic (mutant).

(viii) Derivational time equals real time (so that the order of pronunciation of terminals mimics the order in which lexical tokens access the derivation.

(ix) Movement is the consequence of a phrase being remerged into a new position, and having multiple mothers.

(x) Remerge is not limited to chain formation. When the multiple mothers of a remerged phrase do not stand in a dominance relation, a multiply-rooted phrase marker arises.

(xi) Multiply-rooted structures are formed through overlapping computations, that start out from numerations that intersect.

(xii) The paratactic aspect of syntactic amalgamation can be reduced to 'syntax pushed to the limit'.

VI.2. Directions for Future Research

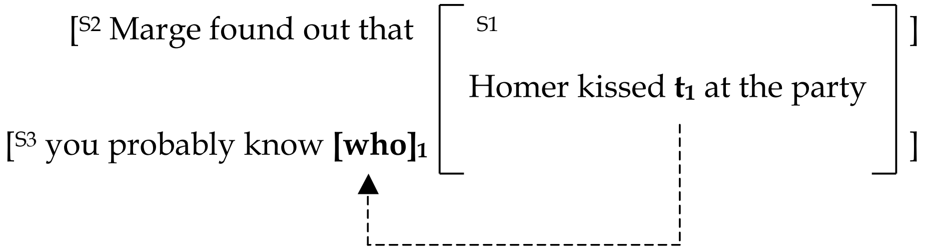
VI.2.1.. Semantics

Once syntactic amalgams are treated in terms of multiply-rooted syntactic representations, a puzzle arises with regards to how such 'Siamese Trees' are to be semantically interpreted. For instance, consider the syntactic amalgam in (01).

(01) Marge found out that Homer kissed you probably know who at the party.

By the analysis proposed chapter **V**, the syntactic structure of (01) is something along the lines of the representation sketched in (02).

(02)



The absence of a single-root in (02) makes it impossible to calculate a truth-value for the amalgamated structure as a whole in any standard fashion. Although it might seem relatively trivial, at first blush, to calculate *quasi*-independent truth-values for each of the sentences constituting the multiply-rooted syntactic representation,[161] it is not obvious how those parallel interpretations can obtain in the desired fashion, the WH-chain is formed with a link within the shared embedded clause and another link exclusively in a distinct matrix clause. In principle, the WH-trace– or equivalent notion (e.g. copy, occurrence, etc) – would count as a variable bound by a WH-operator in the domain of one of the parallel matrix clauses, but remain as a free variable in the domain of the other parallel sentence(s), presumably getting existential import by default, as sketched in (03).

(03) $\exists \boldsymbol{p}$ [$\boldsymbol{p}$ = [Homer kissed a person $\overset{\substack{\textit{free variable}\\ \downarrow}}{\boldsymbol{x}}$ at the party] &
[Marge found out that $\boldsymbol{p}$] &
$\underset{\substack{\uparrow\\ \textit{operator}}}{\exists \boldsymbol{x}}$ [the listener probably knows what the identity of $\overset{\substack{\textit{bound variable}\\ \downarrow}}{\boldsymbol{x}}$ is,
such that $\boldsymbol{p}$]]

The problem is that (03) is not the actual semantic interpretation attested for (01)-(02). Rather, something roughly along the lines of (04) obtains.[162]

[161] i.e. S_2 = [Marge found out that Homer kissed t_1 at the party] & S_3 = [The listener knows who_1 Homer kissed t_1 at the party].

[162] The semantic structure in (03) is compatible with any of the three situations below. The first (i) and the second (ii) cases correspond to the logical possibility where the first (unbound) ***x*** and the second (bound) ***x*** have identical values. The third (iii) case corresponds to the logical possibility where the first (free) ***x*** and the second (bound) ***x*** have distinct values.

(04) $\boldsymbol{\exists x}, \boldsymbol{\exists p}$ [$\boldsymbol{p}$ = [Homer kissed a person $\boldsymbol{x}$ at the party] & *bound variable*
[Marge found out that $\boldsymbol{p}$] &
[the listener probably knows what the identity of $\boldsymbol{x}$ is]]

operator

As opposed to (03), the operator that binds the variable corresponding to the WH-trace in (04) crucially scopes over all the material in the whole amalgamated structure, as if there were a single root in the syntactic representation to where such operator could be adjoined through Quantifier Raising at LF, or whatever the actual grammatical mechanism is.

Providing a solution to this problem is something that would go way beyond the scope of this dissertation. In a parallel research (Guimarães 2003e), I propose to derive (04) from (02), within the truth-theoretical framework of Larson & Ludlow (1993) and Larson & Segal (1995),[163] which stems from previous work by Tarski (1944, 1956) and Davidson (1965, 1967, 1968, 1970, 1984). The crucial tree-node that does the job of the 'illusory single-root' is the node

i: Homer kissed Peggy at the party; and Marge found that out. Also, the addressee of a given utterance of (01) probably knows that Peggy was the one kissed by Homer at the party.

ii: Homer kissed both Peggy and Amy at the party. Marge found out that Homer kissed both Peggy and Amy at the party. Also, the addressee of a given utterance of (01) probably knows that both Peggy and Amy were the ones kissed by Homer at the party.

iii: Homer kissed both Peggy and Amy at the party. Marge found out that Homer kissed Peggy at the party (but she didn't find out about Amy). Also, the addressee of a given utterance of (01) probably knows that Amy was the one kissed by Homer at the party.

On the other hand, the semantic structure in (04) is compatible only with situations (i) and (ii). Given that any utterance of (01) can be true in situations like (i) or (ii), but not in situations like (iii), a semantic analysis along the lines of (04) is empirically adequate, whereas the one in (03) is not..

