3
Combining linguistic theories in a Minimalist Machine

Jason Ginsburg and Sandiway Fong

3.1 Introduction

Theoretical syntacticians, working within the framework of Generative Grammar, investigate the nature of the human grammatical system (i.e. Universal Grammar). Most of this work involves developing theories that predict and explain the structures and well-formedness/ill-formedness of constructions found in natural language. Theoretical linguists generally do not use computer models to test the validity of their theories. In this chapter, we show how a computer model can be a useful tool for combining components of different theories into a single unified model, which also clarifies the inner workings of the target theories.

We have developed a Minimalist Machine that automatically constructs detailed derivations of over 140 examples from recent important work in the Minimalist Program. We discuss particular modifications that are necessary to combine the target linguistic theories into a single unified model. This is an important and necessary step towards making the Minimalist Machine viable because there is no guarantee that theories in the same framework are indeed compatible; for example, they might rely on contradictory assumptions.

This Minimalist Machine is a useful tool for finding and overcoming problems that arise in minimalist theories. In order for the Minimalist Machine to construct the successful derivation of a target sentence, all operations must converge successfully. Inconsistencies in the model will cause a derivation to crash, or the computer program to halt, and thus, problems that otherwise might be difficult to notice become immediately apparent. In a derivational theory, both crashes and computation convergence are key components of the framework that communicate important information about grammar. Convergence will admit valid examples. Crashes are used to rule out illicit examples. In other words, one can use this system to figure out why a particular derivation does or does not converge. Hence, it is important to demonstrate that the grammar does "compute."

The Minimalist Machine has the unique property of demonstrating how minimalist theories work in extreme detail, beyond what is normally possible via typical paper-and-pencil linguistics. This model automatically constructs detailed step-by-step

Minimalist Parsing. First edition. Robert C. Berwick and Edward P. Stabler (eds.) First published 2019 by Oxford University Press.

derivations of target constructions that can be read in a web browser. This output displays charts and tree diagrams that show all core feature-checking operations, Merge operations, and Spell-Out operations (see http://elmo.sbs.arizona.edu/sandiway/mpp/mm.html). This model is thus useful for clarifying exactly how derivations work, as well as for elucidating problems within target minimalist theories, so this work can benefit theory development in linguistics.

The notable components of this probe-goal model[1] are as follows. The model utilizes feature unification, which simplifies cases of multiple agreement, in which a single probe must Agree with multiple goals. The model incorporates a low-Merge position for the expletive *there*, thus accounting for how *there* is licensed in certain core expletive constructions. To account for thematization/extraction, we utilize a light verbal head that must be checked via Agree with a theta-bearing DP. Edge Features, specified for various functional heads, play a role in driving movement. Economy also plays a role in derivations, so that simpler operations block more complex operations—if multiple features on a head can be checked via Agree with a single goal vs. Agree with multiple goals, the more economical Agree relation with the single goal is required. Economy plays an important role in accounting for *that*-trace effects and the presence/absence of *that* in relative clauses.

We demonstrate how this model implements the following types of target constructions in a unified manner: expletives and multiple agreement (Chomsky 2001), thematization/extraction (Chomsky 2001; Sobin 2014), *that*-trace effects and subject vs. object *wh*-movement (Pesetsky and Torrego 2001), and relative clauses (Pesetsky and Torrego 2001; Gallego 2006). We point out certain problems that arise in previous work on these types of constructions, and we explain how we resolved these problems in our computational implementation. Also, it is important to note that we created a single computational implementation that can account for the structures of all of these types of sentences, as well as others (which we do not have space to discuss).

3.2 Multiple Agree (Chomsky 2001)

Chomsky (2001) develops an account of language in which agreement between a probe and a goal is crucial. In Chomsky (2001), however, problems arise with respect to multiple agreement—cases in which a probe must Agree with more than a single goal.

Agreement works as follows. A probe, which is the root of a Syntactic Object (SO), has an unvalued feature (uF) that searches for a goal with a matching valued feature (F). Crucially, the probe must c-command the goal. Case checking results as a reflex of a phi-feature-checking relation. A probe with unvalued phi-features (uPhi) Agrees with a goal that has valued phi-features (Phi) and unvalued Case (uCase). Agreement checks the uPhi on the probe and the uCase on the goal. For example, as shown in (1),

[1] This model does not incorporate the recent labeling-based framework of Chomsky (2013; 2015a). For discussion of labeling-based models, see Ginsburg (2016) and Fong and Ginsburg (in preparation).

given a probe X that has uPhi and the ability to check Case as nominative (Nom), and given a goal Y that has Phi and uCase, Agree between X and Y results in checking of the uPhi on X. As a reflex of this Agree relation, the uCase on Y is valued as nominative.

(1) $\text{Agree}(X_{[\text{uPhi,Nom}]}, Y_{[\text{Phi,uCase}]}) \rightarrow X_{[\cancel{\text{uPhi}},\text{Nom}]} \text{ and } Y_{[\text{Phi,Case:Nom}]}$

Certain issues arise with respect to checking of uCase on a participle. Chomsky (2001: 17), citing the fact that certain past participles show agreement in languages such as Icelandic, takes the position that a past participle requires Case. Thus, in the passive (2), the participle *-ed* (which combines with *award*) must obtain Case.

(2) Several prizes are likely to be awarded. (Chomsky 2001: 7)

The relevant portion of the derivation of (2) is shown in Figure 3.1. A participle, indicated as prt, has a defective set of uPhi, containing unvalued number (uNum) and unvalued gender (uGen), as well as uCase. The prt is defective in that it does not have a complete set of uPhi, as it lacks uPerson. Prt Agrees with the object *several prizes*. The uNum and uGen of prt are checked by the valued number and gender, Num and Gen, of the object *several prizes*. The uCase on prt remains because the object, also an argument, cannot check uCase. The non-finite T, *to*, is defective. It has an EPP feature and an incomplete set of uPhi, which Chomsky (2001: 7) suggests is uPerson. Non-finite T, being defective, cannot check uCase. The non-finite T Agrees with *several prizes*, resulting in checking of uPerson (indicated as uPer) on *to*, and an EPP on *to* forces the object to remerge in embedded subject position. The matrix T has uPhi and can check Case, which it values as nominative. In this case, T forms a

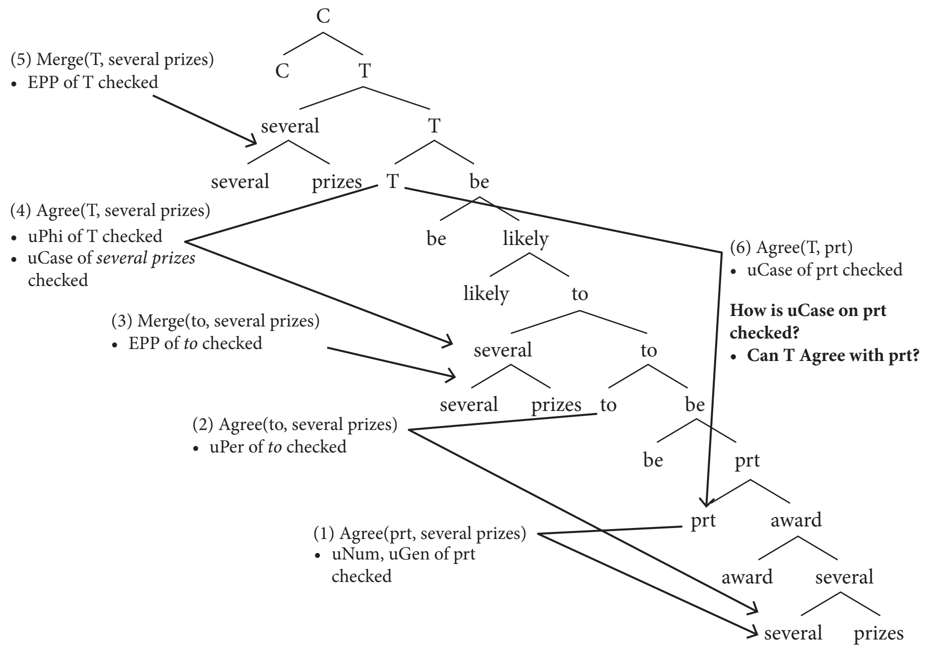


Figure 3.1 Structure of (2)

probe-goal Agree relation with *several prizes*, resulting in uPhi of T being checked. As a reflex of this phi-feature agreement relation, the uCase of *several prizes* is checked and valued nominative. An EPP feature on T also forces *several prizes* to remerge in subject position.

This derivation, however, faces a problem with respect to the participle—it is not clear how the uCase of the participle is checked. In order for uCase on prt to be checked, the matrix T must form an Agree relation with prt, shown in step (6) of Figure 13.1. But the uPhi on T are already checked via Agree with the higher object *several prizes*, shown in step 4. Thus, after T forms an Agree relation with *several prizes*, T has no reason to continue probing and further Agree with prt. Even if T were to Agree with prt, it is notable that the only phi-features that prt has are checked number and gender, raising the question of whether or not checked uPhi of prt are capable of further phi-feature agreement with T.

In order to generate an example such as (2), in which there is multiple agreement, it is necessary to determine exactly how the uCase of prt is checked. To this end, we implemented feature unification. If a probe with a uF Agrees with a goal with an identical uF, the uFs unify.

(3) *Feature unification*
uFs of the same type unify.

With feature unification, checking of uCase on prt is no longer a problem. The derivation of (2) proceeds as shown in Figure 13.2. An Agree relation established between prt and *several prizes* results in checking of uNum and uGen of prt by the

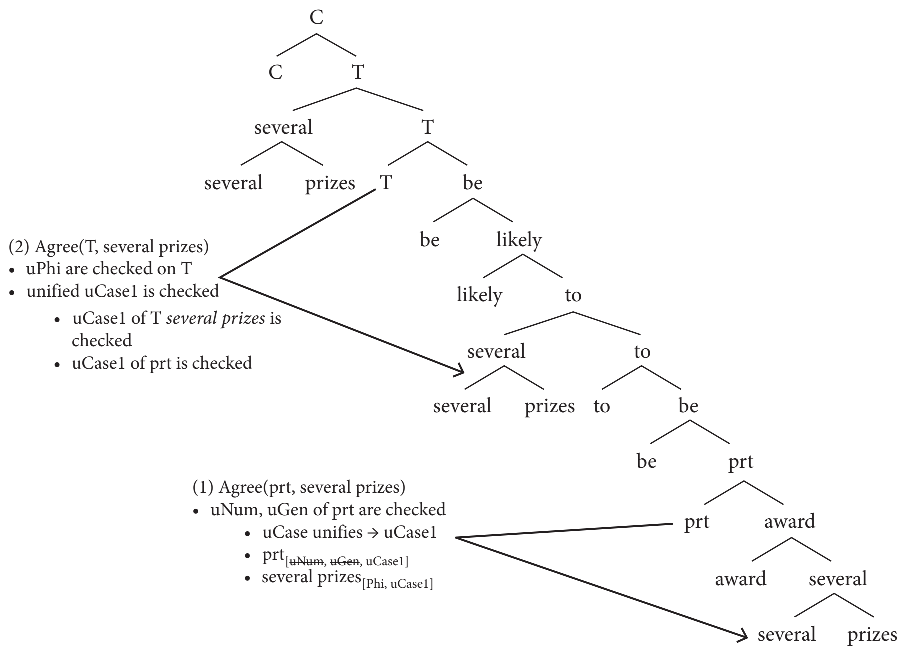


Figure 3.2 Structure of (2) with feature unification

Phi of *several prizes*. Note that both prt and *several prizes* have uCase. Thus, the uCase of prt unifies with the uCase of *several prizes*, indicated as uCase1. This is shown in step (1) of Figure ~~13.2~~. When the matrix T is Merged, shown in step (2), T Agrees with the object *several prizes*, resulting in checking of uPhi (which includes uNum, uGen, and uPer) on T and checking of uCase1 on *several prizes*. Crucially, checking of uCase1 on *several prizes* also results in checking of the unified uCase1 on prt. Thus, feature unification crucially does away with the need for T to probe after it is licensed. In our Minimalist Machine model, we implemented feature unification in this manner.

