

Chapter 1

Setting the Stage

1.1 Phrase Structure Composition in Grammatical Derivation

Within the generative paradigm, our ability to exploit the system of connections between form and meaning afforded by human language is taken to derive from the existence of a mental grammar. A grammar, as understood here, is a specification of the possible connections between form and meaning. I assume that these form-meaning connections are encoded as some sort of formal object, called a *linguistic expression*. Since a given language provides an infinity of connections between form and meaning, a grammar must specify an infinity of possible linguistic expressions. Consequently, grammars are typically specified in some sort of recursive fashion.

This abstract conception of grammar permits a wide variety of instantiations. To at least a first approximation, these can be classified into two categories. *Representationally oriented grammars* determine the set of linguistic expressions using a system of well-formedness constraints. Each of these constraints provides an evaluation of some part of the linguistic expression. The ultimate well-formedness of the entire linguistic expression is determined by combining the evaluations of the individual constraints. Much as logical axiomatizations do not specify how to go about determining which statements are theorems, representationally oriented grammars do not specify how one should find the well-formed linguistic expressions, but only what properties such well-formed expressions must have. *Derivationally oriented grammars*, in contrast, focus exclusively on the process by which well-formed linguistic expressions are found, by providing a procedure for constructing them. In other words, a linguistic expression E is well formed under a derivationally oriented grammar D only if D can construct E .¹

A derivationally oriented grammar generally includes a set of structural atoms, which I call the *basis* of the derivation. This basis is fed into the derivational procedure, which constructs syntactic structures using operations of two types. The first type, which I call *structural composition*, allows either previously constructed syntactic representations or elements of the basis to be combined to form larger representations. In a derivationally oriented grammar with a finite basis, such operations play a fundamental role, in that they provide a way to generate the requisite infinity of possible structures. Operations from the second class, which I call *transformations*, modify an individual syntactic representation in some specified fashion. Such operations have been implicated in the establishment of morphological and syntactic dependencies, as, for example, between a head and its inflectional morphology or a dislocated expression and its locus of interpretation.

The earliest model in generative grammar (Chomsky 1955, 1957) was derivational in character. Here, we can take the basis to consist of a set of lexical items.² Through context-free rewriting, which we can view as a form of structure composition (but see McCawley 1968), the derivation produces simple syntactic tree structures, which I call *kernel structures*.³ In this theory, kernel structures underlie simple monoclausal active declarative sentences. To derive other sentence types, kernel structures can be modified using a number of transformational operations, called *singular transformations*, such as Subject-Aux Inversion, Question Formation, and Passive.⁴ Such derived kernel structures can also be combined using operations called *generalized transformations* to produce, for example, structures involving relative and complement clauses and coordination. The essential details of the resulting derivational system are depicted in figure 1.1. In the figure, square boxes signify representations, while rounded boxes correspond to application of derivational operations. Among the latter, transformational operations are colored white, while structural composition operations are gray.

Consider how an example like (1) would be generated in this model.

(1) The book is believed to have been written by the Etruscans.

The derivation begins with two independent sequences of context-free rewriting to produce the kernel structures in (2) that underlie the simple sentences *The Etruscans have written the book* and *Mary believes it*.

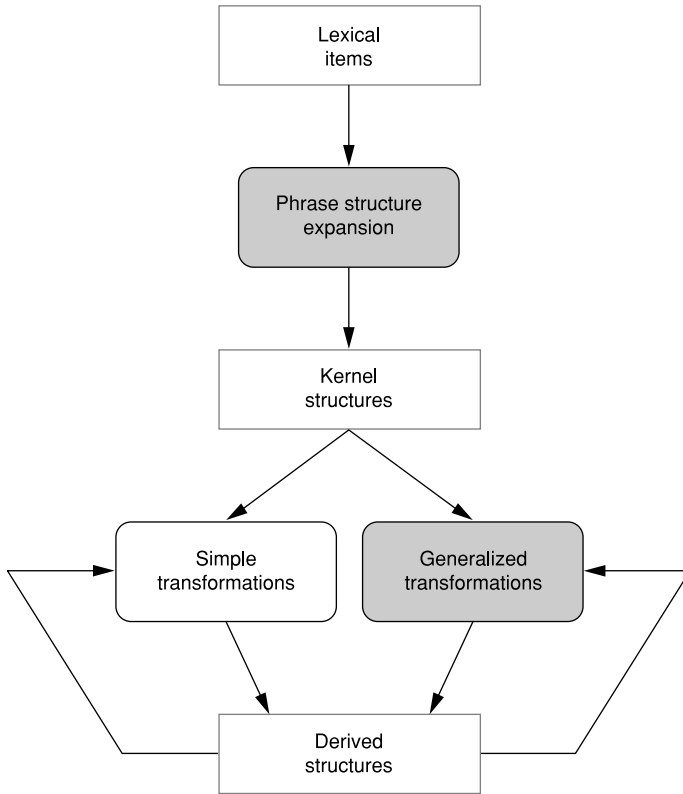
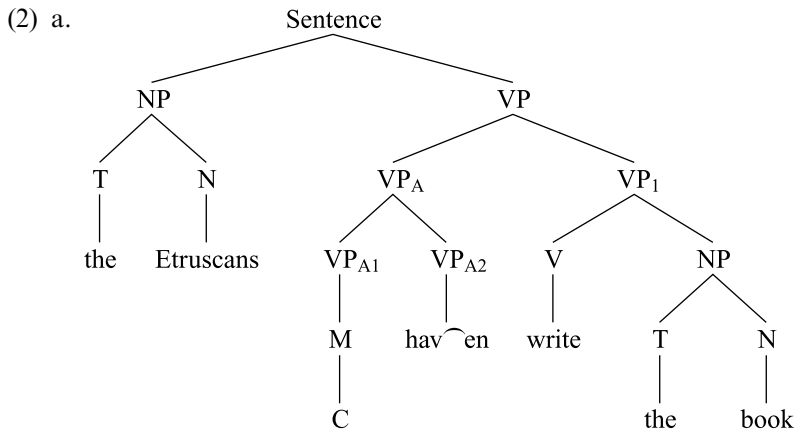
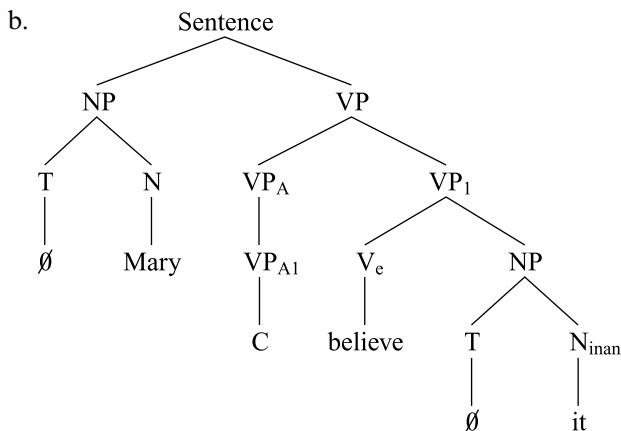


Figure 1.1
Derivational system of Chomsky 1955, 1957





The Passive transformation is then applied to the first of these kernel structures, yielding the phrase marker for the sentence *The book has been written by the Etruscans*. This result is next inserted into the kernel structure in (2b), replacing *it*, using one of a family of generalized transformations that are responsible for embedding clausal arguments. This insertion produces the structure for *Mary believes the book to have been written by the Etruscans*. The Passive transformation applies again, to this structure, producing *The book is believed by Mary to have been written by the Etruscans*. Finally, a transformation of Agent Deletion is applied to yield the sentence in (1).

Observe that in this system, there are two distinguished processes of structural composition, one responsible for building kernel structures, the other for combining them. Note as well that the process of building kernel structures is completely segregated from the application of singular transformations. This means that those singular transformations that are ordered earlier than generalized transformations are guaranteed to apply to phrase markers as large as, but no larger than, kernel structures. Thus, the domain of application of these transformations will naturally be restricted to the structural domain of a kernel structure. Since it is the class of singular transformations that is responsible for the formation of syntactic dependencies, this means that the kernel structure is the locus in which the dependencies formed by these “early” singular transformations must obtain (Chomsky 1955, 534). Indeed, Chomsky uses precisely this localization of application for certain transformations to argue in favor of the conception of kernel sentences that he adopts. In Chomsky’s transformational analysis of English, Passive and Reflexi-

vization are among these early singulary transformations, and he uses this ordering to ensure that these dependencies are properly locally constructed. In our example derivation of (1), for example, Passive applies crucially to the embedded kernel structure prior to its insertion via generalized transformation into the matrix. If Passive did not apply locally in this way, the further application of Passive in the main clause would be impossible, as the structure would not meet the structural description necessary for application of Passive.⁵ Chomsky (1955, 531–532) makes similar arguments concerning the application of the Subject Inversion and Reflexivization transformations.