[163] See also Higginbotham (1985, 1986) and Pietroski (2002, forthcoming-a, forthcoming-b).

corresponding to the shared embedded clause (i.e. S_1), due to some interpretive devices that piggyback on that node to fix the contexts (formalized as Tarskian σ-sequences) according to which WH-quantifiers quantify over. That way, the semantic values assigned to each trace/variable in a given parallel semantic computation are transmitted up and down the tree in the desired fashion, having the effect of variables being bound as in (03).

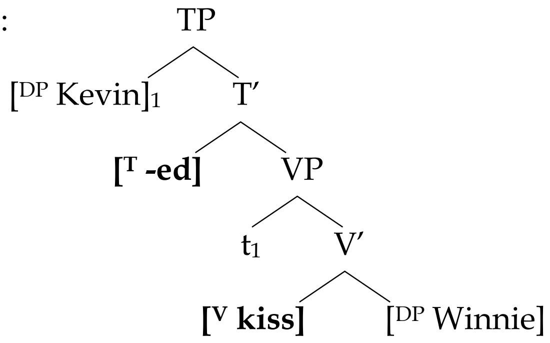
Further research is necessary to refine the formalism proposed in Guimarães (2003e), and to make sure that it is fully compatible with everything I said here (and vice-versa).

VI.2.2. Head Movement

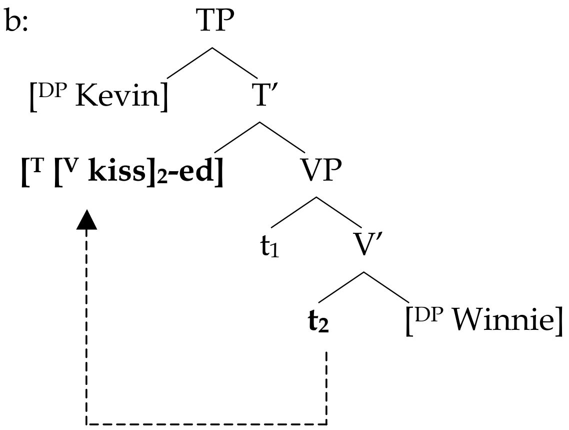
Quite a lot has been said about movement of maximal projections in this dissertation. However, in all parts of the analysis, I abstracted away from any potential instance of head movement other than the movement of the verb within the VP-shell itself (which, following Phillips' (1996, 2003), I take to be a case of 'reprojection').

In principle, head movement should involve the same remerge mechanics involved in phrasal movement. However, head movement typically involves 'morphological incorporation' (cf. Backer 1988), as in (06), whose effects at PF do not follow straightforwardly in a top-to-bottom system, where the order of pronunciation mimics the order in which the terminals access the derivation.

(06) a:



b:



Fortunately, a great amount of work on this topic has been done by Phillips (1996: chapter 4), who advocates for a decompositional approach to head movement (‘early morphologization’ coupled with ‘excorporation’). Although the basic essence of his system is compatible with the one being proposed here, future research is needed in order to work out all the technical details in a way compatible with everything else I said here about movement of phrasal constituents.

VI.2.3. Linearization

Another topic for future research is the status of the LCA in this theory. On the one hand, the LCA seems to be crucial in order to yield the desired prosodic patterns (cf. appendix to chapter IV), by establishing an alignment between boundaries of syntactic constituents and prosodic constituents. On the other hand, the LCA seems somewhat redundant in the system, to the extent that it is unnecessary when it comes to linearization *per se* (since the desired spec-head-comp order follows independently from the principles governing the mechanics of Merge).

Intuitively, if the general design specifications of the model proposed here are on the right track, is must be the case that there is something in the grammar playing the role that the LCA plays in this work, but it is not quite the LCA itself, as stated here. Perhaps, it is the case that, instead of the syntax delivering partial strings of terminals to the phonological component in cascades, it is the phonological component that accesses the derivation 'on the fly', therefore 'interpreting' the phrase marker and keeping track of all constituency changes, so that the effects of an LCA-based prosodic phrasing obtain. For now, I will pay the price of 'redundancy' and I will leave the refinement of the syntax/PF interface for the future.

VI.2.4. ~~Amalgamation as Sluicing~~, Sluicing as Amalgamation

One of the main goals of chapter **III** was to argue that syntactic amalgamation does not involve sluicing. Such goal has been achieved to a large extent.

Nevertheless, it is impossible to deny the striking structural similarity between *bona fide* sluiced sentences and what I have been calling 'invasive clauses' present in syntactic amalgams.

Apart from the obvious resemblance with regards to the string of terminals at PF, the two constructions pattern together in other ways, with regard to context sensitive operations. For instance, like in syntactic amalgams, island effects are absent from sluiced sentences (cf. Merchant 2001: chapter 3). Also, as described in **§II.9**, the co-reference possibilities inside amalgams mimic the ones observed across two paratactically related sentences, one of which is sluiced.

Yet another structural similarity between amalgams and sluiced sentences concerns the absence of successive cyclic WH-movement inside invasive clauses, which is a quite robust empirical generalization left out of chapter **II** for expository reasons.

For reference, consider first the pair in (07).

(07) a: Homer drank only Moe knows exactly how many beers last night.

b: Only Moe knows exactly how many beers Homer drank last night.

As discussed in **§II.3**, in syntactic amalgams, the substring that looks like a parenthetical chunk may be complex, exhibiting (an unbounded number of) embedded sentences in it, as in (08).

(08) a: Homer drank I bet only Moe knows exactly how many beers last night.
b: I bet only Moe knows exactly how many beers Homer drank last night.

However, successive cyclic movement is not tolerated inside those parenthetical chunks in amalgams, as in (09a). Notice that the non-amalgam version of the relevant example does allow successive cyclic movement, as in (09b).