3.3 Expletives (Chomsky 2001)

Expletives raise some problems for Chomsky's (2001) account. According to Chomsky (cf. p. 16), the expletive (Expl) *there* is a defective argument that has uPerson features and that lacks uCase. The Expl is also Merged directly in the Spec position of T and the uPerson feature of *there* is checked via Agree with T. Several issues arise in (4), with the structure shown in Figure 3.3, in which *there* appears in subject position. First, Agree between uPhi of T and uPerson of *there* checks the uPerson of *there*. Note that this Agree relation holds between uFs (uPhi of T and uPerson of Expl), but it is not clear how a uF can check a uF. Second, when T is Merged, T should form a probe-goal Agree relation with the object *a train*. Crucially, this Agree relation between T and *a train* is formed before *there* is Merged in Spec position. Thus, after agreement with the object, T should not have any uPhi available for Agree with *there*. One possibility, pointed out by Chomsky (2000: 128), is that the Expl *there* probes and Agrees with T, but it isn't clear if the checked uPhi on T could check the uPerson of the Expl. Another issue that arises is whether or not a non-root node can probe. The root node of the SO is T, so if Expl probes from Spec position, then there is probing from a non-root node. Richards (2007) points out that probing from a non-root node is problematic—permitting both root and non-root nodes to probe is more complex than limiting probing to a root node. Richards also points out that probing by Expl would go against Chomsky's (2008) proposal that only phase heads can probe.

(4) There arrived a train. (Sobin 2014: 386)

The feature-checking and probing problems raised by expletive constructions are resolved if (a) we assume that an expletive is Merged below T and (b) we utilize feature unification, as described in (3) above. We follow Richards (2007), who, in order to avoid the problem of probing from a non-root node, proposes that *there* is Merged below T.[2] The successful derivation is shown in Figure 3.4. Assume that *there* is Merged as the specifier of a light v projection below T. We assume that this v has an Edge Feature (EF) that requires it to have an element Merged in specifier position.[3] Then when T is

[2] This view is also adopted by Sobin (2014), among others.
[3] An Edge Feature (EF) is akin to an EPP feature, and is checked via Merge with any argument (Chomsky 2008).

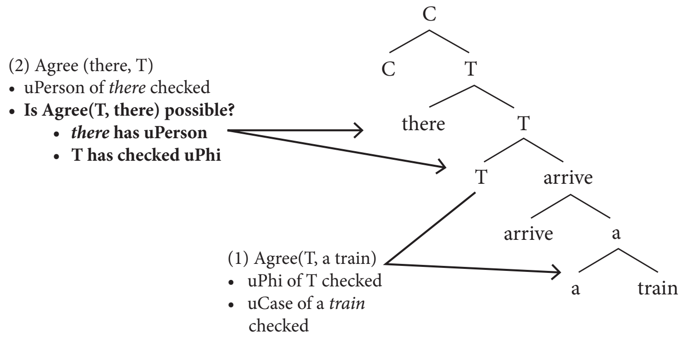


Figure 3.3 Problems with (4)

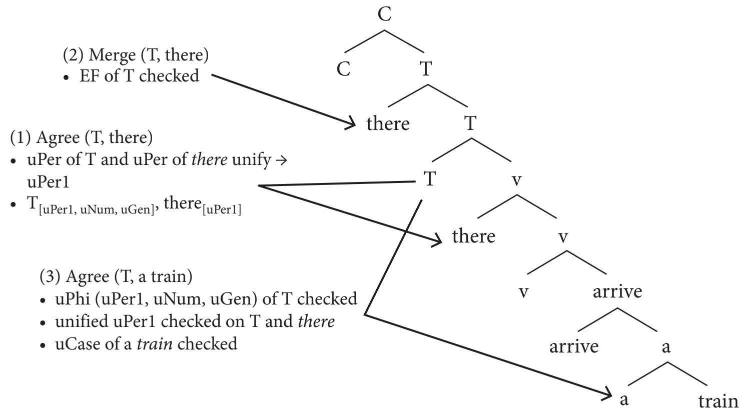


Figure 3.4 Structure of (4) with low Merge of Expl and feature unification

Merged, uPerson of T Agrees with uPerson of *there*, resulting in feature unification, so that T and *there* have unified uPerson features, indicated as uPer1 in step 1 of Figure 3.4. Since *there* is defective, it cannot check any uPhi of T. Thus, T continues to probe and Agrees with the object *a train*. Agree results in checking of the uPhi on T. Crucially, in addition to uNumber and uGender, the uPhi includes the unified uPerson feature. Thus, when the uPerson feature of T is checked, the unified uPerson on *there* is also checked. As a reflex of this Agree relation, uCase on the object is also checked. Note that only T probes, so there is no probing from a non-root node. Since the uPerson of *there* is checked via feature unification, there is no need to rely on uPerson of *there* being checked via Agree with T. Also note that an EF of T forces *there* to raise to the Spec position of T.

3.4 Thematization/Extraction (Th/Ex) (Chomsky 2001; Sobin 2014)

Chomsky (2001: 20) presents the following example in (5), in which the direct object "is extracted to the edge of the construction by an obligatory thematization/extraction rule Th/Ex." The direct object *several packages* undergoes Th/Ex to a preverbal position, as can be seen in ~~(5)b~~. Note that if the object remains in its base position, the result is ill-formed, as shown in (5c), indicating that Th/Ex is obligatory.

(5) a. There were several packages placed on the table. (Chomsky 2001: 20)
b. There were [several packages] placed ~~[several packages]~~ on the table.
c. *There were placed several packages on the table.

Chomsky suggests that Th/Ex is phonological, writing "Th/Ex is an operation of the phonological component" (2001: 21), without offering specifics about how Th/Ex would work as a PF (Phonological Form) phenomenon. However, we follow Rezac (2006) and Sobin (2014), who take a syntactic approach to Th/Ex. Sobin, following Rezac (2006), writes: "Th/Ex positions are the specifier positions of those functional v heads that are θ-open and allow an argument to appear in them" (2014: 398).[4]

Sobin's approach relies on a Split-EPP feature, which has Agree and Merge subfeatures. In order for a Split-EPP feature to be fully checked, the Agree and Merge subfeatures must be checked. The Agree subfeature can take the form of *u*CheckTheta (6), which is checked via Agree with a theta-bearing DP. The Merge subfeature is checked via Merge of an element in specifier position, and can have the form $u\mathrm{D}_{\mathrm{Mrg}}$, shown in (7), which requires Merge of any type of DP (an expletive or a theta-bearing DP are fine).[5]

(6) *Agree subfeature*
*u*CheckTheta—checked via Agree with a theta-bearing DP

(7) *Merge subfeature*
$u\mathrm{D}_{\mathrm{Mrg}}$—checked by Merge of a DP

Adopting Deal's (2009) proposed default verbalizing head v∼ that appears in unaccusative, passive, and progressive constructions, Sobin proposes that English v∼ has the EPP subfeatures shown in (8). It has a *u*CheckTheta, so it must Agree with a theta-role bearing DP, and it also has a $u\mathrm{D}_{\mathrm{Mrg}}$ feature, so a DP must Merge with it in specifier position. Crucially, $u\mathrm{D}_{\mathrm{Mrg}}$ can be checked by any type of DP, including an Expl.

[4] See Rezac (2006) for discussion of why Th/Ex is syntactic. For example, Rezac argues that Th/Ex positions can be iterated, which requires Th/Ex to apply before PF, as a derivation is constructed. Also, in Swedish, a passive participle shows agreement with an argument that appears to have undergone Th/Ex. Assuming that agreement is syntactic in nature, then this too suggests that Th/Ex is not a PF phenomenon.

[5] We only include the Agree and Merge subfeatures that are relevant for the examples presented in this chapter. The full set of Agree and Merge subfeatures utilized by Sobin are as follows.

(i) Agree subfeatures
a. *u* CheckTheta—checked via Agree with a theta-bearing DP
b. $u\mathrm{D}_{\mathrm{Agr}}$—checked via Agree with a DP (any DP is okay, including expletive)

(ii) Merge subfeatures
a. $u\mathrm{D}_{\mathrm{Mrg}}$—checked by Merge of a DP
b. $u\mathrm{Theta}_{\mathrm{Mrg}}$—checked by Merge of a theta-role bearing DP
c. $u\mathrm{D}_{\mathrm{MrgExpl}}$—checked by Merge of an Expl

Sobin proposes that light verbs can have different combinations of the Agree and Merge subfeatures. See Sobin (2014) for further details.

(8) v~ has [*u*CheckTheta, uD_{Mrg}] (Sobin 2014: 393)
  a. *u*CheckTheta is checked via Agree with a theta-bearing DP
  b. uD_{Mrg} is checked via Merge of a DP

The v~ head provides a way to account for Th/EX in English.

The split-EPP feature of v~ accounts for the possibility of either the Expl *there* or the object *a train* appearing in subject position in (9a,b).

(9) a. There arrived a train.
  b. A train arrived. (Sobin 2014: 393)

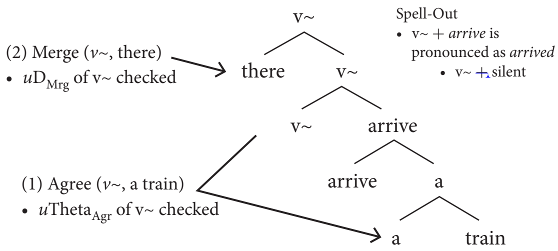


Figure 3.5 Derivation of (9a)

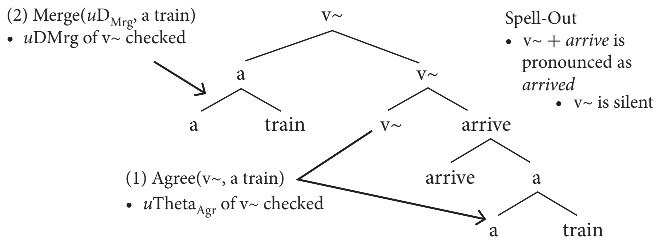


Figure 3.6 v~ in (9b)

In (9a as shown in Figure 3.5, when v~ is Merged, its *u*CheckTheta subfeature Agrees with the theta-role bearing object *a train*. Then, when the Expl *there* is Merged, the uD_{Mrg} subfeature of v~ is checked. Note that uD_{Mrg} can be checked via Merge of any DP, regardless of whether or not it bears a theta-role. Thus, Expl is sufficient.

The structure of (9b) is shown in Figure 3.6. Assume that there is no Expl available for Merge. Thus, both the $uTheta_{Agr}$ and uD_{Mrg} subfeatures of v~ are checked by the same object, *a train*. In this case, the need to check the uD_{Mrg} subfeature of v~ results in the object *a train* undergoing Th/Ex.

Note that in (9a,b), the v~+*arrive* complex is pronounced as *arrived* (assuming that a past-tense T is Merged). The v~ has no pronunciation, which turns out to be an issue, since v~ is pronounced in other examples (see below).

The progressive construction with an expletive (10) has two instances of v~, as shown in Figure 3.7, where prog represents a progressive head that combines with v~ and a verbal root to form an *-ing* construction. The lower v~1 has both its $uTheta_{Agr}$ and uD_{Mrg} subfeatures checked by *a train*. Checking of the uD_{Mrg} subfeature forces *a train* to remerge in specifier position of v∼1, and thus undergo Th/Ex. The *u*CheckTheta subfeature of the higher v∼2 then is checked via Agree with *a train*. Then Merge of an Expl with v∼2 satisfies the $u\mathrm{D}_{\mathrm{Mrg}}$ subfeature of v∼2, thus also preventing *a train* from moving higher.

(10) a. There is a train arriving. (Sobin 2014: 399)
b. A train is arriving.