Subsequent theoretical developments moved away from the use of generalized transformations as devices for structural composition. This was driven by the observation that generalized transformations did not seem to behave in the same fashion or exploit the same kind of expressive power as singulary transformations, in spite of their comparable derivational role. Fillmore (1963) notes, for example, that in spite of the arbitrary interleaving that Chomsky's (1955) model allows between applications of singulary and generalized transformations, no cases appear to necessitate ordering a singulary transformation that applies to a matrix sentence prior to a generalized transformation that embeds a complement within that sentence. Additionally, generalized transformations never need to be extrinsically ordered with respect to one other. These stand in sharp contrast to the extrinsic orderings that at the time were considered necessary among singulary transformations.⁶ Additionally, in comparison to the complex use that singulary transformations make of elementary permutation and recombination operations, the generalized transformations that proved grammatically necessary combined structures in the simplest of fashions. The final nail in the coffin of generalized transformations stems from Chomsky's (1965) observation that the function of generalized transformations, that of building arbitrarily large pieces of phrase structure, can be taken over by the base component once recursive phrase structure rules are permitted. Chomsky's (1955) prohibition on recursive phrase structure rules was at least formally odd, and thus a theory that avoids this stipulation gains in simplicity.

In the model proposed in Chomsky 1965, then, the initial stage of the derivation involves creating an unboundedly large syntactic structure by applying a now recursive set of phrase structure rules. The derivation proceeds by applying singulary transformations to this *deep structure* representation. This derivational model is depicted in figure 1.2. Here,

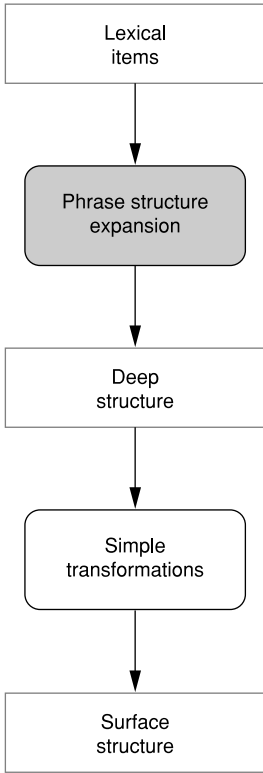


Figure 1.2
Derivational system of Chomsky 1965

structural composition and the transformational operations are completely separate. Since all applications of structural composition precede all applications of transformations, there is no longer any way to derive predictions about the domains over which transformationally derived dependencies will be formed. That is to say, since the phrase structure rules can apply recursively, there is no longer any distinguished structural subunit analogous to the kernel structure over which (certain) transformations might apply. Instead, the effect of forcing certain singulary transformations to apply locally is achieved through imposing a number of stipulations on the application of transformations. For one, the sequence of transformations is required to apply cyclically: first to the substructure containing the lowest sentence domain, next to the larger substructure containing the next lowest sentence domain, and so on, up

through the matrix clause. Additionally, explicit locality conditions are imposed to prevent a single transformation from forming a dependency over too large a structural domain (see Chomsky 1964, 1973; Ross 1967; and much subsequent work).⁷

The lack of interaction between transformations and structural composition in this model means that the proper formulation of transformations does not depend in any way on the manner in which structural composition takes place. As a result, subsequent developments of Chomsky's model up through Chomsky 1981, 1986, though adopting the basic architecture of the framework of Chomsky 1965 in which structural composition precedes transformations, pay less and less attention to the processes by which structure is built. Indeed, with the elimination of an explicit set of phrase structure rules (Stowell 1981), the initial syntactic structure is no longer even seen as deriving from the application of structural composition operations. Instead, the grammatical architecture that is adopted in Chomsky 1981 is a hybrid of the representational and derivational approaches, with the initial phrase marker, now called D-Structure, specified through a set of well-formedness conditions rather than a derivational algorithm. Even with the introduction of well-formedness constraints, however, the grammar remains derivational in one crucial respect: other levels of representation are constructed from D-Structure through cyclic applications of a single general transformational operation, called *Move α* , which accomplishes the dislocation of syntactic elements.

Simultaneously with these developments, another group of researchers moved in the opposite direction, maintaining or increasing the role of structural composition in grammatical derivations while minimizing or entirely eliminating the role of syntactic transformations. In Lexical-Functional Grammar (LFG) (Bresnan 1982b), the lexical basis of the grammar is enriched to allow lexical representations with richer hierarchical structure. In turn, the structural combination operation of context-free phrase structure expansion is strengthened with the additional mechanism of unification. This operation provides a means for combining feature-value information from a variety of sources and can be used to allow complex passing of information during the construction of a syntactic representation (Shieber 1986). Together, these two changes allow syntactic transformations to be eliminated entirely from the derivational system. The formation of previously transformationally established dependencies is instead taken to derive from the application of structural

combination to complex lexical entries that have been transformed by lexical rules. For example, two distinct lexical entries exist for the active and passive forms of an individual verb, and the distinctive combinatory properties of these lexical entries are coupled with unification-driven linkages so as to yield the appropriate thematic dependencies. For long-distance dependencies, unification is used to pass the features of a displaced element back to its θ -role assigner, thereby establishing the proper semantic dependency and ensuring compatibility of, among other things, case and agreement properties. Generalized Phrase Structure Grammar (GPSG) (Gazdar et al. 1985) also eliminates the role of syntactic transformations in grammatical derivations, but takes a different approach to the elimination of transformations. Here, derivation proceeds by context-free phrase structure expansion. In GPSG, dependencies are established not via modification of lexical entries, but via the application of metarules that modify the context-free base rules themselves. Thus, the possibility of, for example, *wh*-question formation derives not from a transformation, but from additional phrase structure rules generated from question-forming metarules. Combinatory Categorical Grammar (CCG) (Steedman 1996) takes yet a third approach, adopting an enriched, universal set of operations for structure composition. The additional combinatory options provided by these operations yield the possibility of generating nonstandard constituent structures, which, in certain cases, avoid the need for transformationally derived dependencies entirely.⁸ For example, no filler-gap dependency needs to be explicitly established in the formation of a relative clause, as the structural composition operation of function composition allows the “fronted” relative operator to combine directly with the remainder of its clause.

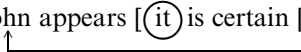
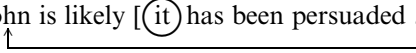
Despite the continued role of phrase structure composition, these non-transformational systems share with the transformational models deriving from Chomsky 1965 the property of having no privileged intermediate level of syntactic structure, like the kernel structure, over which dependencies are formed.⁹ Consequently, some analogue of the cyclicity principle needs to be incorporated to prevent nonlocal dependencies from forming.

Recent work in the transformational paradigm has returned to the idea that phrase structure composition, in the guise of generalized transformations, is a central player in syntactic derivations. There have been a variety of reasons for this shift. One stems from the constraints that have

been assumed to govern the well-formedness of D-Structure representations. Under standard assumptions, the only argument positions that can be filled at this level are those to which θ -roles are directly assigned. The synonymy between examples (3a) and (3b) implies, then, that the subject position of (3b) must not be filled at D-Structure, since (3a) tells us apparently that the subject of (3b), *the Matterhorn*, is in fact assigned its θ -role lower in the structure in the object position of *climb*.

- (3) a. It is tough to climb the Matterhorn.
 b. The Matterhorn is tough to climb.

However, it is also usually assumed that this subject does not raise transformationally from a base position as the verbal object to the matrix subject. Such movement would violate well-attested locality conditions that block movement past one subject position to a higher one, as seen in examples like these:

- (4) a. *John appears [(it) is certain [*t* to win]].

 b. *John is likely [(it) has been persuaded *t* [to come]].


Chomsky (1981) suggests a solution to this puzzling complex of facts in which the object of *climb* in (3b) does not raise to the subject position. Instead, the subject position is filled by a process of lexical insertion that takes place during the application of transformations between D-Structure and other syntactic levels, so that the subject is introduced after D-Structure. Chomsky (1993) notes, however, that under any reasonable understanding of the term *lexical insertion*, this solution is inadequate.¹⁰ Such non-D-Structure subjects can be arbitrarily complex, containing even other instances of such *tough*-movement, as seen in (5).

- (5) That the Matterhorn is tough to climb is easy to see.

To derive such examples under the framework of assumptions just sketched, Chomsky argues that the grammar needs to allow for parallel derivations of the subject clause and the remainder of the sentence, proceeding from distinct D-Structure forms that are integrated prior to S-Structure. In other words, derivations need to make use of a device similar to generalized transformations.

A second reason for the return of generalized transformations to grammatical theory stems from an empirical argument given by Lebeaux

(1988). Lebeaux, along with Freidin (1986), notes that relative clauses that are fronted along with the DP they modify do not exhibit the interpretive effects of having been present in the base position of the *wh*-phrase. That is, whereas in the example in (6a), the subject pronoun may not corefer with *Dave*, a name that it c-commands, such coreference is possible when the relative is fronted as in (6b).

- (6) a. *He_i lived in [the house that Dave_i built] for ten years.
 b. [Which house that Dave_i built] did he_i live in for ten years?

This situation contrasts with that involving clausal complements to nominals. Here, for the purpose of interpretive effects like pronominal coreference the fronted example behaves just like the version in which the phrase is not moved.