(09) a: * Homer drank I wonder how many beers Marge thinks last night.
b: I wonder how many beers Marge thinks Homer drank last night.

Nothing in my analysis makes this prediction. On the other hand, if we take invasive clauses to be sluiced sentences, the pattern follows straightforwardly, as shown in (10).

(10) a: [^{S} Homer drank [^{NP} [^{NP} *e* [^{S} I wonder **[how many beers]**$_{1}$ ~~Marge thinks **t**$_{1}$ Homer drank **t**$_{1}$ last night~~]] last night.
b: * [^{S} Homer drank [^{NP} [^{NP} *e* [^{S} I wonder **[how many beers]**$_{1}$ Marge thinks ~~**t**$_{1}$ Homer drank **t**$_{1}$ last night~~]] last night.

Compare (10) with (11).

(11) a: Marge thinks that Homer drank a certain number of beers last night.
I wonder **[how many beers]**$_1$ ~~Marge thinks **t_1** Homer drank **t_1** last night~~

b: * Marge thinks that Homer drank a certain number of beers last night.
I wonder **[how many beers]**$_1$ Marge thinks ~~**t_1** Homer drank **t_1** last night~~

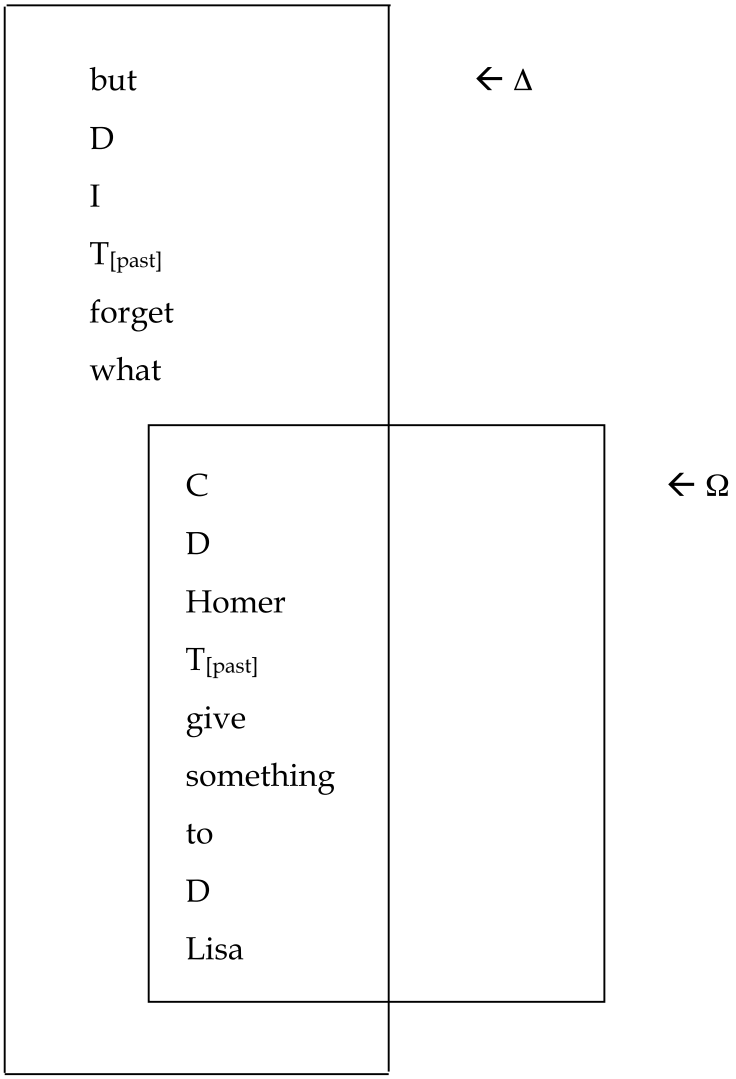
Basically, the pattern follows from whatever independent principle mandates that the deletion process inherent to sluicing affects all the material that follows the WH-phrase.

However, once an analysis along the lines of (09) is adopted, we automatically face all the problems mentioned in **§III.2** with regards to many other empirical generalizations discussed in chapter **II**.

In this context, I would like to suggest one direction of research to be explored in the future, as a step towards resolving this tension. It may be the case that, although amalgamation is not a subcase of sluicing, sluicing is a subcase of amalgamation. What I mean by this is that what we call amalgamation may be just one epiphenomenal byproduct of overlapping computations, which may take place in several other ways.

In all instances of overlapping computations discussed so far, I have only considered the cases where two or more numerations intersect, such that the intersection is a proper subset of all numerations involved. Other logically possible mathematical possibilities exist. For instance, consider (12), where the whole numeration Ω is a proper subset of the numeration Δ.

(12)