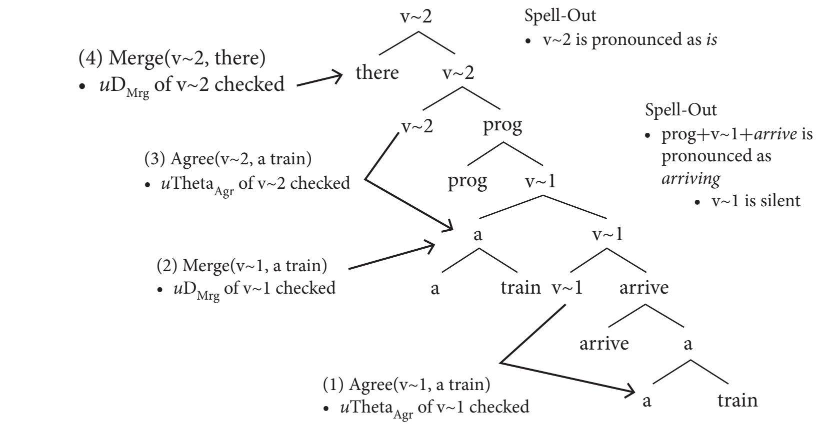


Figure 3.7 v∼ in (10a)

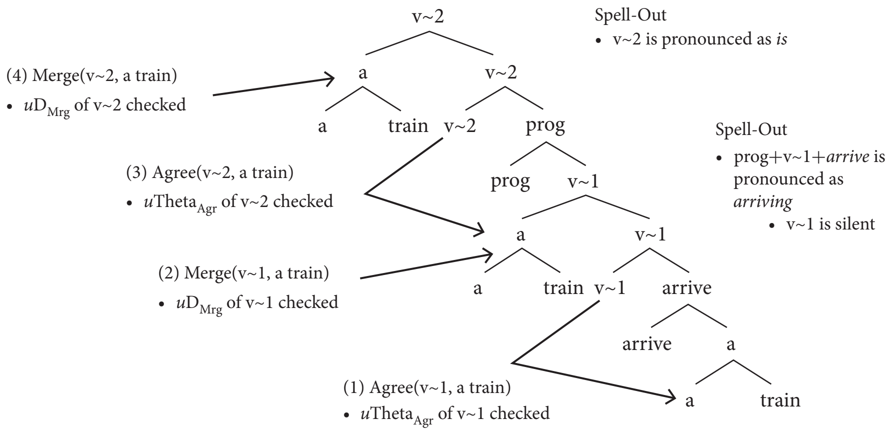


Figure 3.8 v∼ in (10b)

If the derivation lacks an Expl, then the structure shown in Figure 3.8 results. In this case, the object *a train* moves all the way to matrix subject position, due to the lack of an Expl, to check the $u\mathrm{D}_{\mathrm{Mrg}}$ subfeature of v∼2. Thus, *a train* checks the $u\mathrm{Theta}_{Agr}$ and $u\mathrm{D}_{\mathrm{Mrg}}$ features of both v∼1 and v∼2.

Note that pronunciation of v∼ in (10a,b), as shown in Figures 3.7 and 3.8, is complex because the higher v∼2 is pronounced as *is* but the lower v∼1 lacks any pronunciation, assuming that the progressive head Prog and v∼ are pronounced as *arriving*. But it isn't clear how to distinguish the pronunciations of v∼.

Sobin's analysis also accounts for the impossibility of (11), as shown in Figure 3.9. In this case, the $u\text{Theta}_{Agr}$ of v∼1 is checked via Agree with the object *a train*, and the $u\text{D}_{\text{Mrg}}$ feature of v∼1 is Checked by Merge of the Expl *there*. A problem then arises when v∼2 is Merged because the Expl blocks the $u\text{Theta}_{Agr}$ subfeature of v∼2 from Agreeing with *a train*; since an Expl lacks a theta-role, it cannot check $u\text{Theta}_{Agr}$. Sobin (2014) argues that this type of example is ruled out because a phase boundary intervenes between v∼2 and its potential goal. But even if there were no phase boundary, this could be ruled out as an intervention effect caused by the Expl.

(11) *There is arriving a train.

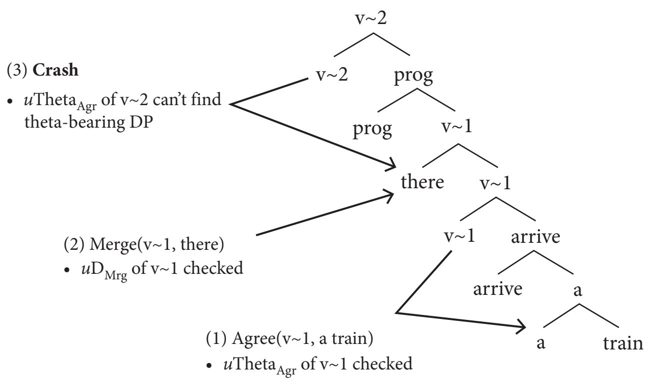


Figure 3.9 Structure of (11)

While Sobin's analysis accounts for Th/Ex of an object to preverbal position, as well as for the distribution of the Expl in English, there are some issues that arise with respect to Spell-Out. In (10a,b), as shown in Figures 3.7 and 3.8, the higher v∼1 is pronounced as *is* but the lower v∼2 lacks any pronunciation, as it combines with *arrive* and the progressive, to form *arriving*. Thus, sometimes v∼ is pronounced, and other times it is not, as shown in Figure 3.10. If both of these are identical v∼ elements, then the question arises of how the Spell-Out component determines the correct pronunciation of v∼. Another problem arises from the placement of *a train* in constructions such as (10a), in which *a train* undergoes Th/Ex to preverbal position. As a result, *a train* intervenes between the progressive head prog and the lower v∼1, which need to combine to form *arriving*, as shown in Figure 3.10.

These Spell-Out problems are resolved in our model as follows. First, since sometimes v∼ is pronounced and other times it is not, we propose that there are actually two different types of v∼. One type, (12a), which we refer to as v∼, is simply a light verb that is pronounced as a form of *be*, (12a). Another type, (12b), which we refer to as v∼unacc, Merges with an unaccusative verbal root, and lacks any pronunciation. Both types of v∼ have the same Agree and Merge features, (12c). We utilize CheckTheta, which is essentially the same as Sobin's $u\text{Theta}_{Agr}$, to refer to a feature that is checked by a theta-bearing argument. Instead of $u\text{D}_{\text{Mrg}}$ we simply utilize an Edge Feature (EF). To deal with the affix-hopping problem, in which an object intervenes between the progressive and v∼, we assume that the progressive head (prog) has an EF, (13). Thus, the EF on

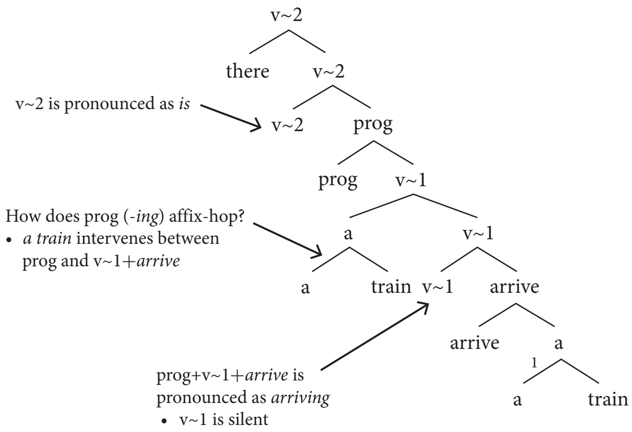


Figure 3.10 v∼ pronunciation in (10a)

prog forces an object to move to the edge of prog, which enables affix-hopping to occur without an intervening overt argument.

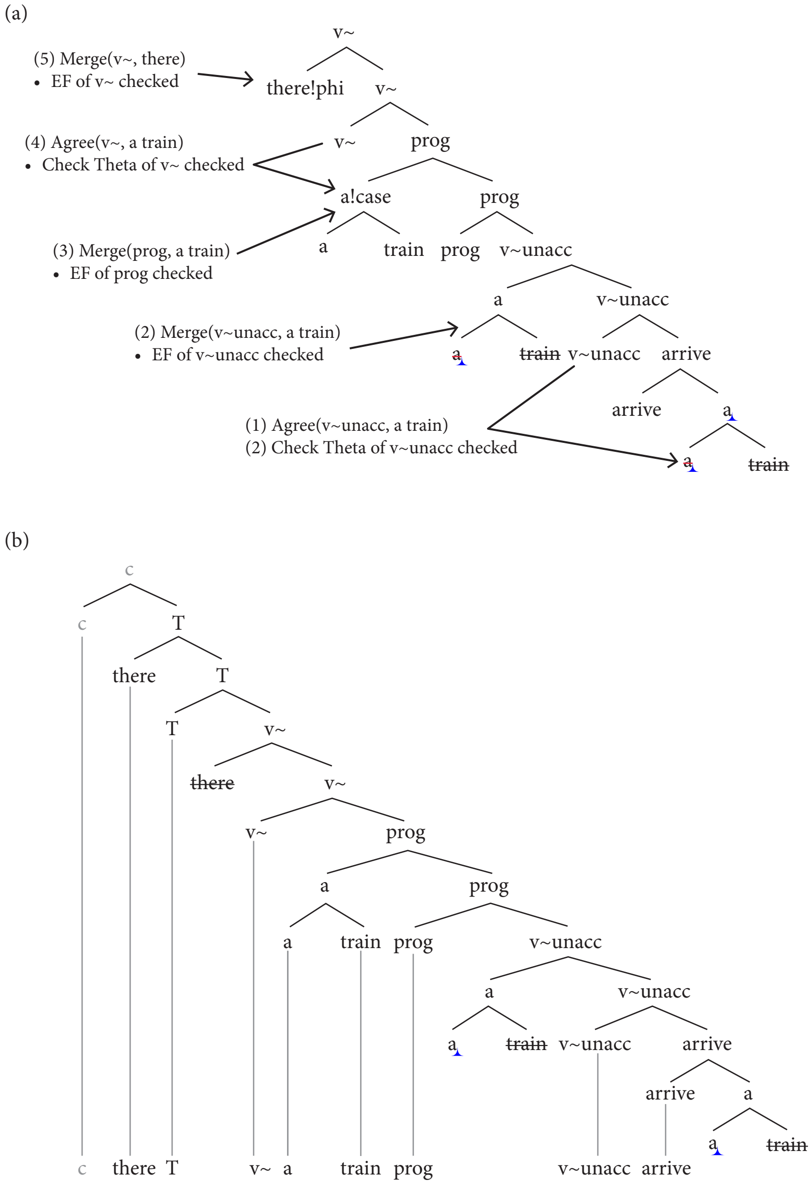
(12) v∼ types
  a. v∼ is pronounced as *be*
  b. v∼unacc has no pronunciation (v∼unacc Merges with an unaccusative verbal root)
  c. v∼ and v∼unacc have CheckTheta and EF

(13) prog has an EF

With these proposals, we next explain how we implemented Sobin's v∼ analysis with our computer model. The derivation of (10a), as generated by our model, is shown in Figure 3.11. This construction contains a v∼unacc and a separate v∼. The CheckTheta feature of v∼unacc is checked via Agree with the object *a train* and the EF feature of v∼unacc is checked via remerge of *a train*. Then the higher prog head is Merged, and prog has an EF that forces *a train* to remerge again in specifier position of prog. Note that because the object moves to the Spec of the prog projection, it no longer intervenes between prog and the v∼unacc+*arrive* complex, so that prog can affix-hop onto v∼unacc+*arrive*. Then the higher v∼ is Merged, and the CheckTheta of v∼ is checked via Agree with *a train*, which is in the Spec of the next lower projection, and thus is accessible. The EF of v∼ is checked via Merge of the Expl *there*. The Expl has uPhi (unvalued Phi), indicated as !phi in the figure. These uPhi are eventually checked via Agree with a higher T. The final structure and Spell-Out steps are shown in Figure 3.11b. Crucially, the higher v∼ is pronounced as a form of *be* and the lower v∼unacc is silent, and prog affix hops onto v∼unacc+arrive, forming *arriving*.

With our assumptions about v∼ combined with our previous assumptions about feature unification, we can now account for the core examples of Chomsky (2001), including those with Th/Ex.

The derivation of (2), repeated below as (14), is shown in Figure 3.12.