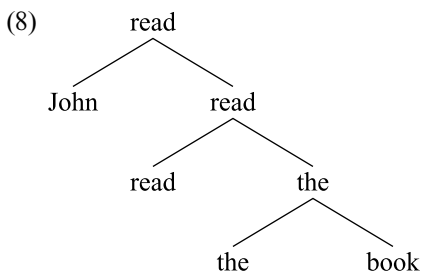
- (7) a. *He_i has been hearing [the claim that Dave_i forged the building permit] for ten years.
 b. *[Which claim that Dave_i forged the building permit] has he_i been hearing for ten years?

Lebeaux explains the contrast between the examples in (6) and (7) by assuming that adjuncts, since they are not assigned θ -roles, are not licensed at D-Structure. Instead, Lebeaux suggests that they are introduced during the transformational derivation with an operation of structural composition he labels *adjunction*. Since adjunction of the relative clause is free to take place after the fronting of the *wh*-phrase in (6b), there is never a point in the derivation at which the structure includes the illicit configuration of the pronoun c-commanding the coreferential name. On the other hand, since clausal complements are assigned θ -roles, they must be present at D-Structure. Hence, the D-Structure representation in the derivation of (7b) will look much like that of (7a), thereby inducing the effect of noncoreference.

Finally, Chomsky (1993) advances a conceptual argument for bringing generalized transformations back into grammatical theory. Chomsky outlines a research program that attempts to formulate what he calls a minimalist theory, in which the only levels of grammatical representation are those that are conceptually necessary. Since the function of grammar is to provide a link between form and meaning, there are minimally two such levels: one that provides an interface with the cognitive systems of articulation/perception, Phonetic Form (PF), and another that interfaces with conceptual/intentional systems, Logical Form (LF). In the

framework of Chomsky 1981, these two levels of representation constitute the outputs of a derivation that begins with the noninterface level of D-Structure. In a minimalist theory, then, there can be no D-Structure starting point of the derivation. Instead, a derivationally oriented minimalist theory that produces PF and LF outputs must include some mechanism of structural composition with which such structures can be built.

In the model proposed in Chomsky 1993 and refined in Chomsky 1995, structures are built out of a set of lexical items, constituting the basis of this system. Chomsky suggests that the grammar includes a single structural composition operation, which he calls *Merge*, that functions by combining two structural elements, either lexical items or previously built structures, into a more complex structure. The usual syntactic tree is now understood to be a representation of a (partially ordered) sequence of applications of *Merge*, much as in Categorical Grammar derivations. The resulting structure's syntactic properties, like its categorial label or agreement features, are inherited from one of the two elements given to *Merge* as input. The tree in (8), then, reflects three applications of *Merge*. First, the lexical item *the* combines with the lexical item *book*, with the properties of *the* being inherited. Next, this result combines with the lexical item *read*, with the properties of *read* being inherited. Finally, this complex combines with *John*, with *read* projecting its properties once again.



Chomsky distinguishes two subcases of *Merge*, *substitution* and *adjunction*, corresponding to the traditional distinction between combination of a syntactic head or phrase with a complement or specifier, and combination of a head or phrase with a modifier or “adjoined” element. The difference between these operations is reflected, essentially, as a diacritic that is added to the label of a structure resulting from adjunction.

In addition to the *Merge* operation, this system includes a single transformational operation, *Move*, just as in Chomsky 1981 and subsequent

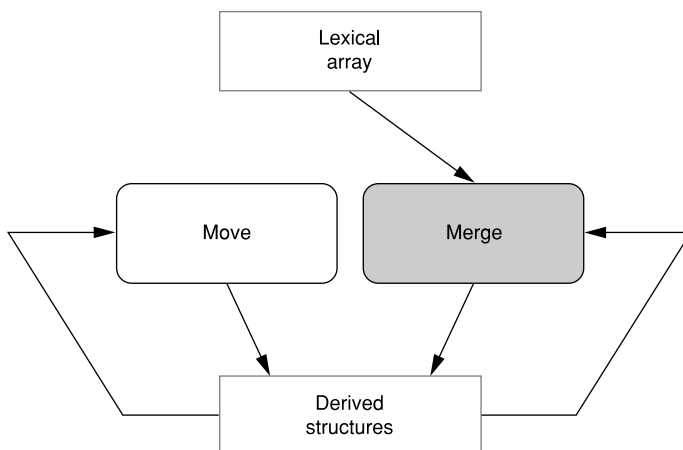


Figure 1.3
Derivational system of Chomsky 1993, 1995

work. With the reintroduction of a structural composition operation, however, the possibility is now open for the application of transformations to interleave with structural composition, as in the model of Chomsky 1955, 1957 (see figure 1.1). This possibility is indeed exploited in the conception of derivation adopted in Chomsky 1993, 1995, which is depicted in figure 1.3. It is important to observe that the objections raised earlier against the presence of generalized transformations in such an interleaved grammar do not apply to this more recent proposal. Those objections related to limited types of interactions occurring between generalized and singular transformations, as compared to the power offered by the possibility of extrinsic ordering. Under the current conception, the generalized and singular transformations, Merge and Move, are of such a general form as to eliminate the utility of extrinsic ordering.

There is another important difference between this model and that originally proposed by Chomsky (1955, 1957). Recall from our discussion of that earlier system that no cases of structural composition accomplished by the rewriting of the (nonrecursive) phrase structure rules could be interrupted by application of transformations. I suggested that this separation had the beneficial effect of providing a privileged structural domain, the kernel structure, over which grammatical dependencies might be localized. In the current model, however, there is no such privileged

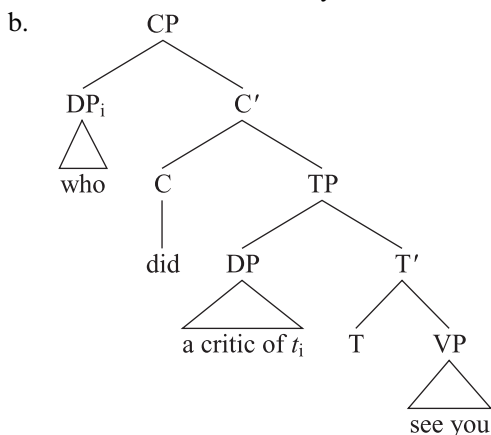
domain. Structure composition is accomplished in a uniform fashion, via Merge. As a result, transformational movement may apply to structures produced at any point in the derivation, large or small. This means that once again locality restrictions on syntactic dependencies will arise in this system only in virtue of explicitly stipulated constraints on the application of movement.

Most recently, Uriagereka (1999) and Chomsky (2000, 2001) propose models in which there is an intermediate structural unit having a privileged derivational status, much like the kernel structures of old. These systems share with Chomsky's (1993, 1995) model the property that applications of structural composition, in the guise of Merge, and applications of transformations, in the guise of Move, may freely intermingle. These models differ, however, in that derivations are not permitted to manipulate structures beyond a certain size, which Uriagereka calls a *derivational cascade* (henceforth, DC) and Chomsky calls a *phase*. The common intuition that Chomsky and Uriagereka pursue is this: once a DC/phase has been constructed, it is sent off for interpretation at the PF and LF interfaces. Consequently, from the point of view of the subsequent derivation, the DC/phase is frozen: subsequent derivational operations may not modify its internal structure, but must treat it as an atomic entity. A completed DC/phase may be merged with other elements, or it may be moved in its entirety; but it may not be altered. In its strongest form, this style of derivation imposes a severe locality condition on the formation of dependencies: namely, they must obtain within a single DC/phase.

Of course, the empirical implications of such a derivational model depend on the size of DCs/phases. Uriagereka proposes that the DCs arise from the need to produce the linearly sequenced representation that is required to interface with the articulatory/perceptual systems. He suggests that syntactic structure is unordered and that the simple procedure by which structure is linearized, which reverses the order of Merge operations, functions only with particularly simple structural configurations, in particular, those that are uniformly right branching. When two complex structures are merged, this simple procedure will not suffice to determine linear ordering. To avoid this problem, Uriagereka suggests that the derivation first "spells out" one of the complex structures, so that it is treated subsequently as a sort of complex word. When this complex word is merged with the other complex structure, the simple linearization

procedure is now able to determine an ordering between these elements, the complex word preceding the complex structure. As Uriagereka notes, this proposal derives so-called left branch effects, in which dependencies like *wh*-movement may not extend out of a complex substructure on the left. That is, an example like (9a) is not generable since the DP subject *a critic of who*, as a left branch, must be spelled out before it is combined with *see you*, leaving it syntactically frozen. As a result, the subsequent extraction of *who* out of this DP to the specifier of CP is impossible.

(9) a. *Who did a critic of see you?



Uriagereka's proposal does not, however, explain a number of other well-known locality restrictions on dependency formation. For example, it captures neither Relativized Minimality effects, like that seen in the *wh*-island in (10a), nor the class of adjunct island effects in which the adjunct is rightwardly attached, as seen with the relative clause in (10b): both of these structures form a single DC with a higher clause, and consequently they should form domains that are transparent to movement.¹¹

(10) a. *Why did you see [what a critic wrote *t* *t*]?

b. *Who did you see a critic [that wrote about *t*]?