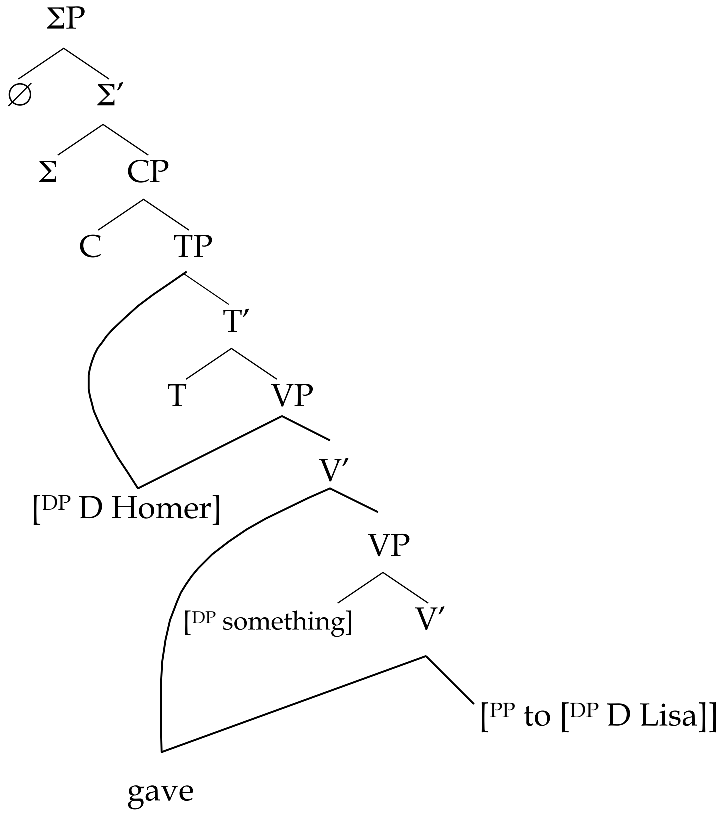


Perhaps, this is the input that leads to the typical case of sluicing in (13).

(13) Homer gave something to Lisa. But I forgot what.

The relevant derivation would start from numeration Ω and build the master clause in (14), whose corresponding PF-string is (15).

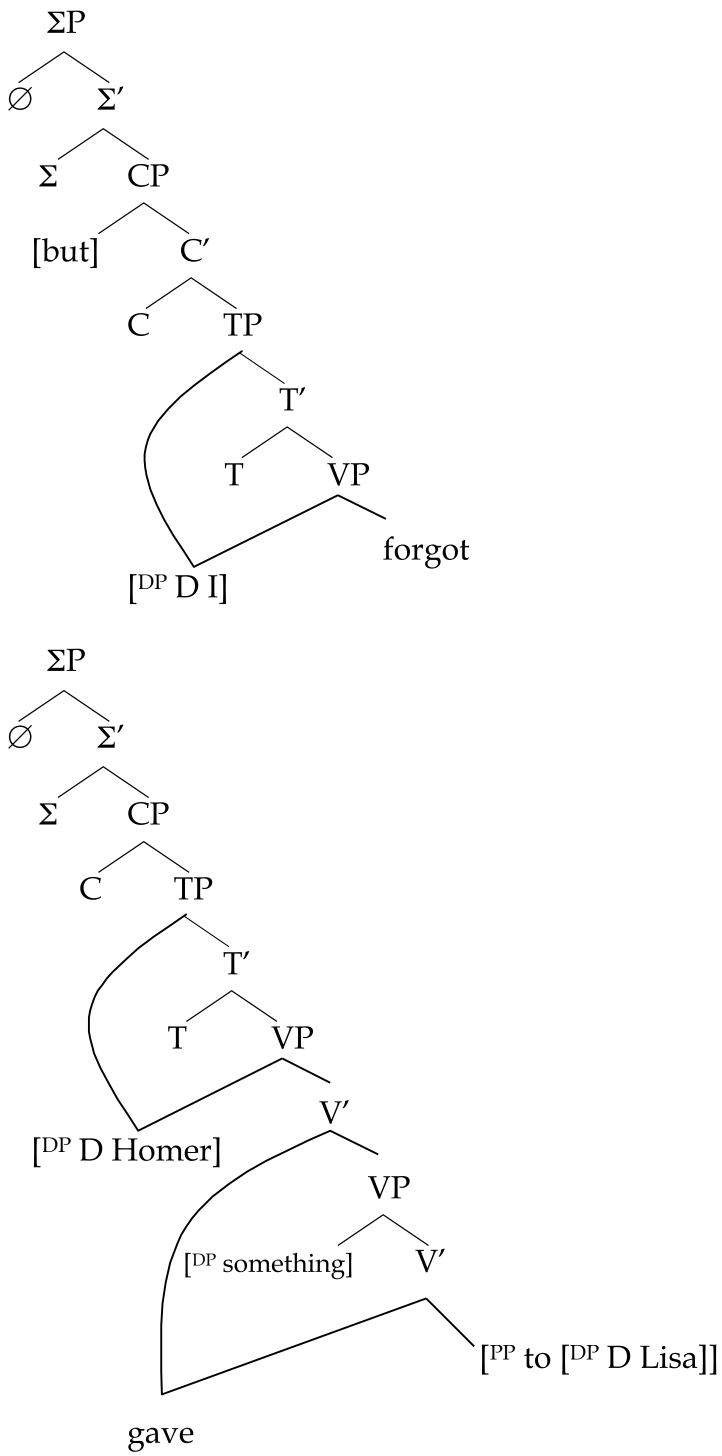
(14)



(15) Homer gave something to Lisa...

Then, the computational system starts building the subservient clause from the tokens in numeration Δ, up to the point in (16).