Spell-out:
there -s be a train -ing arrive (after morpheme realization)
there be -s a train arrive -ing (after affix-hop)
there is a train arriving

Figure 3.11 Derivation of (10a)

(14) Several prizes are likely to be awarded. (Chomsky 2001: 7)

When prt is Merged, Agree between prt and *several prizes* results in unification of uCase of prt and *several prizes*, as well as checking of uNumber and uGender of prt. An EF of prt forces *several prizes* to remerge in Spec position. Then when v∼ is Merged, the CheckTheta feature of v∼ is checked via Agree with the theta-role bearing *several prizes*. An EF on v∼ also forces *several prizes* to remerge in Spec position. Next, the

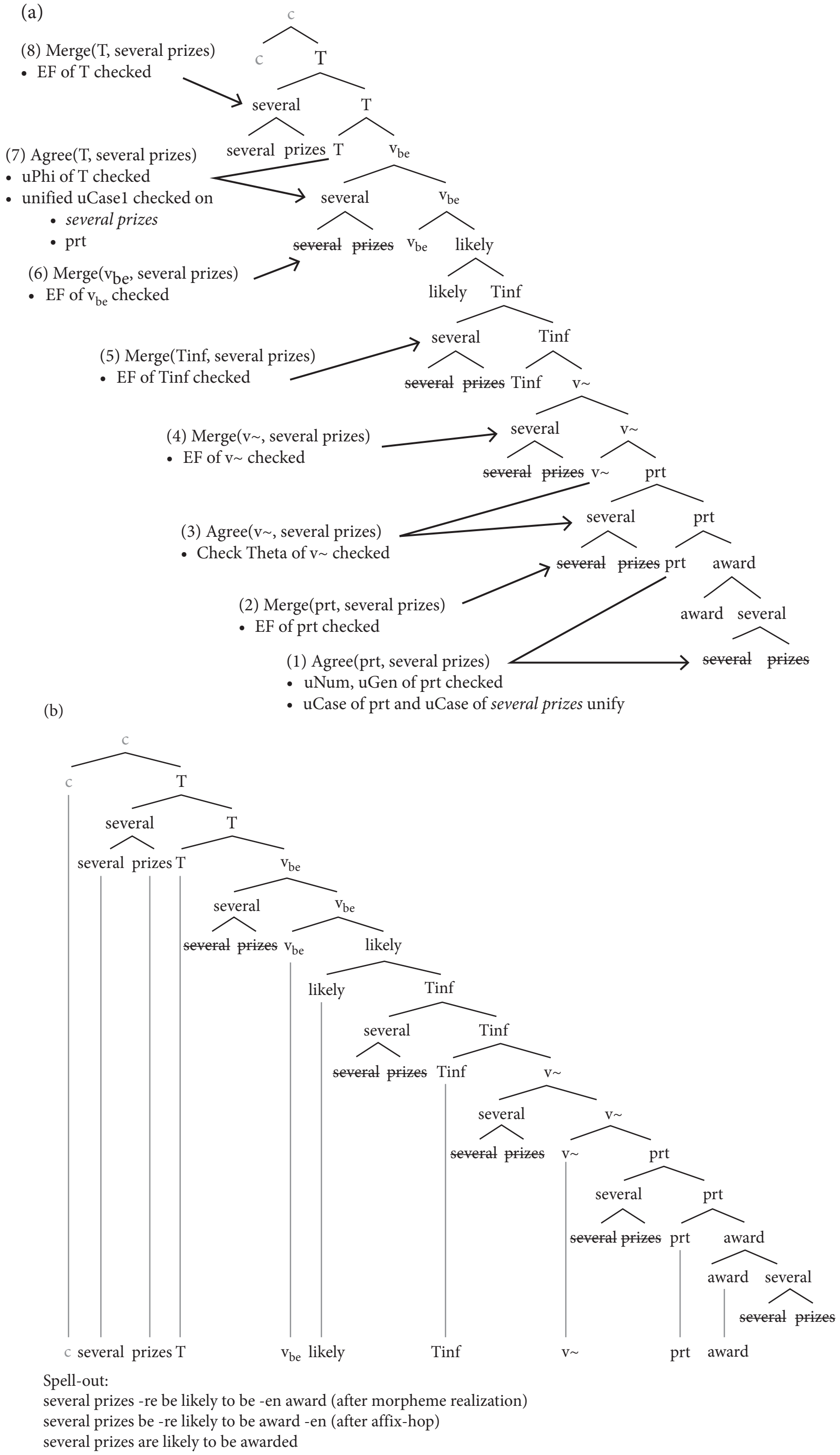


Figure 3.12 Derivation of (14)

non-finite Tinf has an EF that forces *several prizes* to remerge in Spec position. The higher light verb *be*, v_{be}, also has an EF that forces remerge of *several prizes*. When the matrix T is Merged, T Agrees with *several prizes*, resulting in checking of the uPhi of T and the uCase of *several prizes*. Crucially, the uCase of *several prizes* is unified with uCase of prt, so the unified uCase of prt is also checked. Then an EF of T is checked by reemerge of *several prizes* and the derivation eventually converges. The final tree and Spell-Out are shown in Figure 3.12b.

We next explain how we extended this model to account for completely different types of examples (*that*-trace effects, *wh*-movement, and relative clauses).

3.5 *That*-trace effects and subject vs. object *wh*-movement (Pesetsky and Torrego 2001)

Pesetsky and Torrego (2001), hereafter P&T, present an analysis that accounts, in a unified manner, for the English subject/object *wh*-movement asymmetry and the *that*-trace effect. We explain how we implemented their proposals.

P&T's analysis relies on the idea that certain C heads have a uT feature that must be checked in the most economical manner. P&T propose that a uT feature on C can be checked by T or by nominative Case, which is checked uT on D, as summarized in (15). Nominative Case can check uT because Nominative is a checked uT feature.

(15) Certain C heads have a uT feature
 a. uT can be checked by T
 b. uT can be checked by Nominative Case

P&T also propose that T in C is pronounced as *that*. Thus, when *that* appears in an embedded clause, T has checked a uT feature on C. In a *wh*-question, they assume that an interrogative C has uWh and uT features that need to be checked in order for a derivation to converge. From the perspective of economy, Agree between an interrogative C and a single goal that can check both uWh and uT features is most economical. This is stated in (16), where we utilize uQ instead of uWh; we assume that a *wh*-phrase is headed by Q (cf. Hagstrom 1998; Cable 2007) and it has an iQ feature that checks a uQ feature on an interrogative C. If the option of checking both uT and uQ features is not available, then it is possible for C to form two separate Agree relations, one which checks the uT feature and one which checks the uQ feature.

(16) *Economy*
 Given $C_{[uQ,uT]}$, if goal X can check uT but not uQ, and goal Y can check both uQ and uT, then Agree with Y blocks Agree with X.

This analysis accounts for the optionality of *that* in embedded clauses, as in (17).

(17) Mary thinks (that) Sue will buy the book. (P&T 2001: 373)

The embedded clause is shown in Figure 3.13a. The embedded non-interrogative C is indicated as C_e (our notation). P&T assume that C_e has a uT feature. Thus, one option is for the uT feature to be checked via Agree with the subject *Sue*, in which case Nominative Case of the subject checks the uT feature. P&T assume that uT has an EPP subfeature that forces remerge of whatever uT Agrees with. We utilize an Edge Feature (EF), assuming that whatever Agrees with C_e must remerge with the C_e projection to check the EF. Thus, an EF of C_e forces the subject *Sue* to remerge. Since the subject is in C_e, there is no pronunciation of *that*, as can be seen in the final structure, shown in Figure 3.13b. Note that the matrix clause contains what we refer to as v_{unerg}, which occurs with a verbal root such as *think* that takes a clausal complement. This v_{unerg} is an unergative light verb that assigns a subject theta-role, but does not check Case or have uPhi.[6]

Another possibility is for *that* to be pronounced in (17), as shown in Figure 3.14. In this case, the uT feature of C_e is checked via Agree with T and an EF forces T to raise to C_e.[7] T in C is pronounced as *that*, as shown in Figure 3.14b.

We next demonstrate how P&T (2001) account for *wh*-movement in an example such as (18). Although P&T do not discuss this exact example, it is important for demonstrating the need for a verbal element to raise to T.

(18) What will Mary buy?

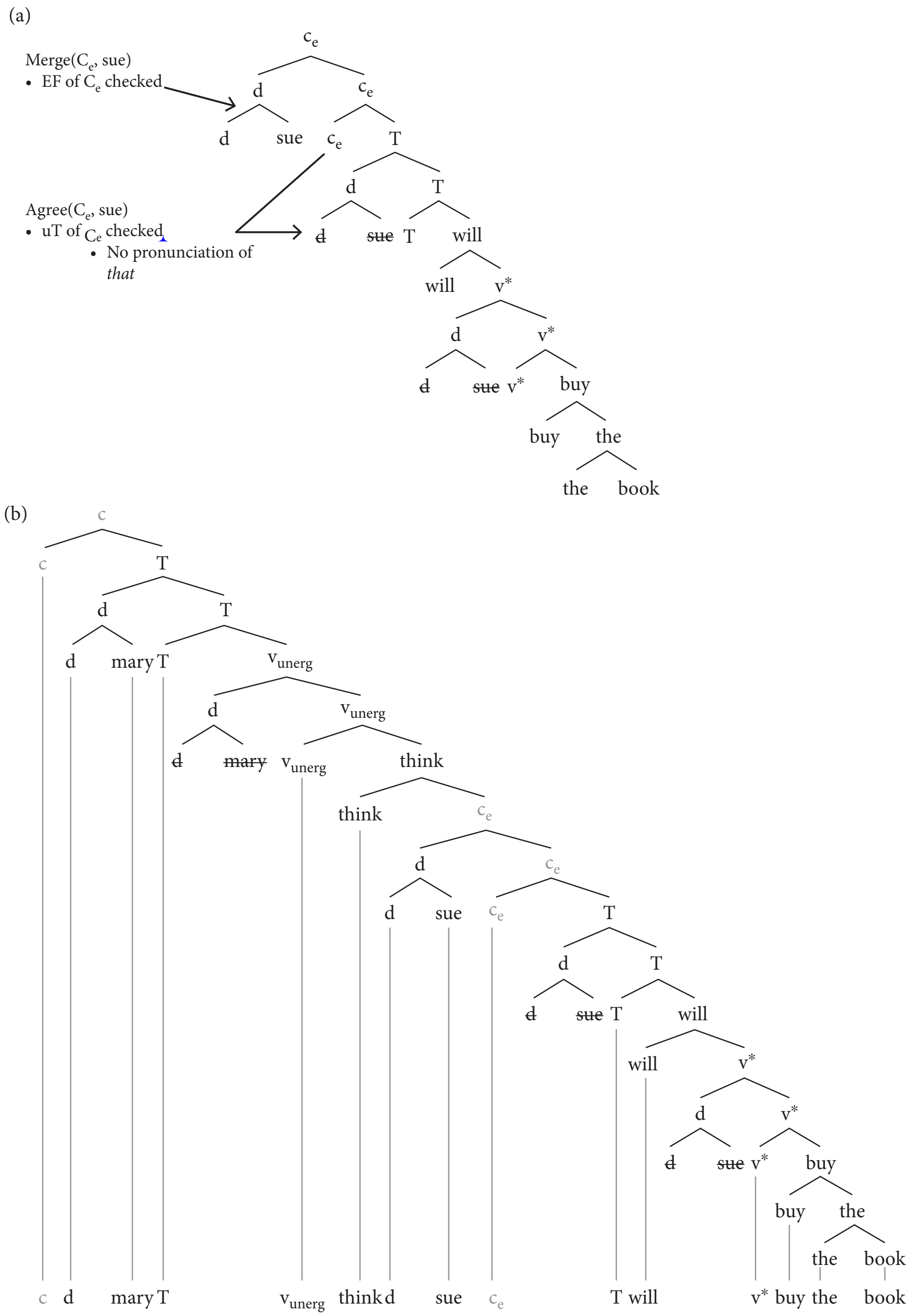
Initially, shown in Figure 3.15a, after T is Merged, the modal *will* raises to T, which is necessary so that the modal can eventually raise to C. We implemented this movement by giving T an unvalued v feature, uv, that requires checking by the closest verbal head. The interrogative C, which we indicate as C_Q, has uQ, uT, and EF features. The EF feature forces whatever Agrees with C_Q to remerge with C_Q. Initially, C_Q forms an Agree relation with T, resulting in checking of uT. An EF on C_Q then forces T to remerge. Furthermore, the modal *will* is pied-piped with T. This has the desired effect of bringing the modal into pre-subject position. Thus, we assume that the highest verbal element raises to T (to check a uv feature) and that when T raises to C_Q, this verbal element also raises. While movement of the verbal element to T may seem like an imperfection, this is necessary to account for the correct word order. If T alone were to move to C, without the highest verbal element, then the result would be the ill-formed (19) in which *will* remains below the subject, and T in C is pronounced as *do*.