Chomsky's conception of the limitation on phase size is different. He suggests that phase boundaries are the reflection of breaks between semantically saturated phrase structures. CP, corresponding to a saturated Davidsonian event structure, constitutes one type of phase boundary, and *v*P, as the phrasal instantiation of a saturated lexical predicate, constitutes the other.¹² Under this proposal, then, a single derivational

phase may contain at most one such semantically saturated head, that is, either a *C* or a *v*. Chomsky argues that phases provide the appropriate context for computing derivational economy of the sort I discuss in chapter 4.

Following the intuition that phases are immutable once they have been completed, one would expect on Chomsky's proposal that movement of a *wh*-element out of a CP should be impossible. Yet, as is well known, this movement may apply across an unbounded number of CP boundaries.

- (11) What_{*i*} do you think [_{CP} that Alice would suggest [_{CP} that Peter ask us [_{CP} PRO to do *t_i*]]]?

To avoid this undesirable result, Chomsky weakens the degree to which phases are frozen. Chomsky (2000) proposes a Phase Impenetrability Condition stating that only elements within the complement domain of a phase's *C* or *v* head are inaccessible to operations outside the phase. This leaves, roughly, a phase's head and its specifiers as accessible to operations outside the phase. For cases of unbounded *wh*-movement, then, it is sufficient for the *wh*-phrase to move successive cyclically through the specifier of each phase's head, a position that plays the traditional role of an escape hatch for movement out of an otherwise closed domain.¹³

Chomsky's approach to the apparent bleeding of dependencies across phase boundaries bears a striking resemblance to conditions from older theories. Under the Subadjacency Condition, for example, transformational movement could only proceed within a certain type of domain, but could escape one domain by moving first to a position at its left edge. While this way of opening up the walls of phases may be empirically desirable, it significantly reduces the explanatory force exerted by the derivational modularity of the distinct phases. Indeed, the ease with which Chomsky is able to introduce an escape hatch to phases emphasizes the fact that in his system, phase immutability does not arise from any general property of the grammatical derivation. Once the grammatical architecture permits the internal constituents of distinct phases to interact with one another, the degree of such interaction becomes simply a matter of stipulation.¹⁴

The main proposal of this book is that there is in fact a way to maintain the idea that the locality of movement-derived syntactic dependencies stems from the derivational independence of structural units that

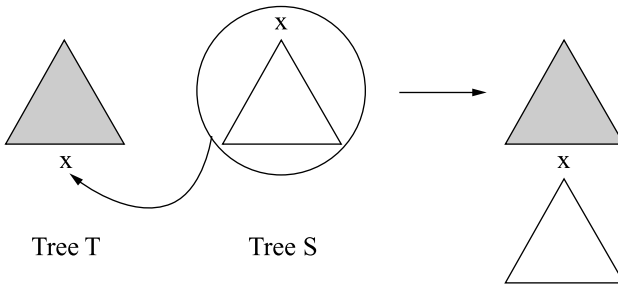
are bounded in size. The key to doing this, I argue, lies in reconsidering the set of operations that the grammar makes available for structural composition. I propose that we make use of the derivational machinery of Tree Adjoining Grammar (TAG), a formal grammatical system that bears certain interesting relations to Chomsky's oldest and most recent derivational models. By using the structural composition operations that TAG provides—namely, *Adjoining* and *Substitution*—we can overcome the problems posed by examples like (11), without sacrificing the strict separation between independent derivational units. Such cases, rather than involving iterated movement across multiple structural domains, necessitate only local movement over a single derivationally distinguished structure, which is then combined with other independently derived domains using the TAG machinery for structural composition. We will see how this conception of the grammar leads to considerable simplification in the principles of grammar. The remainder of this chapter outlines the fundamentals of the TAG formal system and sketches its potential role in linguistic description.

1.2 TAG Basics

Tree Adjoining Grammar (TAG) (Joshi, Levy, and Takahashi 1975; Joshi 1985) was developed some twenty-five years ago as a mathematically restrictive formulation of a mechanism for structural composition, inspired in part by Chomsky's earlier work on generalized transformations. Unlike the well-known grammar formalisms from the Chomsky hierarchy that operate by rewriting strings (i.e., regular, context-free, context-sensitive, and unrestricted grammars), TAG is a system of tree rewriting in which a derivation manipulates a set of predefined pieces of tree structure, called *elementary trees*. During a TAG derivation, the elementary trees are expanded and combined with one another, a conception that is closely related to frameworks in which structures are built up through generalized transformations.

TAG provides two operations for the expansion of an elementary tree *T*. The first of these, *Substitution*, rewrites a node *N* along *T*'s periphery, or *frontier*, as another tree *S* that is rooted in a node having same label as *N* (say, *X*). Alternatively, one can think of Substitution as a tree combination operation in which the root of a structure *S* is identified with a node *N* on *T*'s frontier. Such an application of Substitution is depicted schematically in (12).

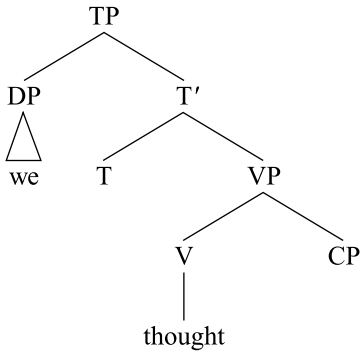
(12)



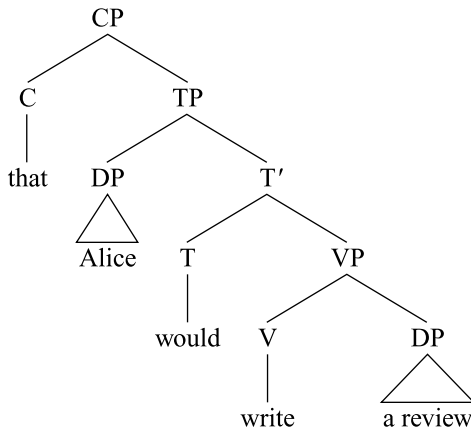
For an application of Substitution like this, we say that *S is substituted into T at N*.

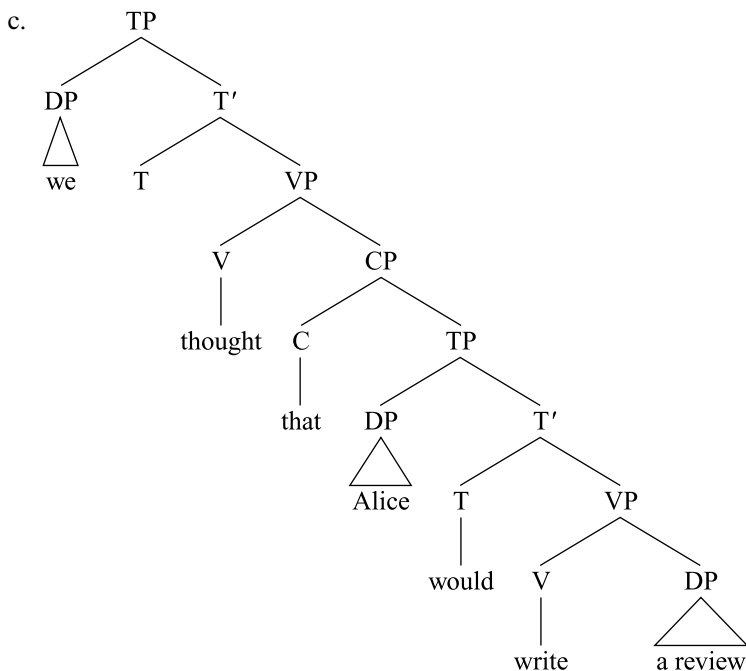
Substitution can be used to derive complex sentences involving clausal complementation. Substituting the tree in (13b) into the CP complement node of the tree in (13a) produces the structure in (13c).¹⁵

(13) a.



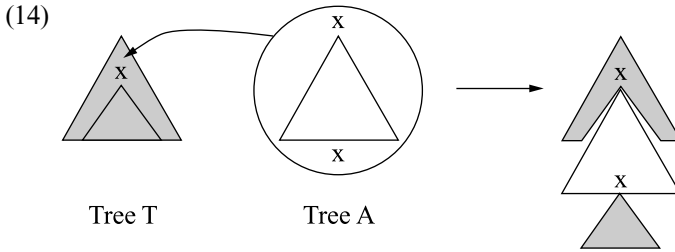
b.