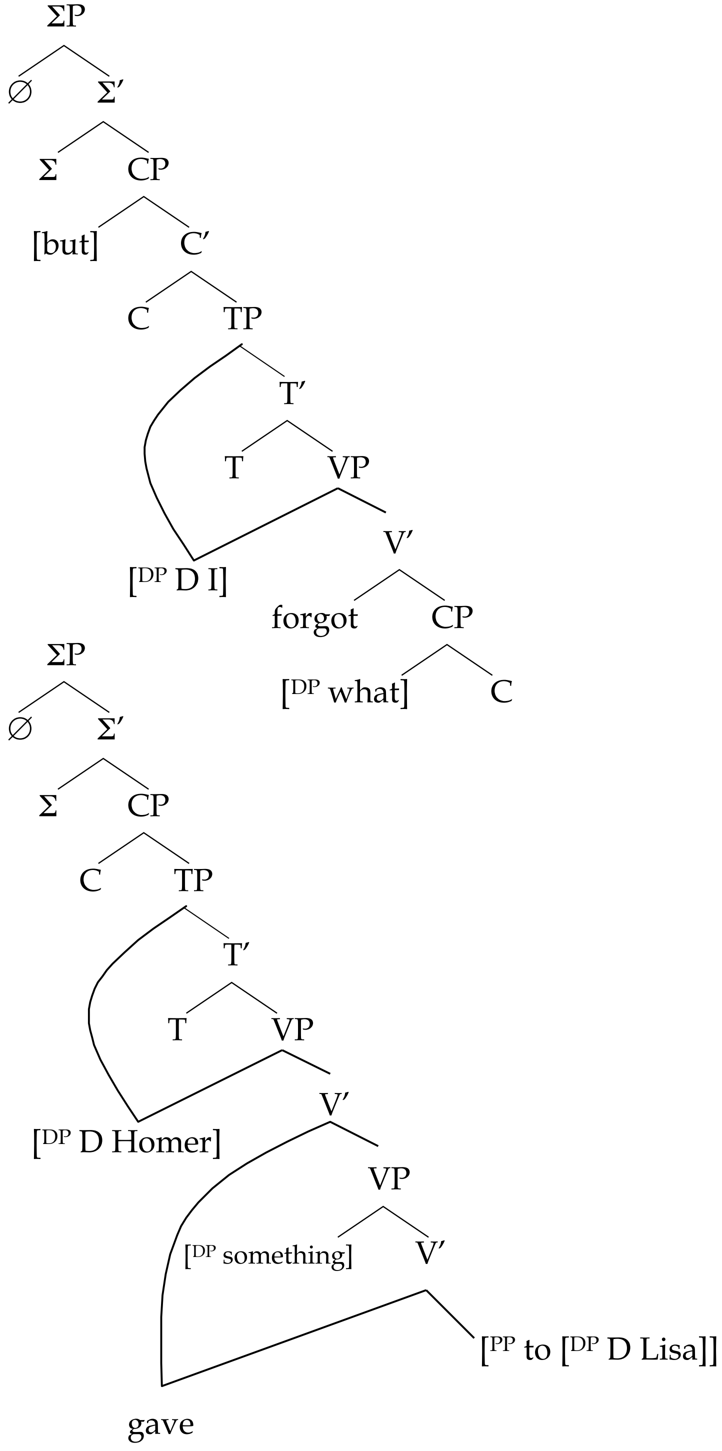
(16)



(17) Homer gave something to Lisa... But I forgot

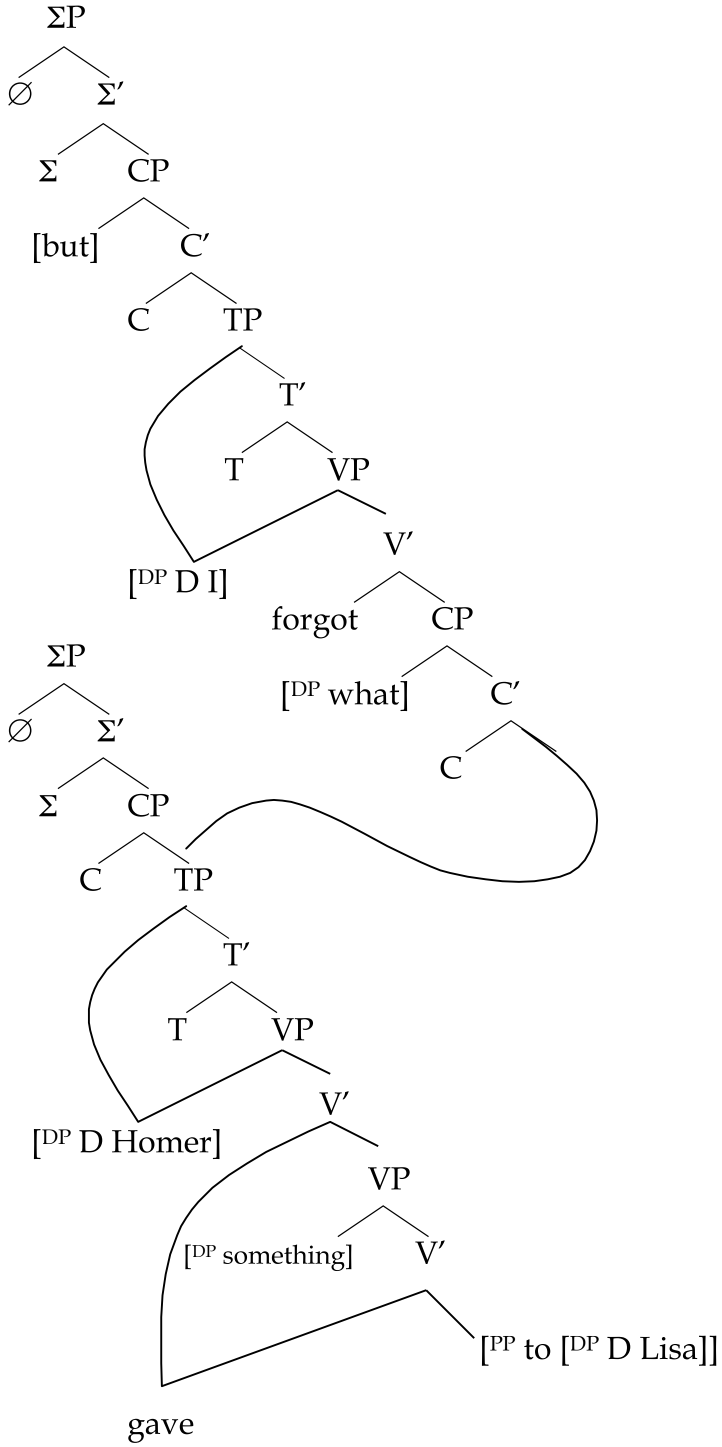
The next step would be the introduction of the WH element, as in (18).