(19) *What does Mary will buy?

If there is no overt auxiliary/modal that can raise to T, then the default *do* is inserted. Continuing on, the uQ feature of C_Q is checked via Agree with the Q-phrase *what*, and an EF of C_Q forces *what* to remerge. Note that two operations are required to check the uT and uQ features of C_Q. The derivation then converges as shown in Figure 3.15b.

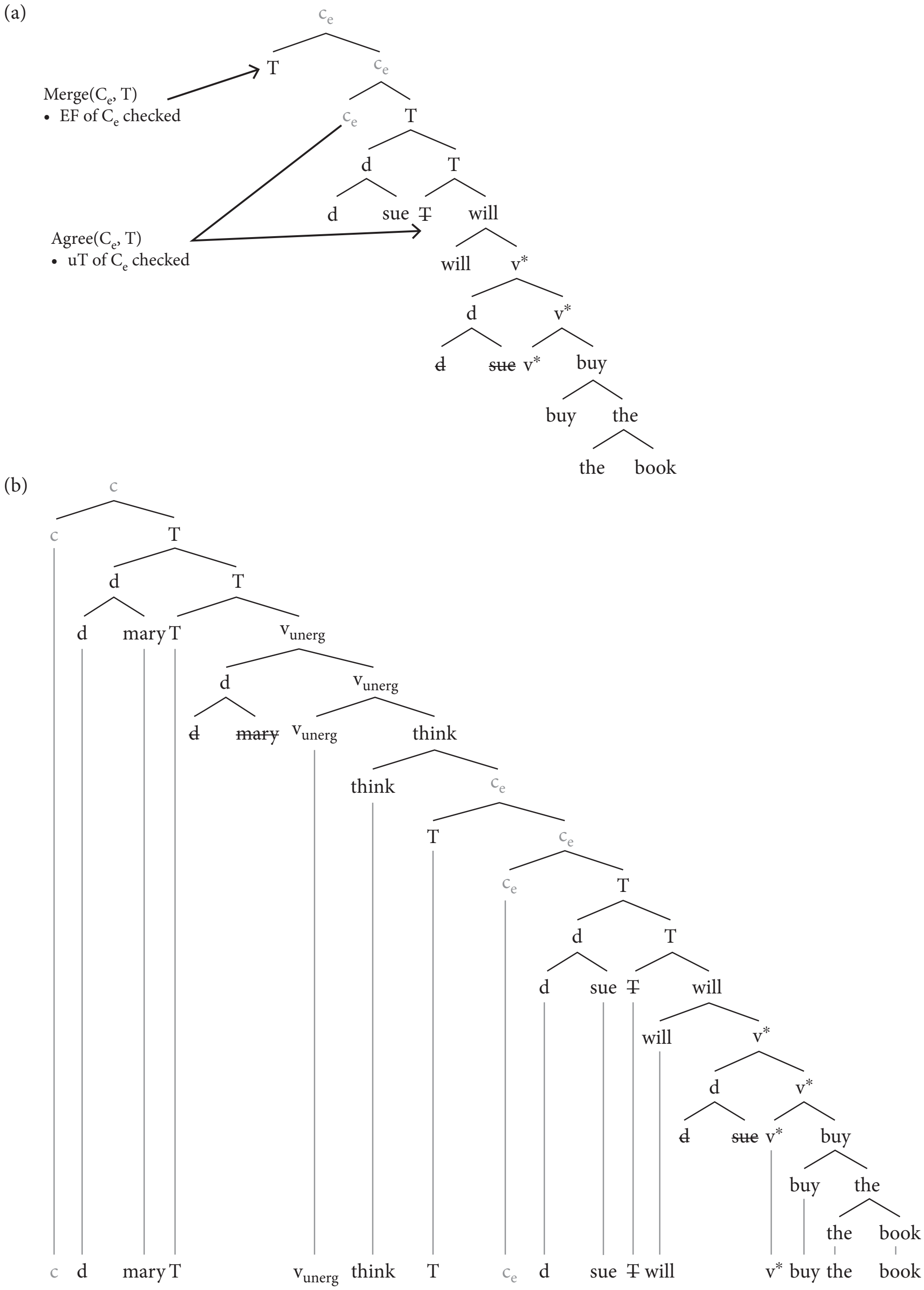
[6] Another possibility is that if v* doesn't assign Case to an argument, then its uPhi don't have to be checked. Also see Epstein et al. (2016) for an account that relies on pair-Merge.

[7] For P&T, an EPP subfeature of uT forces raising.



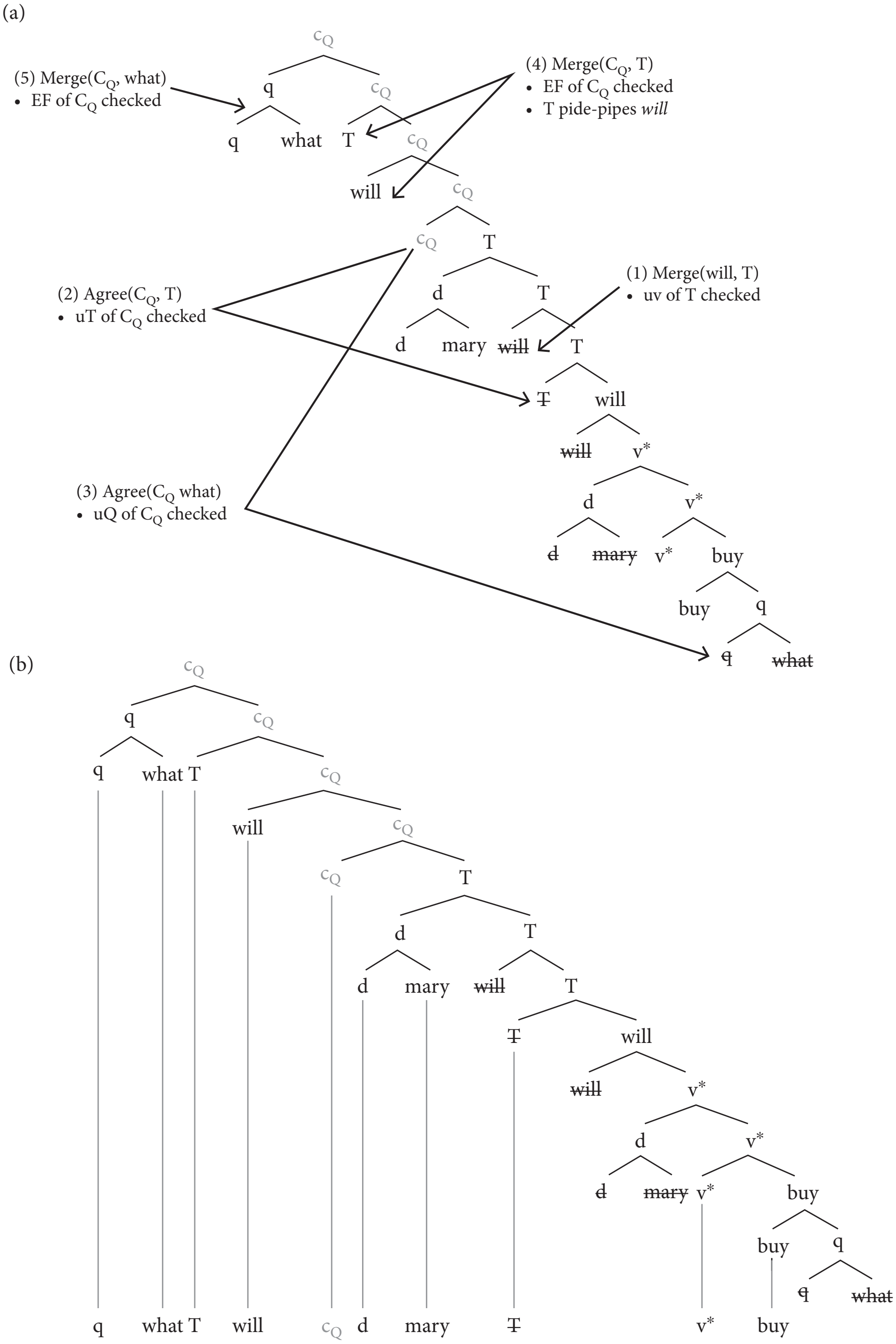
Spell-out:
mary -s think sue -s will buy the book (after morpheme realization)
mary think -s sue will -s buy the book (after affix-hop)
mary thinks sue will buy the book

Figure 3.13 Derivation of embedded clause of (17) without *that*



Spell-out:
mary -s think that sue -s will buy the book (after morpheme realization)
mary think -s that sue will -s buy the book (after affix-hop)
mary thinks that sue will buy the book

Figure 3.14 Derivation of embedded clause of (17) with *that*



Spell-out:
what -s will mary buy (after morpheme realization)
what will -s mary buy (after affix-hop)
what will -s mary buy (after morpheme realization, stage 2)
what will mary buy

Figure 3.15 Derivation of (18)

P&T (2001) present an economy-based account for the subject/object *wh*-question asymmetry, exemplified in (20a–c). When the object *what* undergoes *wh*-movement, *did* is pronounced (generally considered to be an indication of T to C movement), but when the subject *who* undergoes *wh*-movement, *did* is not permitted, as in (20b,c).

(20) a. What did Mary buy?
b. Who bought the book?
c. *Who did buy the book? (unless *did* is focused) (Koopman 1983, per P&T: 357)

The relevant portions of the derivation of (20)a are shown in Figure 3.16. As with the previous example, the highest verbal element, in this case v*, remerges with Tpast (past tense). An Agree relation between C_Q and Tpast checks the uT feature of C_Q and an EF forces Tpast to remerge. When Tpast remerges, v* is also pied-piped. Then C_Q forms a second Agree relation with the Q-phrase *what*, resulting in checking of the uQ feature of C_Q, and an EF forces *what* to remerge. In this case, Tpast in C_Q is pronounced as *did*, assuming that when T in C_Q lacks a phonological form, a form of *do* is required. As in the previous example, separate Agree operations are required to check the uQ and uT features of C_Q.

The subject *wh*-question in (20)b has the structure shown in Figure 3.17. In this example, the subject *who* is a Q-phrase, and thus it can check a uQ feature. Furthermore, the subject has Nominative Case, which it obtains from T. When ~~Tpast~~ Agrees with *who*, both the uT and uQ features are checked. Then an EF of C_Q forces remerge of *who*. The other option would be for the uT feature of C_Q to be checked by Tpast, and then for the uQ feature to be checked via a separate Agree relation with *who*. But P&T argue that this latter derivation is ruled out by economy. Since the uQ and uT features of C_Q can be checked via a single Agree relation formed with *who*, the less economical derivation involving two separate Agree relations is blocked. As a result, T cannot move to C, and (20b) is ruled out.[8]