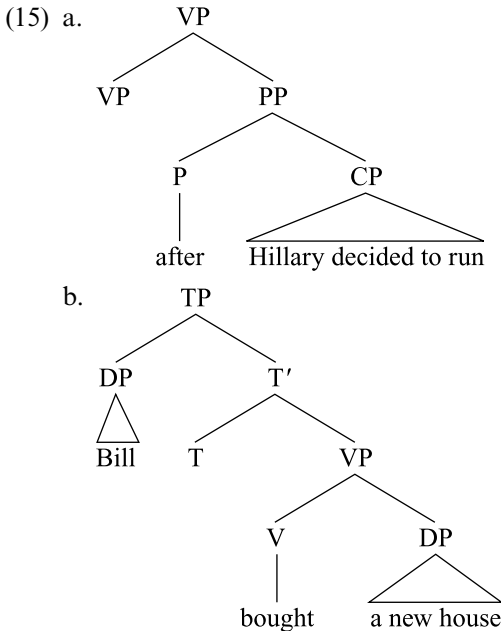
Used in this way, Substitution accomplishes effects similar to those of (some of) the generalized transformations from Chomsky 1955 and the Merge operation from Chomsky 1995: it inserts XPs into the argument positions of syntactic predicates. Substitution differs from Merge, although not from the original generalized transformation proposals, in that the argument site “inhabited” by the substituted elementary tree in the derived structure is present prior to Substitution, while such a position is created under Merge.¹⁶

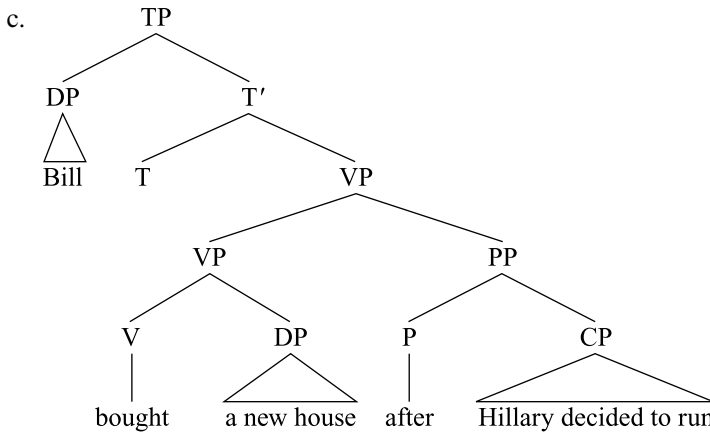
While Substitution rewrites or expands only nodes along the frontier, the second TAG operation, *Adjoining*, is capable of rewriting or expanding any node in an elementary tree. To do this, Adjoining makes use of a special class of recursive structures called *auxiliary trees*. An auxiliary tree is a structure whose root is labeled identically to some node along its frontier, the *foot node*. Given an auxiliary tree *A* recursive on *X*, Adjoining operates by rewriting as *A* some node *N* within an elementary tree *T* that is also labeled *X*. Any structure that originally appears below *N* in *T* is attached below the foot node of *A* in the derived phrase marker. This is depicted schematically in (14).



When such adjoining takes place, we say that *A adjoins to T at N*. As with Substitution, Adjoining can also be conceived of as a tree-combining operation. In such terms, adjoining at a node N labeled X of an elementary tree T first removes the subtree of T dominated by N, then attaches an auxiliary tree A in place of T, and finally reattaches the subtree of T to the foot node of A.¹⁷

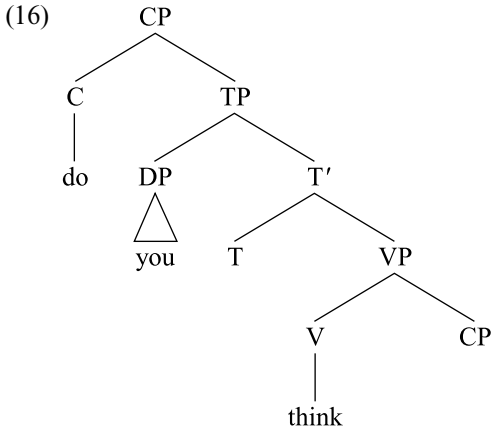
Adjoining is crucially involved in the TAG derivation of a variety of grammatical structures. The first of these is the class of structures underlying modification. By adjoining a VP recursive auxiliary tree like that in (15a) to the VP node of the clausal structure in (15b), we derive the temporally modified clausal structure in (15c).





Here, the VP recursion between the root and foot nodes of the auxiliary tree is used to introduce a Chomsky-adjunction structure into the VP in the main clause.

This use of Adjoining in constructing adjunction structures suggests a possible similarity between the TAG Adjoining operation and the adjunction operation familiar from recent work (Lebeaux 1988; Chomsky 1993, 1995). Despite the similarity of name and the overlap in function between the two operations, however, there are a number of significant differences between the two. The first is analogous to the difference observed earlier between the operations of Substitution and Merge. During application of Adjoining, no nodes are added, as the “modification” auxiliary tree in (15a) already includes both segments of the VP to which it attaches. A second difference derives from the greater generality of Adjoining. The derivation in (15) makes use of a restricted form of auxiliary tree in which the foot node is the child of the root. When there is instead greater structural distance between the root and the foot of an auxiliary tree, the output of Adjoining no longer resembles that of adjunction. To see why not, consider again the generation of a structure involving clausal complementation, like that in (13c). Observe that if the matrix clause is represented with a CP-rooted elementary tree, as might be the case in a tree like (16) where an auxiliary verb has been moved to C, this elementary tree has the necessary recursive structure to function as an auxiliary tree: the CP complement is categorially identical to the structure’s root.



Now, although the CP embedded clause in (13b) may still be substituted into the CP complement position of (16), this combination may also take place by adjoining (16) to the CP root node of (13b). Quite clearly, this structure could not have resulted from an application of the traditional adjunction operation. As a result of these differences, I will use the term *Adjoining* to refer uniquely to the TAG operation, reserving *adjunction* to refer to the more traditional operation.

Conspicuously absent from this review of the TAG machinery has been any mention of transformations. In fact, this is the most central difference between TAG derivations and those in both Chomsky's oldest proposals and his most recent ones. As formally defined (see section 1.3), TAG derivations operate by combining a fixed set of elementary trees with Adjoining and Substitution. I assume that the incorporation of TAG into a grammatical theory entails that this derivational structure should remain unchanged. That is, in a TAG-based theory, no other operations, whether transformations or structural composition, may be interleaved with applications of Adjoining and Substitution. Thus, it is not possible in the TAG context to create dependencies spanning two elementary trees via the application of transformational movement.¹⁸ Since we take operations of structural composition not to be implicated in the creation of syntactic dependencies, this means that any dependencies that are expressed in a syntactic representation R must be expressed within the elementary trees that make up R. This observation leads to the fundamental hypothesis underlying the application of TAG to syntactic theory:

(17) *The fundamental TAG hypothesis*

Every syntactic dependency is expressed locally within a single elementary tree.

In evaluating the fundamental TAG hypothesis, it is crucial that we know what constitutes the syntactic domain of an elementary tree. Yet, to this point, we have said nothing about this topic. Since elementary trees are not composed during the TAG derivation but are instead provided to the TAG operations in their final form, the TAG formalism itself has nothing to say about what they will look like. If we are to apply the TAG machinery to grammatical theory, then, we must provide some independent specification of the elementary trees that make up the grammar of a language. Since we take Substitution and Adjoining to be a universal component of the grammatical architecture, any differences that exist among the grammars of different languages must reside entirely in what elementary trees they take to be well formed. Since this set of elementary trees must be finite for any particular language, one could in principle specify the set of elementary trees that are present in the grammars of English, Italian, Japanese, and so on, merely by listing them. Of course, it would not be surprising if such a listing approach turned out to be the best way at present for constructing grammars for practical applications, given our limited understanding of abstract grammatical principles.¹⁹ However, explanatory adequacy demands that we do more than this. We must characterize the commonalities and limited differences that exist among grammars, with the aim of overcoming the argument from the poverty of the stimulus. I assume, therefore, that a TAG-based grammatical theory must include some additional component that determines the well-formedness of elementary trees in a principled fashion.

In developing a theory of elementary tree well-formedness, we will be guided to a large degree by the fundamental TAG hypothesis in (17): our conception of elementary trees must allow for the necessary localization of dependencies. However, the adoption of the TAG operations does not implicate any particular conception of a theory of elementary tree well-formedness. One can imagine a variety of ways in which such a theory might be expressed, ranging from transformational derivations to unification-based constraint satisfaction to optimality calculations to categorial inference. In fact, the TAG formalism is perhaps unique in having attracted a rich variety of perspectives on the proper characterization of structural well-formedness including Lexicon Grammar (Abeillé

1988, 1991; Abeillé and Schabes 1989), Head-Driven Phrase Structure Grammar (Kasper et al. 1995), Categorical Grammar (Joshi and Kulick 1997), and Government-Binding Theory (Kroch 1989b; Kroch and Santorini 1991; Frank 1992; Hegarty 1993a,b).²⁰ Regardless of which of these is chosen, the basic TAG architecture constrains any mechanism or well-formedness condition to apply strictly within an elementary tree.

I will adopt here the general perspective of principles-and-parameters theory in which universal linguistic principles, as instantiated by the values of parameters set for a given language, determine which elementary trees are licit in a TAG derivation. Beyond this, I assume that the specification of well-formed elementary trees is given at least in part in terms of a derivational process. In chapter 4 especially, I will entertain the hypothesis that elementary trees are constructed using derivations much like those considered in Chomsky 2000, involving Merge and Move. This gives rise to the model depicted in figure 1.4. From this perspective, the set of elementary trees that may take part in TAG derivations (i.e., that are combined by Adjoining and Substitution) in a given language has no more status than the set of well-formed phrase markers in the theory developed in Chomsky 2000. Such a set is an entirely derivative object, and focus on it obscures the primacy of the underlying principles of grammar.²¹

Note that in the model of figure 1.4, the Merge/Move portion of the derivation cannot interact with the TAG portion of the derivation, involving applications of Adjoining and Substitution. That is to say, Merge and Move may manipulate representations only as large as the domain of a single elementary tree. Once the derivation reaches a stage in which it has constructed a representation larger than this, the only operations that may apply are the mechanisms for structure composition provided by the TAG formalism. TAG elementary trees, then, provide the sort of intermediate structural domain long missing from grammatical theory that determines the structural context in which transformational operations may apply.