(18)



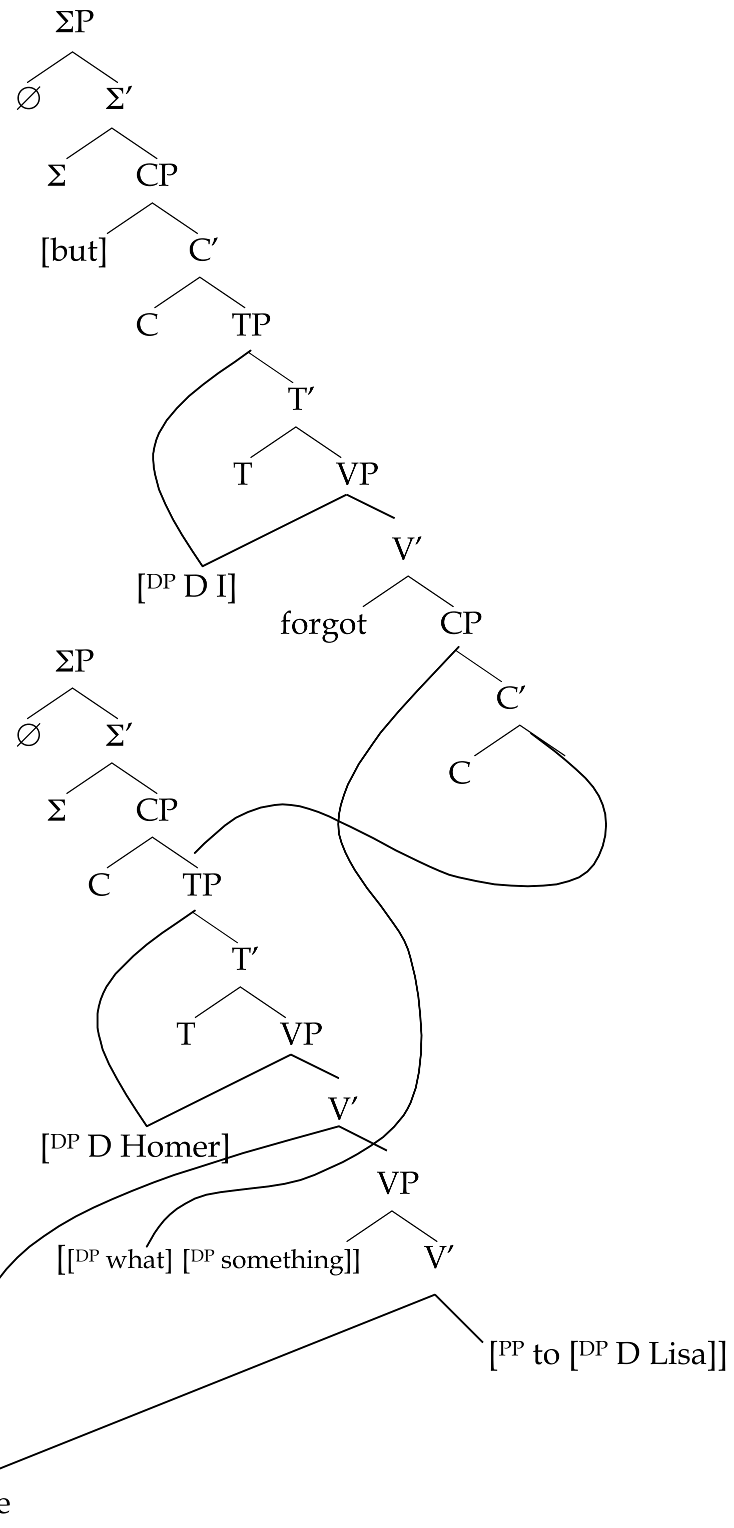
After that, the TP is shared, as in (19).

(19)



The WH-element is then lowered and adjoined to the indefinite, in a process akin to 'vehicle change', as in (20)

(20)



What I have just shown is obviously just a mere intuition to be explored. Many technical details need to be worked out.

At any rate, if something roughly along these lines is on the right track, then we may have an explanation for the pattern below.

(21) a: Homer gave a book to Lisa. But I forgot what.

b: Homer gave a book to Lisa. But I forgot which.

c: * Homer gave a book to Lisa. But I forgot which book.

(22) a: Homer gave a book about saxophones to Lisa. But I forgot which.

b: * Homer gave a book about saxophones to Lisa. But I forgot which book.

c: ** Homer gave a book about saxophones to Lisa. But I forgot which book about saxophones.

(23) a: Homer gave a book about saxophones written by Paul Desmond to Lisa. But I forgot which.

b: * Homer gave a book about saxophones written by Paul Desmond to Lisa. But I forgot which book.

d: ** Homer gave a book about saxophones to Lisa. But I forgot which book about saxophones.

c: ** Homer gave a book about saxophones written by Paul Desmond to Lisa. But I forgot which book about saxophones written by Paul Desmond.

The basic idea is that a ‘bare’ WH-phase like **<u>what</u>** is really something like **<u>wh</u>** + **<u>something</u>**. And the ‘vehicle-change’-like process in (21) is nothing but ordinary syntactic and semantic composition. The more and more distant from the bare indefinite that the ‘antecedent’ of the WH gets, the worse the result of the ‘vehicle-change’-like process gets.

Bibliography

Abels, K. 2001. *Move?* Doctoral Research Paper. University of Connecticut, Storrs.