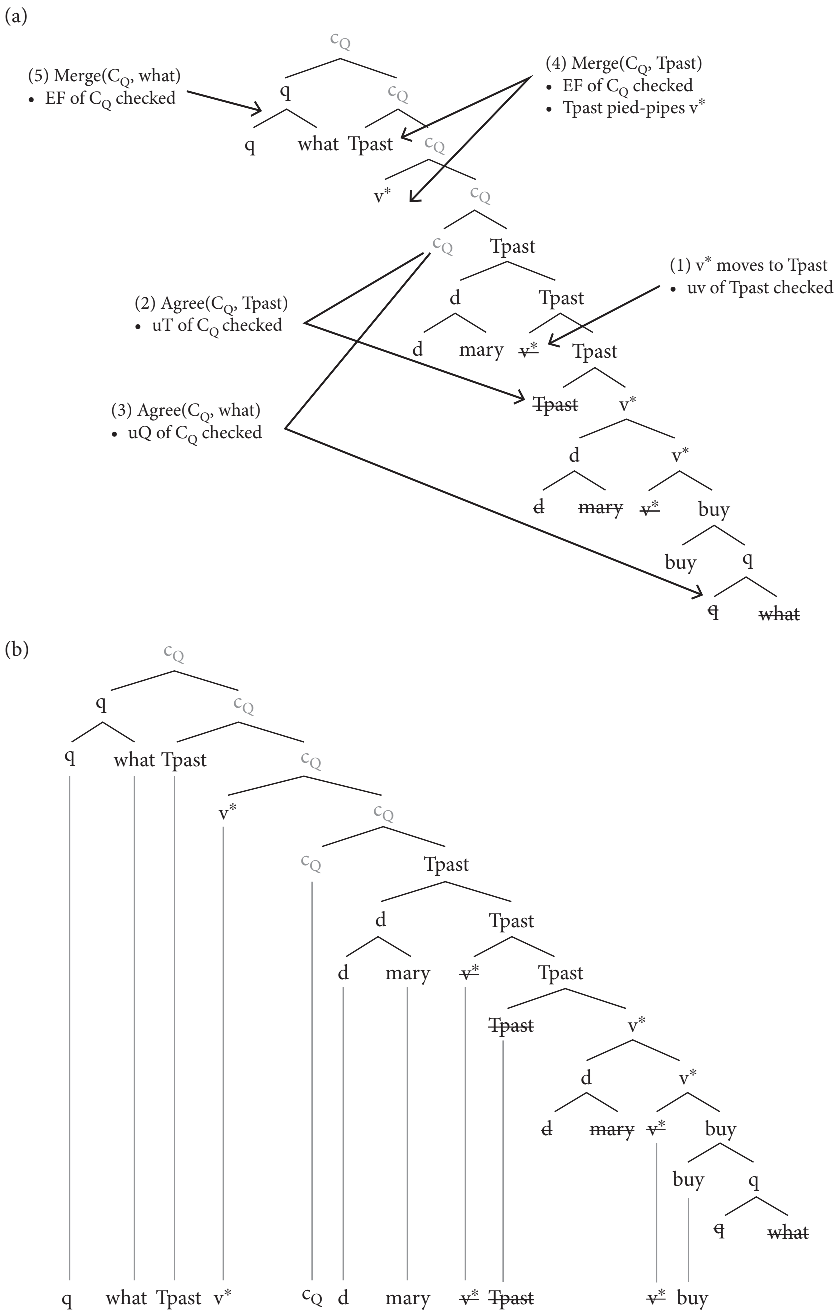
We next explain how we implemented P&T's account of the *that*-trace effect, as found in (21a,b).

(21) a. Who did John say will buy the book?
b. *Who did John say that will buy the book? (P&T: 371)

The embedded clause contains what we refer to as C_{eQ}, which is an embedded non-interrogative C that hosts a *wh*-phrase. C_{eQ} contains uT and uQ features, but it cannot fully license a Q-phrase. Rather, it hosts a Q-phrase that undergoes movement from an embedded clause to a higher clause. The embedded clause of (21a) is shown in Figure 3.18a. Assume that a Q-phrase has an unvalued scope feature, uScope, which is

[8] Note that according to P&T's analysis, a subject can also check uT on C_Q, as in (i) in which the subject *Mary* checks the uT, but this is ruled at Spell-Out for reasons of interpretation—only an exclamative subject (not a *wh*-phrase) can appear in a specifier of a C_Q resulting in an exclamative. Thus, P&T rely on what is essentially a conceptual interface well-formedness condition to block (i).

(i) *What Mary bought (P&T: 358)



Spell-out:
what -ed(sg) do mary buy (after morpheme realization)
what do -ed(sg) mary buy (after affix-hop)
what did mary buy

Figure 3.16 Derivation of (20a)

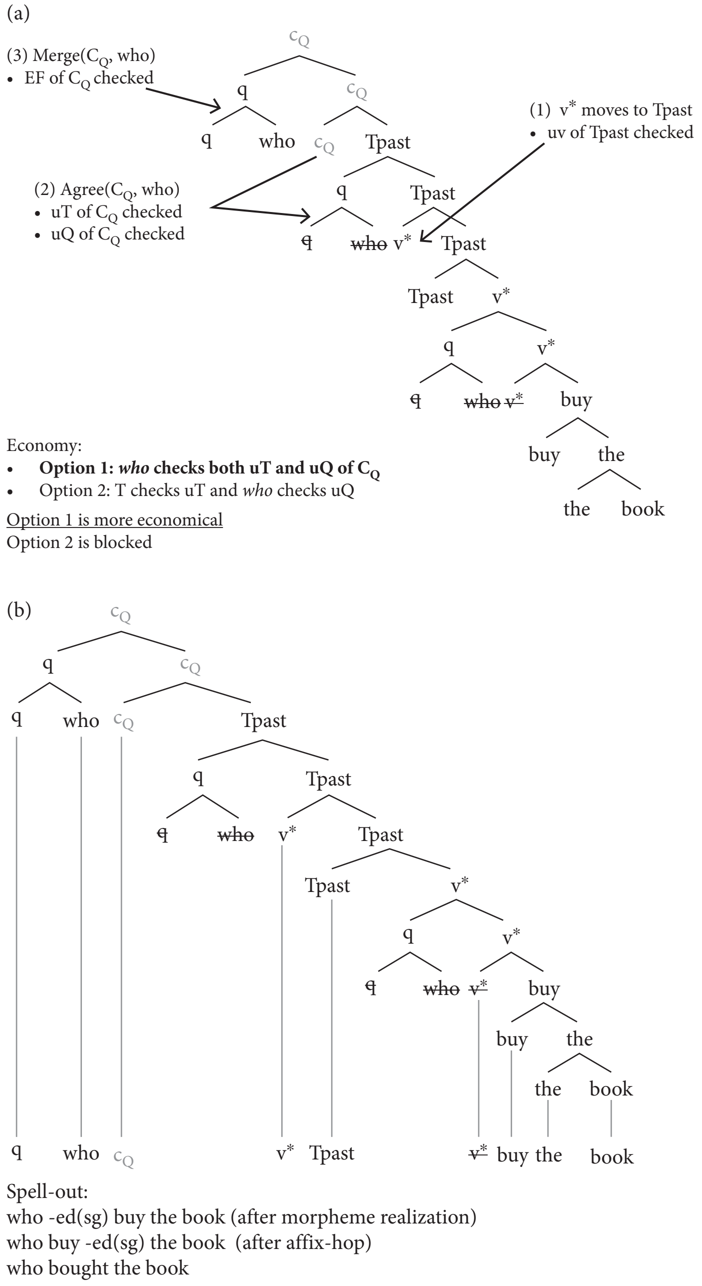
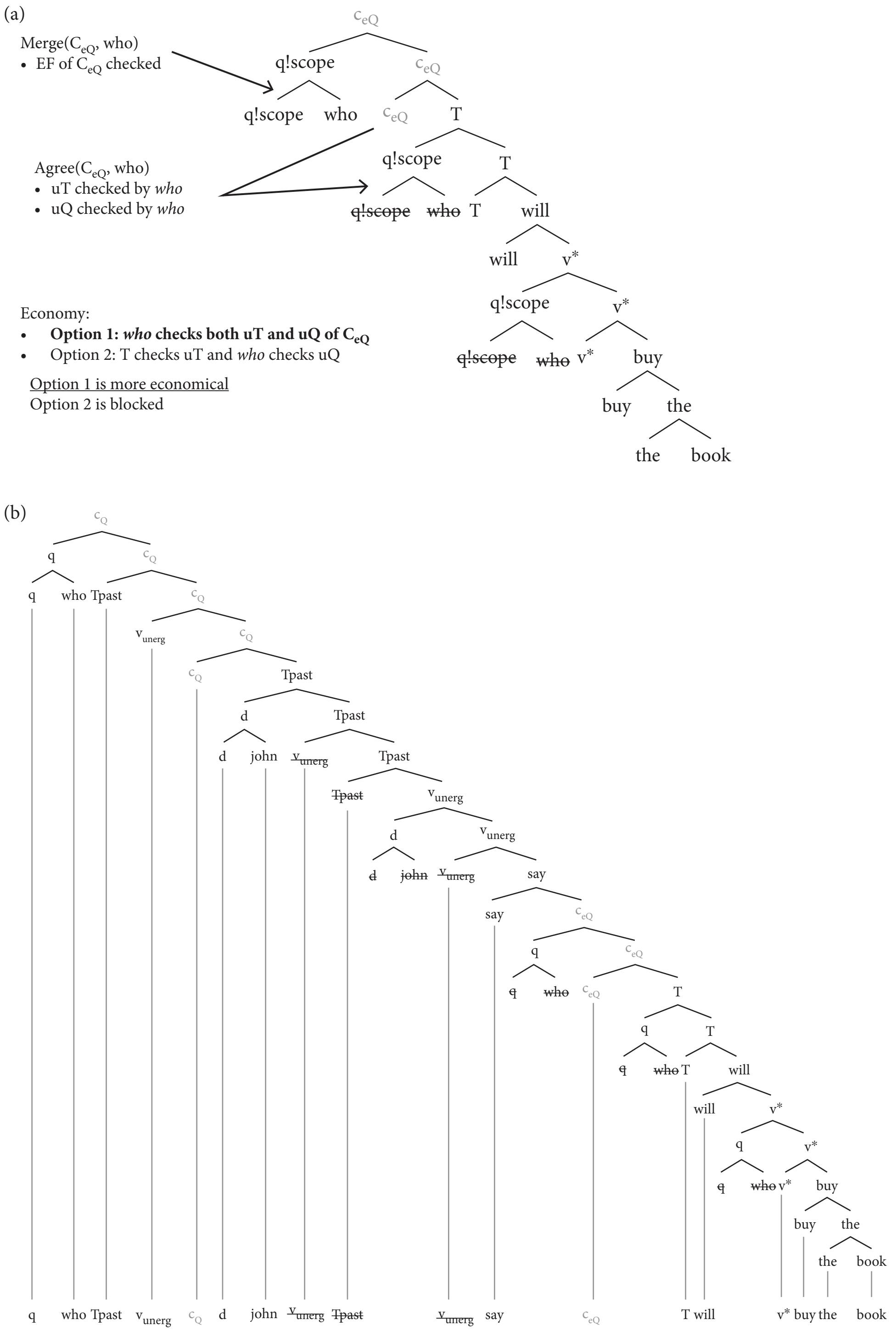


Figure 3.17 Derivation of (20b)

indicated as !scope in Figure 3.18a. This uScope is not checked via Agree with C_{eQ}—uScope must be checked by a truly interrogative C_Q. As in the previous example, C_{eQ} forms an Agree relation with *who*, thereby checking the uT and uQ features via a single Agree relation, since *who* can check a uQ feature and the Nominative Case of *who* can check the uT feature. The other option of checking the uT feature via Agree with T,



Spell-out:
who -ed(sg) do john say -s will buy the book (after morpheme realization)
who do -ed(sg) john say will -s buy the book (after affix-hop)
who did john say will buy the book

Figure 3.18 Derivation of (21a)

which would result in pronunciation of *that*, and the uQ feature via Agree with *who*, is less economical, and thus banned. Therefore, the ill-formedness of (21b) is blocked due to economy. The final output and Spell-Out are shown in Figure 3.18b. Note that in the matrix clause, the uT feature of C_Q is checked via Agree with T, assuming that the Nominative Case of *who*, from the embedded clause, is not available to check uT in the matrix clause.

In a construction such as (22a), there is no *that*-trace effect because economy is not an option for checking the uT and uQ features of the embedded C_{eQ}, since there is no single element that can check both uT and uQ features.

(22) a. What did John say that Mary will buy?
b. What did John say Mary will buy? (P&T: 370)

In (22a), as shown in Figure 3.19a, the embedded C_{eQ} forms an Agree relation with T, which checks the uT feature. As a result, an EF of C_{eQ} forces T to remerge, and T in C is pronounced as *that*. In (22b), as shown in Figure 3.19b, C_{eQ} forms an Agree relation with the subject *Mary* instead of T. Thus, *Mary* remerges with C_{eQ} to check an EF. Since there is no T in C, there is no pronunciation of *that*. In both examples (22a,b), C_{eQ} also forms an Agree relation with the object *what*, which checks the uQ feature of C_{eQ} and forces *what* to remerge.

We have thus demonstrated how we implemented the core proposals of P&T. We changed a few of the formalisms of P&T (e.g. we used an EF instead of EPP subfeatures, and uQ instead of uWh). We also had to implement raising of a verbal element to T in matrix interrogative clauses, which, although not discussed by P&T, is crucial for producing the correct outputs. Our model thus demonstrates that P&T's proposals can be successfully implemented; and this implementation is not in conflict with the constructions, involving multiple agreement and expletives, presented in the preceding sections. We next explain how we extended this model to account for relative clauses.