The degree to which localizing Merge and Move to elementary trees is empirically desirable obviously depends to a great degree on what one takes to be the domain of an elementary tree, a topic to which I turn in chapter 2. For the moment, let me tentatively suggest that elementary trees should be thought of along the lines of the kernel sentences in Chomsky's (1955) model, essentially clausal in extent. This suggestion

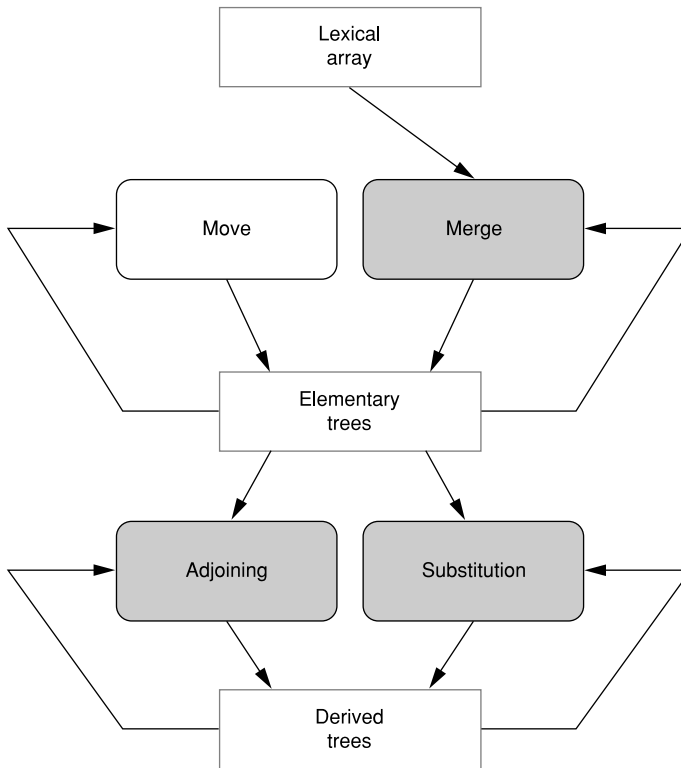


Figure 1.4
TAG-based derivational model

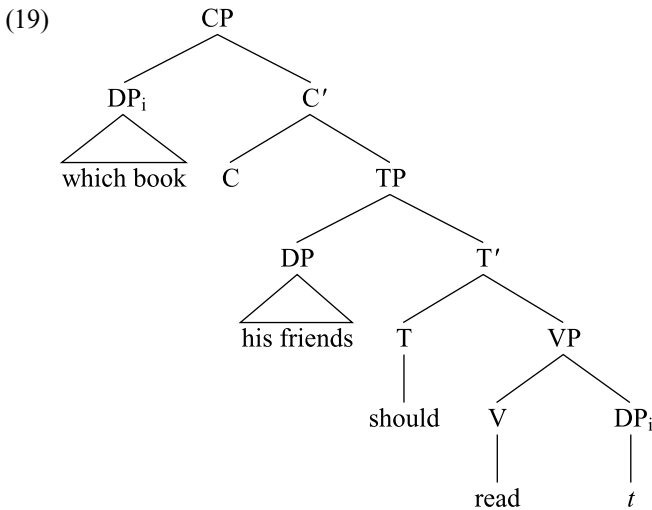
goes some way toward supporting the fundamental TAG hypothesis, as it allows the thematic dependencies of the predicate heading the clause to be localized within an elementary tree. Moreover, we can see why it is crucial that elementary trees like the one in (13a) must have a preexisting position for their complement: it yields a structural basis for the expression of the verb-complement dependency within the elementary tree. Similarly, the existence of the lower VP segment within the auxiliary tree in (15a) allows for the structural expression of a predication relation between the PP and the VP within this tree.

Unfortunately, as soon as we set our sights on a wider range of phenomena, the tenability of the fundamental TAG hypothesis seems to erode, as there are a great many syntactic dependencies that can cross

clausal boundaries. For example, the relationship between the base and surface positions of a phrase that undergoes *wh*-movement, topicalization, or raising can span a number of clauses, as seen in the following examples:

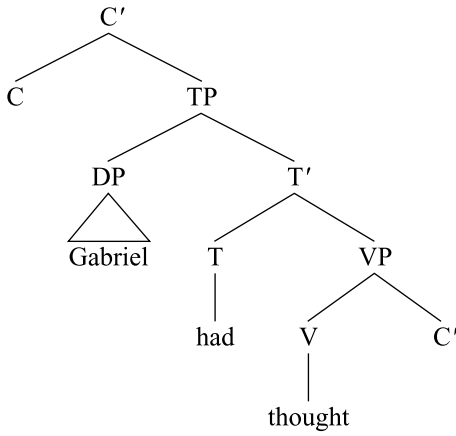
- (18) a. (I wonder) [which book] Gabriel had thought his friends should read *t*.
 b. [A meal cooked by Steve], I can't believe that you would turn down *t*.
 c. [That tyrant] is likely *t* to defeat Alice in the election.

It is assumed in this and all other linguistic investigations in the TAG framework that such violations of the fundamental TAG hypothesis are only apparent. In fact, there is a natural TAG derivation of examples like those in (18). To derive (18a), for example, the elementary tree representing the embedded clause will already contain a dependency between the fronted DP and the position of its trace.

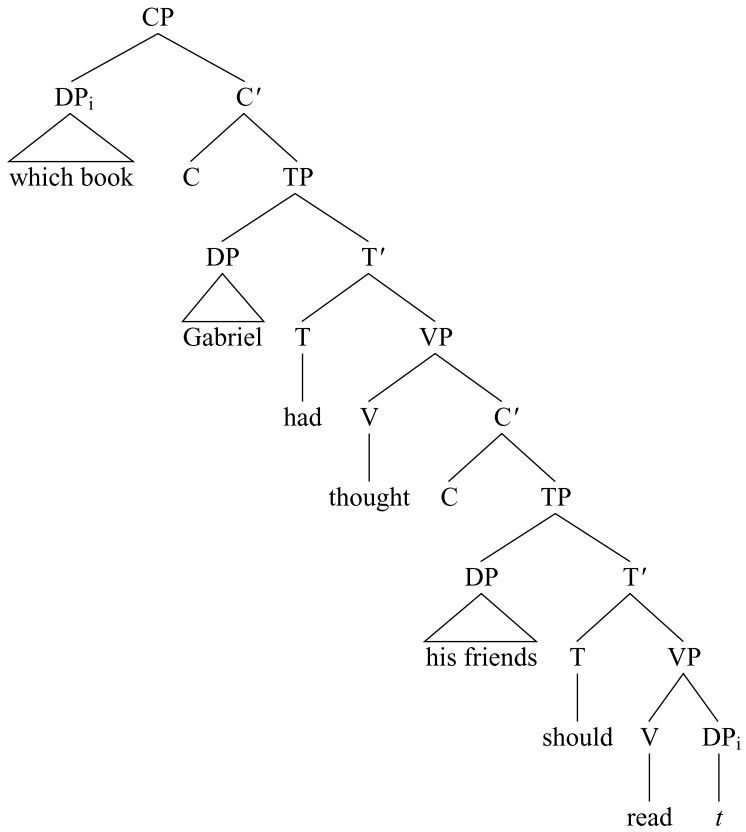


The apparent nonlocality of this dependency is created by adjoining the auxiliary tree in (20a) to the *C'* node in (19). The result is the structure in (20b).²²

(20) a.



b.



By interposing the auxiliary tree in (20a) between the *wh*-moved DP and its trace, we effectively stretch an originally local relation to one that is no longer clause bounded. As we shall discuss at length in subsequent chapters, similar derivations are possible for the other cases in (18). For (18b), we use Adjoining to interpose an auxiliary tree containing the lexical material *I can't believe* between the topicalized DP *a meal cooked by Steve* and the clause *that you would turn down* that form part of the same elementary tree. Likewise, for (18c), we adjoin an auxiliary tree containing the lexical material *is likely* between the subject DP *that tyrant* and the rest of its clause *to defeat Alice in the election*.

Recall that Adjoining functions by rewriting some node of an elementary tree as a recursive piece of structure, an auxiliary tree. This means that whenever local dependencies are stretched in the manner just sketched, such stretching may result only from the introduction of recursive structure. Thinking about this from the point of view of decomposing nonlocal dependencies, one can state the following corollary of the fundamental TAG hypothesis:

(21) *Nonlocal dependency corollary*

Nonlocal dependencies always reduce to local ones once recursive structure is factored away.