Abraham, W., S. Epstein, H. Thráinsson & C. Zwart. 1996. *Minimal ideas: Syntactic studies in the minimalist framework*. Amsterdam: John Benjamins.

Aoun, J., N. Hornstein & D. Sportiche. 1981. Some Aspects of Wide Scope Quantification. *Journal of Linguistic Research*. 1: 69-95.

Baker, M. 1988. *Incorporation*. Chicago: University of Chicago Press.

Bobaljik, J. 1995. In Terms of Merge: copy and head movement. *MITWPL* 27: 41-64.

Bobaljik, J. 1995. In Terms of Merge: copy and head movement. R. Pensalfini & H. Ura (eds.) *Papers in minimalist syntax: MIT working papers in linguistics* 27, pp. 41-64.

Brody, M. 1995. *Lexico-logical Form: A radically minimalist theory*. Cambridge/MA: MIT Press.

Brody, M. 1998. Projection and phrase structure. *Linguistic Inquiry* 29, pp. 367-398.

Brody, M. 2000. Mirror theory: syntactic representation in perfect syntax. *Linguistic Inquiry* 31, pp. 29-56.

Chametzky, R. 1996. *A Theory of Phrase Markers and The Extended Base*. Albany: SUNY Press.

Chomsky, N. 1955. *The logical structure of linguistic theory*. University of Chicago Press [1975].

Chomsky, N. 1973. Conditions on Transformations. in: S. Anderson & P. Kiparski (eds.) *A Festschrift for Morris Halle*. New York: Holt, Rinehart and Wilson.

Chomsky, N. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, N. 1982. Concepts and consequences of the theory of government and binding. Cambridge, Mass.: MIT Press.

Chomsky, N. 1986a. *Barriers*. Cambridge, Mass.: MIT Press.

Chomsky, N. 1986b. *Knowledge of language*. New York: Praeger.

Chomsky, N. 1988. Language and problems of knowledge. Cambridge, Mass.: MIT Press.

Chomsky, N. 1991. Some notes on economy of derivation and representation. In R. Freidin (ed.), *Principles and parameters in comparative grammar*. Cambridge, Mass.: MIT Press, pp. 417-54. [Reprinted in Chomsky (1995)]

Chomsky, N. 1993. A minimalist program for linguistic theory. In K. Hale and S. J. Keyser (eds.), *The view from Building 20*. Cambridge, Mass.: MIT Press, pp. 1-52.

Chomsky, N. 1994. *Bare phrase structure. MIT Occasional Papers in Linguistics* 5. Cambridge, Mass.: MITWPL. [Reprinted in G. Webelhuth (ed.) (1995) *Government and binding theory and the minimalist program*. Cambridge, Mass.: MIT Press, pp. 383-439.]

Chomsky, N. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Chomsky, N. 1998. Minimalist inquires: The framework. *MIT Occasional Papers in Linguistics* 15.

Chomsky, N. 2000. Minimalist Inquiries: the framework. in: R. Martin, D. Michaels & J. Uriagereka (eds.) *Step by Step*. Cambridge, MA: MIT Press, pp. 89-155.

Chomsky, N. 2001a. Beyond explanatory adequacy. *Papers in Linguistics* 20. Cambridge, Mass.: MITWPL.

Chomsky, N. 2001b. Derivation by phase. In M. Kenstowicz (ed.), *Ken Hale*. Cambridge, Mass.: MIT Press, pp. 1-52.

Citko, B. 2000. *The implications of (dynamic) antisymmetry for the analysis of (free) relatives*. [Paper presented at the Workshop on the Antisymmetry Theory, Cortona, May 15-17]

Citko, B. 2002. ATB Wh-Movement and the Nature of Merge. Talk given at *NELS 33*.

Collins, C. 1997. *Local economy*. Cambridge/MA: MIT Press.

Cornell, T. 1999. *Derivational and representational views of minimalist transformational grammar*. University of Tübingen [ms.].

Drury, J. 1998. The promise of derivations: atomic merge and multiple spell-out. *Groninger Arbeiten zur Germanistischen Linguistik* 42, pp. 61-108.

Drury, J. 1999. Movement as Remerge and C-Command as Subderivational Precedence. Paper presented at the *GLOW 1999 Workshops in Postdam*.

Echepare, R. 1997. *The grammatical representation of speech events*. Ph.D. dissertation, University of Maryland.

Epstein, S. & N. Hornstein. 1999. *Working minimalism*. Cambridge/MA: MIT Press.

Epstein, S., E. Groat, H. Kawashima & R. Kitahara. 1998. *A derivational approach to syntactic relations*. Oxford University Press.

Epstein, S., E. Groat, R. Kawashima & H. Kitahara. 1998. *A Derivational Approach to Syntactic Relations.* Oxford: Oxford University Press.

Frank, R. & K. Vijay-Shanker. 1999. *Primitive c-command* [ms.] Johns Hopkins University and University of Delaware.

Fukui, N. & Y. Takano. 1998. Symmetry in Syntax: merge and demerge. *Journal of East Asian Linguistics* 7: 27-86.

Gärtner, H-M. 2002. *Generalized Transformations and Beyond: reflections in minimalist syntax*. Berlin: Akademie Verlag.

Goodal, G. 1987. *Parallel Structures in Syntax*. Cambridge UP, Cambridge.

Guimarães, M. 1999. Phonological Cascades and Intonational Structure in Dynamic Top-Down Syntax. Talk given at the *Fall 1999 UMCP Linguistics Dept Student Conference*. College Park, MD, March 14th; available at [http://www.ling.umd.edu/Events/StudentConference/1999.html]

Guimarães, M. 2001. *Syntactic amalgams in dynamic top-down derivations*. [Doctoral research paper] University of Maryland.