3.6 Relative Clauses (Pesetsky and Torrego 2001; Gallego 2006)

Gallego (2006) proposes an analysis of relative clauses that follows P&T's (2001) views about economy and feature checking. Therefore, it is natural for us to extend our analysis to utilize Gallego's insights. Gallego proposes that a relative C has uRel and uT features. Following P&T, the uT feature can be checked by T or by a nominative subject. The uRel feature, however, must be checked by a relative DP. Gallego also assumes that there is a c head above C that has uPhi (cf. Bianchi 2000 for a similar proposal), and N raises to c to check uPhi. Furthermore, these uRel, uT, and uPhi features can have EPP subfeatures, although there is some variation with respect to when an EPP subfeature is present. We explain how Gallego accounts for subject relative clauses, discuss some problems, and then explain our implementation.

English subject-relative clauses permit *who* but not *who that*, as show in (23a,b).

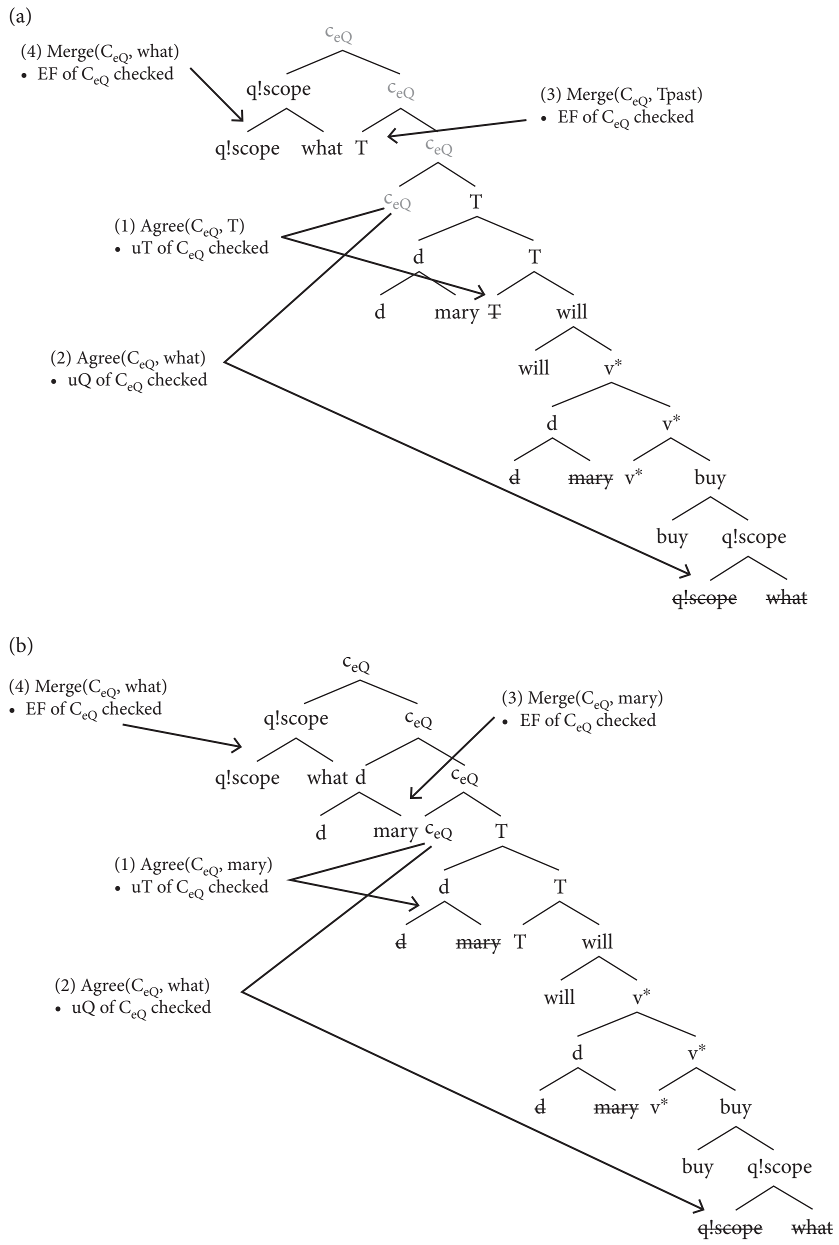


Figure 3.19 Derivation of (22a)

(23) a. the man who loves Mary
b. *the man who <u>that</u> loves Mary (Gallego 2006: 156)

The relevant portion of the derivation of (23a) is shown in Figure 3.20. The relative C has uT and uRel features and each of these features has an EPP subfeature, thus meaning that anything that Agrees with uT or uRel must also remerge with C. When C Agrees with *who man*, the uT feature is checked by the Nominative Case of *who man*

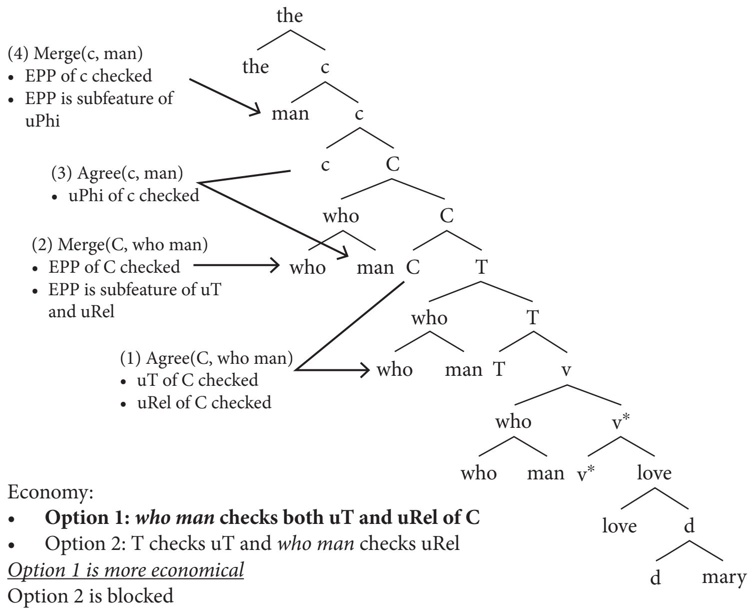


Figure 3.20 Derivation of (23a)

and the uRel feature is also checked, assuming that *who man* has an iRel feature. The EPP (which actually refers to a subfeature of uT and another subfeature of uRel) is then checked via remerge of *who man*. Then there is a higher c head, which has uPhi. The c head Agrees with the phi-feature bearing *man*. The Phi of *man* checks the uPhi of c, and an EPP subfeature of uPhi forces *man* to remerge with c. Then the determiner *the* is externally Merged. Note that this derivation involves a single Agree relation between the relative C and *who man*. The other option would be for the uT feature of C to be checked by T and then the uRel feature to be checked by *who man*, which would involve two Agree relations, and would also result in pronunciation of *that*, due to T appearing in C. But economy rules out this option, given that a single Agree relation is possible. Thus, economy accounts for the impossibility of *that* in subject relatives with a relative D *who*, thus blocking (23b).

Gallego also accounts for the fact that *that* is required and a zero relative is banned, in a subject relative clause such as (24a,b).

(24) a. the boy that called Mary (Gallego 2006: 158)
b. *the boy called Mary

In this case, Gallego assumes that a DP with a null D cannot move, following Chomsky's (2001) view that an empty category cannot be pied-piped. In the derivation of (24a), as shown in Figure 3.21, the relative DP does not move. As a result, the relative DP does not remerge with C. Note also that because the relative DP has a null D, it cannot even move to Spec, T. The uT of C forms an Agree relation with T and an EPP subfeature of uT forces T to remerge. Note that T in C results in pronunciation of *that*. The uRel of C forms an Agree relation with the relative DP. But the uRel, conveniently, lacks

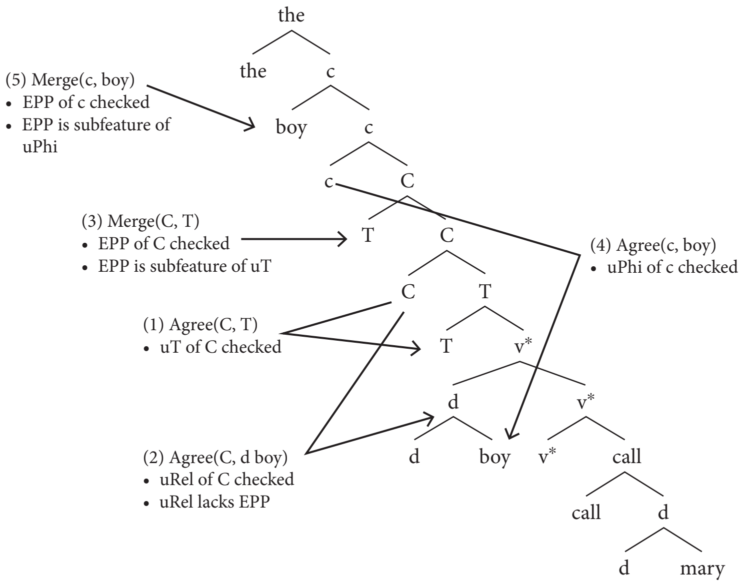


Figure 3.21 Derivation of (24b)

an EPP subfeature, so the relative DP does not move. Then, the uPhi feature of c Agrees with *boy* and an EPP subfeature of uPhi forces *boy* to remerge with c. Crucially, because the relative DP cannot move, the uT of C must be checked by T, which forces pronunciation of *that*.

Gallego's analysis is not without problems. First of all, it isn't clear why a DP that has a null D cannot move, but a DP with an overt D can move. Also, it isn't clear why a uRel feature of C has an EPP subfeature when there is an overt D, as in (23a), but lacks an EPP subfeature when there is a null D, as in (24a). The uRel lacks an EPP subfeature only when convenient—i.e. in cases in which the relative DP cannot move. Another consequence of this analysis is that a subject with a null D cannot move to Spec of T, but subjects are generally considered to be in Spec of T. Also, Gallego's analysis requires a c head, as shown in Figures 3.20 and 3.21, but the question arises of why c Agrees with the relative noun to the exclusion of the relative D. Also, it isn't entirely clear why c is necessary.

We implemented a modified and improved version of Gallego's analysis. We adopt Gallego's proposal that a relative C has uT and uRel features. But, crucially, we do away with the c head, and we also do away with the stipulation that a null d can't move. There is also no Agree relation that targets the relative N to the exclusion of the D head. Our core assumptions about relative clauses are as follows.

(25) Relative C (C_{rel}): [uRel,uT,EF]

(26) N: [uD]

(27) Relative Ds:
a. d_{rel}: [Rel]
b. who_{rel}: [Rel,T]

We assume that a relative C, our C_{rel}, has uRel and uT features, as well as an EF, as shown in (25). Our EF serves the purpose of remerging anything that an Agree relation is established with. So if a relative DP Agrees with the uRel of C_{rel}, then the EF of C_{rel} forces remerge of the relative DP. We propose that N, shown in (26), has a uD feature that requires checking via a D element. Therefore, if D and N Merge, the uD feature on N is typically checked by ~~an~~ D feature of D. A relative D, shown in (27), however, is defective in that it is unable to check a uD feature, as it lacks a D feature. This has the important consequence of leaving the N complement of a relative D unlicensed. We also assume that the null relative D, indicated as d_{rel}, (27a), has a Rel feature that can check a uRel, but it lacks what we refer to as a T feature. This T is akin to nominative Case that can check a uT. As a result of lacking a T feature, d_{rel} is unable to check a uT feature. Note that we assume that even if a relative DP headed by d_{rel} has Nominative Case, it cannot check a uT feature, due to a lack of a T feature. The other relative head who_{rel}, (27b), has both Rel and T features. Thus, it can check both uRel and uT. We also assume that there is an operation of Last Resort, (28), whereby an unlicensed argument remerges with the root of an SO. This serves the purpose of bringing an unlicensed argument into the appropriate position for forming a relative clause.

(28) *Last Resort*
At a phase boundary, remerge an unlicensed element with the root.