Much of the remainder of this book will be devoted to showing that this corollary accurately characterizes the types of dependencies present in natural language.²³

1.3 The Structure of TAG Derivations

With the basic ideas of TAG laid out, let us now turn to the task of characterizing the notion of TAG derivation. Note first of all that by *TAG derivation*, I mean only the combination of elementary trees via Adjoining and Substitution and not the process of elementary tree construction using Merge and Move envisioned in figure 1.4. The basic intuition is this: a TAG derivation consists of a sequence of combinations of elementary trees using Adjoining and Substitution. To formalize this idea, we will make use of a representation of the sequence of derivational steps, called a *derivation structure* (Vijay-Shanker 1987). The idea of a derivation structure has a long history within generative grammar, going back to the T-marker in the theory of Chomsky 1955. A TAG derivation structure is a tree in which each node corresponds to an elementary tree.

The daughters of a given node *N* represent the trees that are adjoined or substituted into the elementary tree represented by *N*. Since there may be ambiguity about where in an elementary tree another tree is substituted or adjoined (owing to the presence of multiple nodes with the same categorical label within a single elementary tree), the links connecting any pair of nodes are annotated with the location in the mother elementary tree where Adjoining or Substitution has taken place.

Let us briefly consider some examples of derivation structures. For the VP modification derivation in (15), the root node of the derivation structure will represent the main clause elementary tree. Its only daughter will be a node corresponding to the VP modifier auxiliary tree. This yields the derivation structure in (22).

- (22) Bill bought a new house
 VP |
 after Hillary decided to run

The picture becomes more interesting when we consider the somewhat more complex example in (23).

- (23) Bill bought a new house after Hillary decided to run because their fight was over.

This sentence is ambiguous: the *because* adjunct clause may be construed as modifying either the act of buying or the act of deciding. Let us assume for present purposes that the *because* adjunct is introduced into the derivation with a TP-modifying auxiliary tree, similar in structure to the one in (15a). The two readings of the sentence are distinguished by their derivations. The first derivation involves the *because* tree first adjoining into the *after* tree, with the result adjoining into the *bought* tree. This derivation is depicted in (24).

- (24) Bill bought a new house
 VP |
 after Hillary decided to run
 TP |
 because their fight was over

In the second derivation, both adjunct clauses adjoin separately into the main clause elementary tree at the VP and TP nodes. The resulting derivation structure, shown in (25), is quite different in shape.

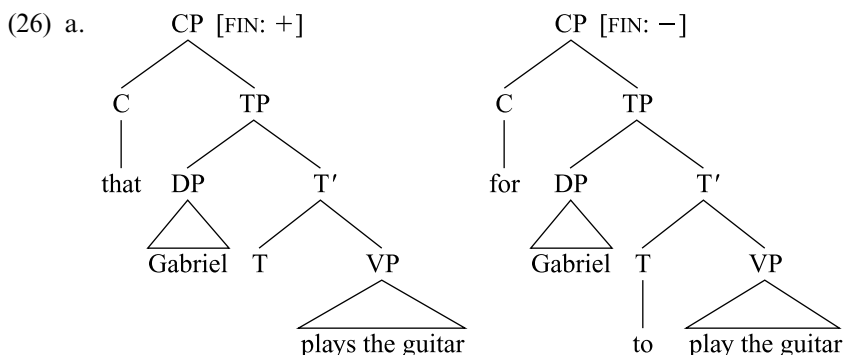
- (25)
- | | |
|------------------------------|------------------------------|
| Bill bought a new house | |
| VP | TP |
| after Hillary decided to run | because their fight was over |

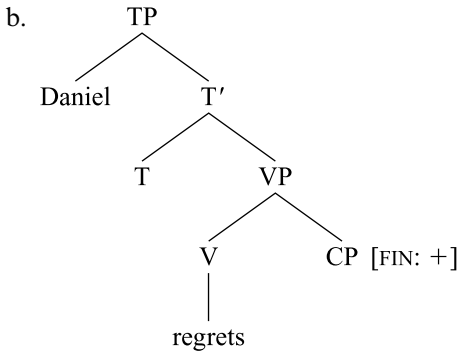
The fact that the derivation structure concisely encodes the distinction between the two readings suggests that we might want to follow Chomsky's (1955) line in taking the derivation structure (T-marker) to constitute the interface with the interpretive component.²⁴ Derivation structures have largely been ignored in recent work in the Minimalist Program, perhaps because in a system that builds structure one level of projection at a time, as with the Merge operation, the derivation structure is to a large extent indistinguishable from the derived structure. It is therefore unclear whether it is the derived phrase marker or the derivation structure that is the object of grammatical interest in these more recent proposals. If I am correct in taking the derivation structure to be the interpretive interface, this raises the question of what role, if any, is played by the derived phrase marker. Though pursuing this matter would take us far afield, it may be that the derived phrase marker's unique function is that of providing an input to the phonological component. I return briefly to such speculations in chapter 6.

Once the TAG derivational system is provided with a set of elementary trees, any combination of these trees using Substitution and Adjoining will be representable in terms of a derivation structure. This free combinability is, however, subject to one formal restriction that ensures that derivation structures maintain a certain formal simplicity. Recall that since a derivation structure is a tree, the daughters of any node in the derivation structure may have daughters of their own. That is, they may themselves be the locus for the adjoining or substitution of other elementary trees. We have already seen an example of such embedding in the derivation structure in (24). Note, however, that neither this nor any other derivation structure specifies whether the derivation has taken place in a top-down or a bottom-up fashion. For the derivation structure in (24), one can imagine either that the two auxiliary trees have first combined, the result being adjoined into the main clause elementary tree, or that the *after* auxiliary adjoins first into the main clause, the *because* auxiliary then being adjoined into this complex. There is little reason to prefer one of these derivations over the other given the derivation structure, and it therefore seems reasonable to assume that both derivations

ought to be possible. To guarantee that this will always be the case, it is sufficient to require that every combination of elementary trees τ and τ' indicated by a mother-daughter relation in a derivation structure must be possible independently of other combinations indicated by the derivation structure. For the derivation structure in (24), this requirement has the effect of ensuring that the combination of the *after* auxiliary tree with the *bought* elementary tree is possible independently of the prior combination of the *after* and *because* auxiliary trees. This restriction on possible derivations imposes a context-free or Markovian character on these derivation structures, and indeed it can be proven that TAG derivation structures are strongly context free. I return to empirical implications of this restriction in chapter 3.

In a TAG-based grammatical theory, certain grammatical constraints will turn out to have their effects by imposing additional restrictions on derivations. A simple example arises in enforcing selectional properties, as in a verb like *regret*'s requirement that its CP complement be finite. Since the grammar presumably contains both finite and nonfinite CP-rooted elementary trees, we will need to find some way of permitting only finite ones to be inserted—say, via Substitution—into the complement position of an elementary tree containing the verb *regret*. Let us make the standard assumption that nodes in a phrase marker bear certain features. In particular, let us assume that the CP roots of elementary trees like those in (26a) contain a specification of the finiteness of these clauses, and that the CP frontier node of the elementary tree in (26b) bears a finiteness feature as a result of *regret*'s selectional properties.²⁵





Suppose now that in identifying the root of the substituted structure with the frontier node into which substitution takes place, Substitution also merges the feature sets of the two nodes, where this Merge operation is understood as feature unification. This yields the desired result that certain substitutions that would be possible on the basis of node label compatibility are in fact no longer possible. In the case at hand, the nonfinite CP in (26a) cannot substitute into the CP node of (26b), because of the feature clash. Other derivational constraints might arise from interface conditions, in the sense of Chomsky (1993, 1995). Suppose, for example, that we take the conceptual system that interfaces with syntax to be capable of interpreting only structures that carry complete propositional interpretations. This would mean that the output of a derivation that failed to produce a phrase rooted in CP would not be interpretable. Finally, as I will discuss in detail in chapter 4, a class of restrictions on applications of Adjoining derives from a certain notion of derivational economy.

In the context of this discussion, it is important to note that the TAG formalism tolerates only constraints on derivations that have a local character. That is, the well-formedness of a derivation must be determinable on the basis of consulting only mother-daughter relations between elementary trees in the derivation structure. The formalism does not permit restrictions on derivation structures that make reference to global structural properties (e.g., binary branching or restrictions on certain c-command relations) or global constraints on the derived phrase markers. This limitation of the TAG formalism has a significant impact on the nature of grammatical constraints and processes that may form part of a TAG-based theory of grammar: they may only specify properties of individual elementary trees.²⁶

This limitation on allowable grammatical principles is quite restrictive, but appears to capture the kind of constraints that are generally taken to hold in human grammars. Indeed, truly global conditions on syntactic well-formedness have rarely been proposed, and for those that have been, alternative analyses suggest that they should be seen as extrasyntactic constraints. For example, binding theory Condition C, which requires that names, or R-expressions, not be c-commanded by a coreferential element, has been reconceptualized as a morphological condition (Burzio 1989), pragmatic constraint (Reinhart 1986), or interpretive principle (Chomsky 1993). If one of these suggestions is on the right track, then the fact that TAG cannot encode global principles provides empirical support for the link between the formal restrictiveness of the TAG formalism and the properties of human grammar.