Guimarães, M. 2002. Syntactic Amalgams as Dynamic Constituency in Top-Down Derivations. *Proceedings of ConSOLE-X*. Universiteit Leiden; available at http://www.sole.leidenuniv.nl/index.php3?m=1&c=11&garb=0.7125535677557686&session=

Guimarães, M. 2003a. Towards a Descriptively Adequate Theory of Syntactic Amalgams. Talk given at the *1st Joint UMass/UConn/MIT/UMD Syntax Workshop*. Storrs, CT, February 8th. [http://www.linguistics.uconn.edu/sx_wkshp.html]

Guimarães, M. 2003b. *Parallel Derivations that Eventually Converge*. University of Maryland at College Park, ms.

Guimarães, M. 2003c. Hidden Superiority in Syntactic Amalgams. Talk given at the *13th Colloquium on Generative Grammar*. Ciudad Real, Spain, April, 2nd-4th.

Guimarães, M. 2003d. Effects of Pied-Piping on Syntactic Amalgams in Romance. Talk given at the 33rd *Linguistic Symposium on Romance Languages*. Bloomington, IN, April 24th-27th.

Guimarães, M. 2003e. *A Preliminary Investigation on the Semantics of Multiply-Rooted Phrase-Markers*. University of Maryland at College Park, ms.

Hornstein, N. 1995. *Logical Form: from GB to Minimalism*. Oxford: Blackwell.

Hornstein, N. 2001. *Move! A Minimalist Theory of Construal*. Oxford: Blackwell.

Huck, G. & J. Goldsmith. 1995. *Ideology and Linguistic Theory: Noam Chomsky and the Deep Structure Debates (history of linguistic thought)*. London: Routledge.

Jaeggli, O. 1981. Topics in Romance Syntax. Dordrecht: Foris.

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

Kitahara, H. 1997. *Elementary Operations and Optimal Derivations*. Cambridge, MA. MIT Press.

Kuroda, K. 2000. *Foundations of Pattern Matching Analysis, A New Method Proposed for the Cognitively Realistic Description of Natural Language Syntax*. Ph.D. dissertation. Kyoto University. downloadable from: http://clsl.hi.h.kyoto-u.ac.jp/~kkuroda/

Lakoff, G. 1974. Syntactic Amalgams. *Papers from the 10th Meeting of the Chicago Linguistic Society*.

Lasnik, H. 1991. On the necessity of binding conditions. In: R. Freidin (ed.) *Principles and Parameters in Comparative Grammar*. MIT Press, pp. 7-28.

Levinson, S. 1983. *Pragmatics* (Cambridge Textbooks in Linguistics series). Cambridge: Cambridge University Press.

May, R. 1985. *Logical Form: its structure and derivation*. Cambridge, MA: MIT Press.

McCawley, J. 1982. Parentheticals and Discontinuous Constituent Structure. *Linguistic Inquiry*, 13: 91-106.

McCawley, J. 1987. Some Additional Evidence for Discontinuity. In: G. Huck & A. Ojeda (eds.) *Syntax and Semantics 20: Discontinuous Constituency*. New York: Academic Press, 185-200.

Merchant, J. 2001. *The Syntax of Silence: sluicing, islands, and the theory of ellipsis*. Oxford: Oxford University Press.

Moltmann, F. 1992. *Coordination and Comparatives*. Ph.D. dissertation, MIT.

Muadz, H. 1991. *Coordinate Structures: a planar representation*. Ph.D. dissertation, University of Arizona.

Müller, G. 1998. *Incomplete category fronting: a derivational approach to remnant movement in German*. Dordrecht: Kluwer.

Newmeyer, F. 1996. *Generative Linguistics: History of Linguistic Thought*. London: Routledge.

Phillips, C. 1996. *Order and Structure*. Ph.D. dissertation, MIT.

Phillips, C. 2003. Linear Order and Constituency. *Linguistic Inquiry* 34: 37-90

Richards, N. 1999. Dependency formation and directionality of tree construction. *MIT Working Paper in Linguistics* 34 (Papers on morphology and syntax: cycle two.

Richards, N. 2003. Very Local A' Movement in a Root-First Derivation. in: S. Epstein & D. Seely (eds.) *Derivation and Explanation in the Minimalist Program*. Oxford: Blackwell.

Rizzi, L. 1990. *Relativized Minimality*. Cambridge, MA: MIT Press.

Ross, J. R. 1969. Guess Who?. in: R. Binnick et alii. (eds.) *Papers from the 5th Regional Meeting of the Chicago Linguistic Society*. Chicago: Chicago Linguistic Society, 282-286.

Tsubomoto A. & J. Whitman. 2000. A type of head-in-situ construction in English. *Linguistic Inquiry* 31, pp. 176-183.

Tsubomoto A. & J. Whitman. 2000. A Type of Head-in-Situ Construction in English. *Linguistic Inquiry* 31: 176-183.

Uriagereka, J. 1999. Multiple spell-out. S. Epstein & N. Hornstein (eds.) *Working Minimalism*, Cambridge/MA: MIT Press, pp. 251-282.

Uriagereka, J. 2002. *A mind plan?* Colloquium talk at Postdam University.

van Riemsdijk, H. 2000. *Free Relatives*. Tilburg University, ms.

Watanabe, A. 1995. Conceptual basis of Cyclicity. *MIT Working Papers in Linguistics*, 27, pp. 269-291.

Weinberg, A. 1995. Parameters in sentence processing: the case of Japanese. R. Mazuka & N. Nagai (eds.) *Japanese sentence processing*. Erlbaum.

Wilder, C. 1998. Transparent free relatives. *ZAZ Working Papers in Linguistics*, pp. 191-199.

Wilder, C. 1999. Right Node raising and the LCA. *Proceedings of WCCFL* 18.