The derivation of (23a), as produced by our model, is shown in Figure 3.22. The relative DP has the form *who*$_{rel}$ *man*, with the relative head *who*$_{rel}$, which has both Rel and T features. Thus, when C_{rel} Agrees with the relative DP, both the uRel and uT features of C_{rel} are checked. An EF on C_{rel} forces the relative DP to remerge. Note that the EF feature is not a subfeature of uRel or uT, *contra* Gallego. Rather, any element that Agrees with C_{rel} is forced to remerge due to C_{rel}'s EF. The other option would be to check the uRel feature by the relative DP and the uT feature by T, but this would involve two Agree relations, which is blocked by economy (16). This analysis accounts for why *that* is not possible, as in (23b). The uT feature cannot be checked by T, so *that*, which is T in C, is not permitted. When C_{rel} is complete, the relative N *man* is still unlicensed, assuming that it has a uD feature. At this point, the operation of Last Resort (28) applies and *man* remerges with the root node. Since *man* is a head, and the C_{rel} projection is an XP, the head *man* labels.[9] Then, when the external D *the* is Merged, the uD feature on *man* is checked and *the* labels. The D *the* has an unvalued uCase feature (indicated as !case), which is eventually checked when the DP obtains Case, for example as a subject or object of a larger syntactic structure.

The derivation of (24a) is shown in Figure 3.23. In this example, the relative DP is headed by d_{rel}, which crucially lacks a T feature, so it cannot check uT on C_{rel}. Therefore, there is no option of the relative DP checking both the uT and uRel features

[9] See Cecchetto and Donati (2015) for a similar head-raising account of relativization, In Cecchetto and Donati's account, only heads (not phrases) may raise to relabel a clause. Furthermore, head raising is constrained by Gross Minimality considerations. We leave comparison of our account with that of Ceccehetto and Donati for future work.

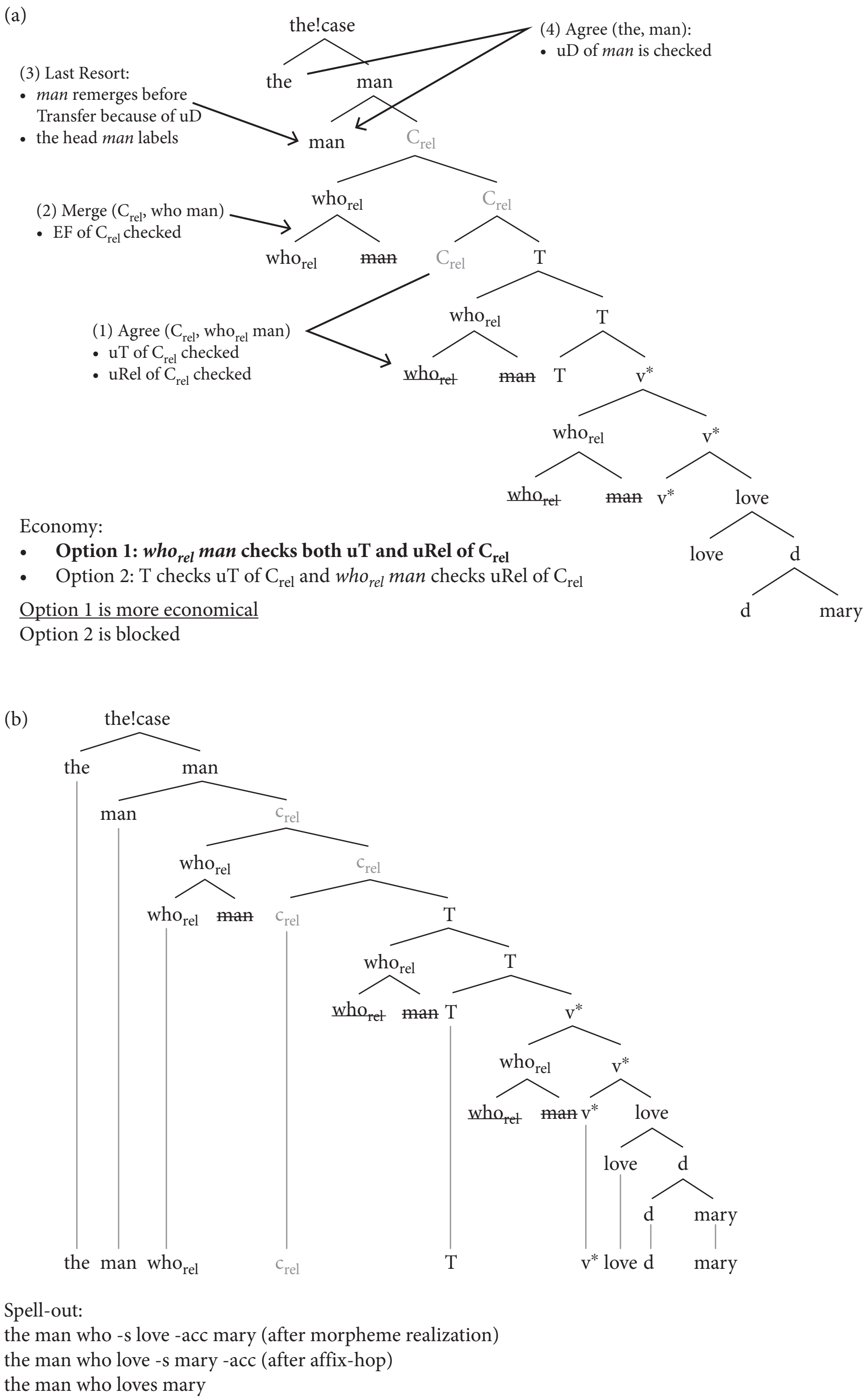
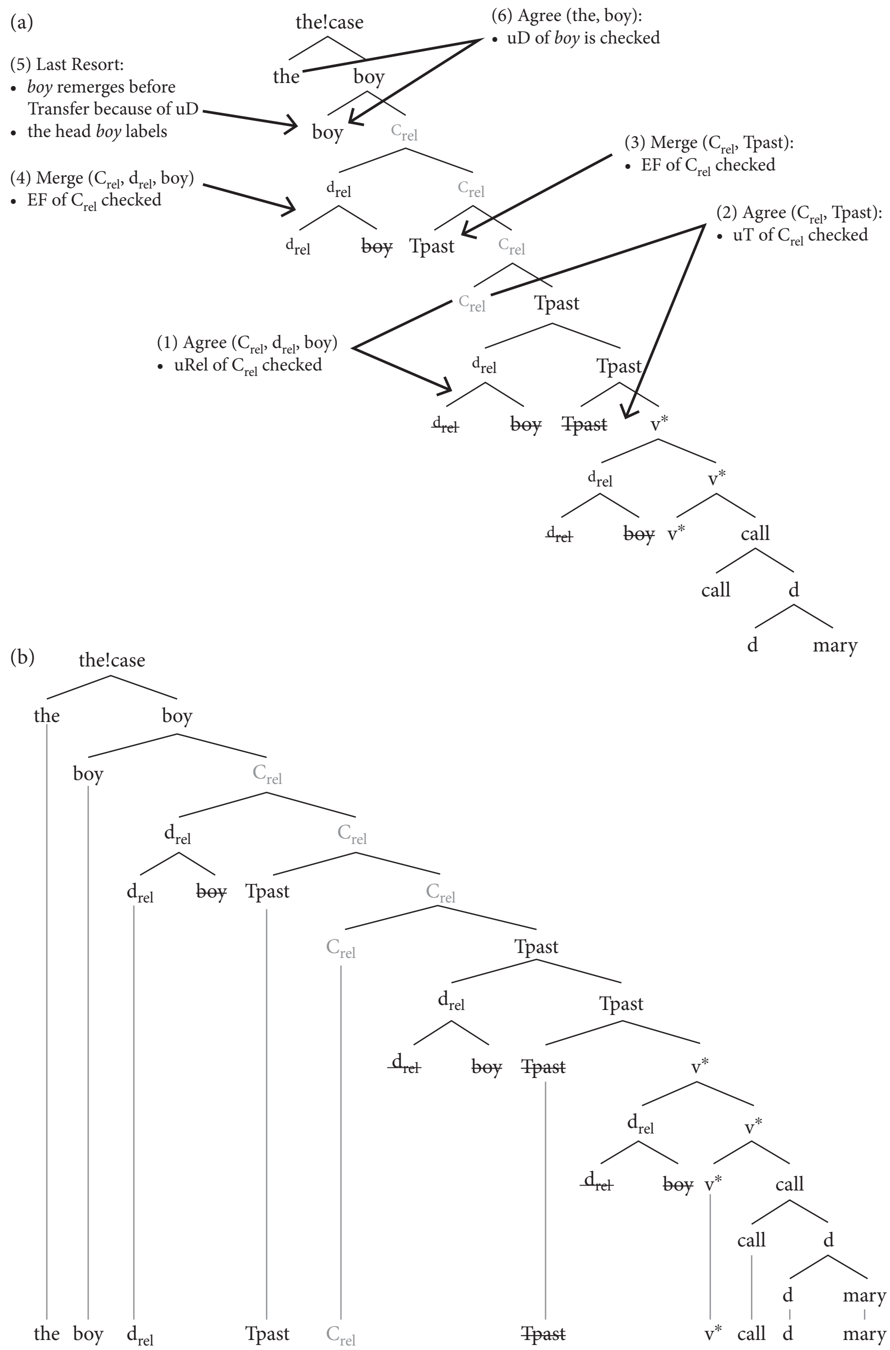


Figure 3.22 Derivation of (23a) (implementation)



Spell-out:
the boy that -ed(sg) call -acc mary (after morpheme realization)
the boy that call -ed(sg) mary -acc (after affix-hop)
the boy that called mary

Figure 3.23 Derivation of (24a) (implementation) of C_{rel}. As a result, C_{rel} forms an Agree relation with d_{rel} *boy*, which checks the uRel feature, and C_{rel} must form a separate Agree relation with T to check the uT feature. An EF feature forces both T and d_{rel} *boy* to remerge. Remerge of T in C results in pronunciation of *that*. Then when the C_{rel} projection is complete, Last Resort applies to force the unlicensed relative noun *boy* to remerge, due to its uD feature. Being a head, *boy* labels. Then when the external D *the* is Merged, the uD feature of *boy* is checked and the relative clause is successfully completed as shown in Figure 3.23b.

Following the insights of P&T and Gallego, we have implemented relative clauses without an extra projection in the left periphery of the clause, and without recourse to optional EPP subfeatures. Rather, we simply use the idea that a relative C head, C_{rel}, has uT and uRel features, and a relative D differs in whether or not it can check the uT feature of C_{rel}. When the relative D can check uT, then economy requires this operation, thus blocking pronunciation of *that*. When the relative D cannot check uT, then T must check uT, thus requiring the presence of *that*.

3.7 Conclusions

We have highlighted important details of our implementation of expletives and multiple agreement (Chomsky 2001), thematization/extraction (Sobin 2014), *that*-trace effects and subject vs. object *wh*-movement (P&T 2001), and relative clauses (P&T 2001; Gallego 2006). We explained how we dealt with particular problems that arose in the theories that we implemented and how these problems can be resolved. Problems arising with respect to multiple agreement in Chomsky (2001) were dealt with via feature unification. Problems arising with respect to feature checking and the position of the expletive *there* were also resolved with feature unification, as well as with the notion that an expletive is Merged low in a position below T. Problems from Sobin (2014) regarding locality of affixes and Spell-Out were resolved via the use of EFs (prt has an EF) and distinguishing v∼ into two separate types that differ in pronunciation (v∼unacc and v∼). We implemented P&T's account of *wh*-movement and *that*-trace effects, and in the process clarified how the fine details work; we utilized an EF instead of an EPP subfeature and we implemented movement of the highest verbal element to T in a matrix clause. Lastly, we resolved problems with Gallego's (2006) account of relative clauses. We utilized relative Ds that differ with respect to whether or not they can check a uT feature. We also incorporated a notion of ~~last resort~~ whereby an unlicensed relative N can undergo movement, thus doing away with the notion that an extra projection is needed in the left periphery. Crucially, all of these modifications and improvements of work from Chomsky (2001), P&T (2001), and Gallego (2006) were implemented via a single unified model, thus demonstrating that a wide variety of constructions can be accounted for with a single theory.

In conclusion, we have integrated core components of different linguistic theories to create a single unified theory that accounts for a wide variety of data. Our theory provides possible explanations for target phenomena, and this theory has been successfully tested.