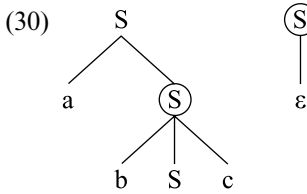
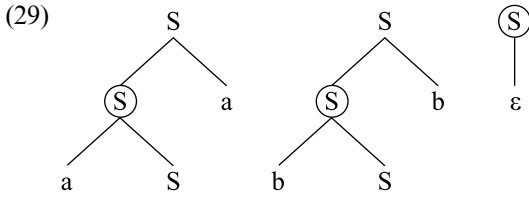
1.4 Formal Grammar and Human Grammar

In addition to investigations of the sort we are engaged in here concerning the relevance of TAG to linguistic theory, the TAG formal system has been well studied as a mathematical object (Joshi 1985; Vijay-Shanker 1987). Perhaps the central question in such work on formal grammars concerns generative capacity, that is, the range of languages for which a given formalism can provide grammars. Language can be understood here as a set of strings, in which case we talk of weak generative capacity, or a set of structural descriptions, in which case we talk of strong generative capacity. For the formalisms in the Chomsky hierarchy, generative capacity has been well characterized. It is known, for example, that while context-free grammars can be given for the string languages in (27), none can be given for the closely related string languages in (28).

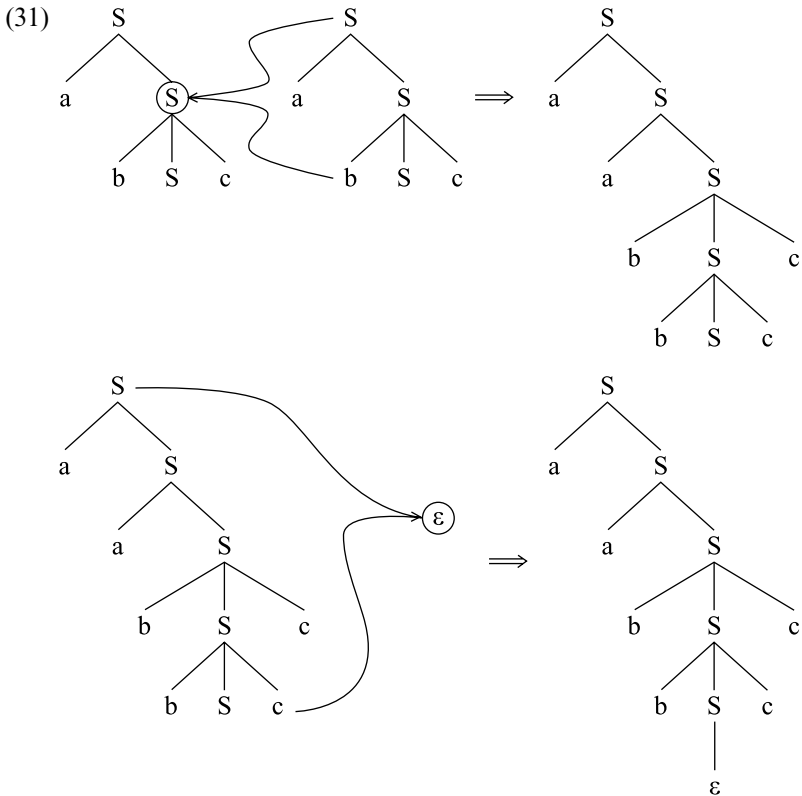
- (27) a. $L_1 = \{ww^r \mid w \in \{a, b\}^*\}$ (where w^r is the reversal of w)
 b. $L_2 = \{a^n b^n \mid n \in \mathbb{N}\}$

- (28) a. $L_3 = \{ww \mid w \in \{a, b\}^*\}$
 b. $L_4 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$

Where does TAG fall with respect to generative capacity, then? It is not difficult to show that every context-free (string) language is also a tree adjoining language (TAL). Moreover, for each of the languages in (28), there is a TAG (i.e., a set of elementary trees) that generates it. These are given in (29) and (30) for the languages L_3 and L_4 , respectively.



The grammar in (30) generates strings in L_4 by repeatedly adjoining one copy of the auxiliary tree on the left to another at the circled S node. For each such adjoining, the number of as , bs , and cs increases by one. Finally, to complete the derivation, the derived auxiliary tree adjoins to the S node of the ϵ -tree. Such a derivation for the string $a^2b^2c^2d^2$ is depicted in (31).



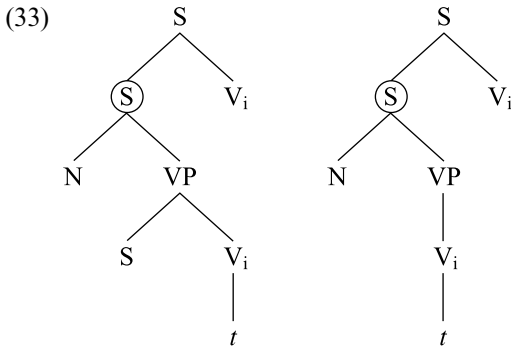
It should be clear how this sort of derivation can be extended to generate any string in L_4 . To ensure that this grammar does not generate strings outside of L_4 , we must guarantee that the *as*, *bs*, and *cs* remain properly partitioned. To do this, we must prevent Adjoining from taking place anywhere other than at the nodes indicated in this derivation. This can be accomplished by adopting the system of derivational constraints proposed by Vijay-Shanker and Joshi (1985), which allows the nodes of an elementary tree to be marked as null adjoining (NA), indicating that nothing may adjoin at that node. Adopting the convention that all non-circled nodes in an elementary tree are marked NA, the grammars given in (29) and (30) indeed generate L_3 and L_4 .

Though the weak (and strong) generative capacity of TAG extends beyond that of context-free grammars, its extension into the realm of the context-sensitive languages is extremely limited. So, the languages in (32), while closely related to those in (28) and well within the realm of the context-sensitive languages, are not generable by any TAG (Vijay-Shanker 1987).

- (32) a. $L_5 = \{www \mid w \in \Sigma^*\}$
 b. $L_6 = \{a^n b^n c^n d^n e^n \mid n \in \mathbb{N}\}$

The potential linguistic interest of discussions of generative capacity lies in the degree to which a limitation on generative capacity plays a role in characterizing grammatical competence. As Chomsky has pointed out on numerous occasions (e.g., Chomsky 1965, 60–62; 1981, 11–13), there is no reason a priori to expect that formal properties like generative capacity should be relevant to such a characterization. Indeed, for the classes of grammars in the Chomsky hierarchy, the corresponding generative capacities do not appear to match up with the properties of natural language in any interesting way. Chomsky (1956, 1957) demonstrates that regular grammars are not sufficiently powerful in either their weak or their strong generative capacity to describe natural language, and similar arguments have been made more recently for context-free grammars (Culy 1985; Shieber 1985). The next step in the hierarchy, context-sensitive grammars, is however so expressive that it offers little limitation on what could count as a natural language. It is interesting to note, however, that the “mild” context-sensitivity of TAG is sufficient to allow treatment of grammatical phenomena whose analysis has been shown to lie beyond the power of context-free grammars.²⁷ One such case, discussed by Culy (1985), concerns a reduplication phenomenon in Bambara. Culy

shows that this gives rise to a sublanguage that is in formal respects identical to L_3 , a language for which a TAG is already provided above. Shieber (1985) brings up the case of Swiss German cross-serial dependencies, in which nominal arguments and their associated verbs appear in crossing rather than nested orders. A TAG of the form in (33) is adequate to generate sentences of this type, assigning them well-motivated structural descriptions (Kroch and Santorini 1991).²⁸



I take the possibility of analyzing these non-context-free phenomena within TAG to suggest that there is a correspondence between the limited formal power of TAG and the expressive demands of natural language.²⁹

Let me emphasize that I am not contending that the restricted weak or strong generative capacity (or any other formal property) by itself recommends the use of some grammatical formalism in a theory of human linguistic competence. Similarly, I am not claiming that the existence of a polynomial-time parsing algorithm lends any increased plausibility to the place of some formalism in human grammar (cf. Gazdar et al. 1985). Rather, I am claiming that by considering a significant range of empirical data, we are led to conclude a posteriori that exactly the sort of formal restrictiveness embodied in TAG constitutes a part of grammatical competence (cf. Chomsky's (2000) argument for limited computational complexity, appropriately defined, in the grammar).³⁰ As discussed above, the adoption of TAG substantially limits the range of possible grammatical principles to those that can be expressed over the domain of elementary trees. As we find ourselves continually able to analyze grammatical phenomena in these terms, our confidence that TAG forms part of the grammar should correspondingly increase. A second type of support for the adoption of TAG comes from the fact that TAG's formal restrictiveness allows us to simplify or entirely eliminate the statement of previously

complex grammatical restrictions. For example, as I will discuss in chapter 5, there is no need in a TAG treatment of *wh*-movement for an analogue of a principle governing the locality of movement like Subjacency, as its effects already follow from the mechanisms of the formalism.

The discovery that human language can be simply and naturally characterized in terms of the limited formal power of TAG, if correct, would reveal a rather abstract and very surprising property of human language indeed. Therefore, just as any grammatical theory must explain the existence of other empirically discovered constraints, so we should expect that any theory should also explain why the computational system of human language is limited in its formal power.