Primitive C-Command*

Robert Frank K. Vijay-Shanker

Abstract

Work in syntactic theory almost universally assumes that hierarchy in syntactic structure is characterized in terms of an abstract primitive relation of dominance. In this paper, we suggest that hierarchy should instead be determined via a primitive c-command relation. This perspective turns out to restrict the range of possible syntactic structures in a linguistically natural way, deriving restrictions on possible configurations that must otherwise be stipulated. Furthermore, we show that a primitive c-command view of syntactic structure provides the basis for a radically simplified conception of adjunction structure, one which not only allows us to understand why the adjunction operation exists alongside substitution, but also explains why these operations have their distinctive structural and derivational properties.

keywords: phrase structure, c-command, dominance, branching, adjunction, derivations

1 Introduction

Any descriptively adequate grammatical theory must precisely specify the well-formed representations for the sentences of all possible human languages. Part of this task involves determining what kind of formal object constitutes the basis for grammatical representations. From the outset of generative grammar and in almost all subsequent work within the transformational paradigm, it has been assumed that phrase structure should be represented in terms of certain type of abstract graph, a *tree*, whose leaves correspond to lexical items and whose internal connectivity, what is often called *constituency*, reflects a range of distributional facts. Yet, the question of what exactly tree structures are formally speaking has been relatively neglected in the literature on syntactic theory. There seems to be widespread, if only implicit, adoption of the view sketched in Partee, ter Meulen, and Wall (1993), inter alia, in which trees are determined by two primitive relations: dominance and precedence. Yet, so far as we are aware, there have been no empirical arguments adduced in favor of this choice.

In this paper, we argue that the choice of dominance and precedence as primitives for syntactic description brings with it a number of puzzling dilemmas for syntactic theory, all centering around the over-expressiveness of these relations. These dilemmas motivate our proposal that hierarchical structure be obtained in terms of a different primitive structural relation, namely c-command. In section 4, we investigate the formal issues involved in characterizing phrase structure and constituency in terms of c-command, and in section 5 we compare the expressiveness of this system to one utilizing primitive dominance and precedence.[1] In particular, we show that the class of structures that can be characterized in terms of c-command is in fact quite restricted, and corresponds closely to the trees permitted under Kayne's (1994) proposed Linear Correspondence Axiom (LCA). We take this to show, then, that primitive c-command imposes a linguistically sensible restriction on the topologies of phrase structure trees. In section 6, we leave behind the conservative notion of tree structure, and show that primitive c-command leads to a radically new conception of adjunction structures. Finally, section 7 explores the import of primitive c-command in a derivational view of grammar, and demonstrates how the novel conception of adjunction provides new insight into cyclicity differences between the adjunction and substitution operations.

2 What's in a tree?

To begin, it will be useful to recall the representational desiderata that have motivated the choice of trees as representations of phrase structure. It seems to us that there are at least three properties of natural language syntax that are assumed to be derived from the nature of phrase structure. First, as demonstrated by a variety of distributional tests, sentences consist of groupings of elements, called constituents, which are embedded hierarchically one inside the other, to form one all encompassing grouping. Second, these constituents are (at least partially) ordered as reflected by the sequence in which they are pronounced. Thirdly, in order to account for similarity in distribution, constituents are labeled by their grammatical type; constituents of identical type have essentially identical distribution.

As mentioned above, these desiderata have driven syntacticians to exploit trees as the formal objects underlying the theory. Let us consider, then, what is meant precisely by a tree.[2] The following definition, adapted from Partee, ter Meulen, and Wall (1993), is probably that which is most widely assumed:[3]

(1) A tree is a 5-tuple $\langle N, Q, D, P, L \rangle$, where

N is a finite set of nodes

Q is a finite set of labels

D is a weak partial order (i.e., a reflexive, antisymmetric and transitive relation) in $N \times N$, the *dominance* relation

P is a strict partial order (i.e., an irreflexive, asymmetric and transitive relation) in $N \times N$, the *precedence* relation

L is a function from N into Q, the labeling function

How does this formal structure satisfy the representational desiderata mentioned above? The fundamental notion of constituent is embodied directly in the concept of node. Similarly, the requirement that constituents be assigned a grammatical type is given a direct implementation via the labeling function L.

What, then, about the need to group constituents hierarchically? This is represented by the the dominance relation D. If a node x is part of the constituent represented by node y, we say that y dominates x, or more formally $(y, x) \in D$. Our intuitions about constituency tell us that if node x forms a proper subpart of constituent y, y cannot in turn be part of constituent x. Thus, the dominance relation must be antisymmetric. Similarly, if a constituent x is part of constituent y, and y is part of constituent z, then x is part of z. Hence, dominance must be transitive.[4] These formal properties do not, however, suffice to encode our intuitions about constituency. One deficiency concerns the fact that there is no guarantee that entire sentence will form a single constituent. Thus we need to impose the following additional requirement:

(2) **Rootedness condition**: $\exists x \forall y [xDy]$

"There is a node which dominates all other nodes."

The final desideratum, that constituents be ordered with respect to one another, is instantiated in the precedence relation P: we interpret x precedes y, or $(x, y) \in P$, as the statement that x is pronounced before y. As before, our intuitions tell us that if x is ordered before y, that is x is pronounced before y, then y cannot be pronounced before x. Hence precedence must be asymmetric. Similar arguments suffice to establish the irreflexivity and transitivity of precedence. While this formalization of precedence embodies intuitions about ordering, it allows for the possibility that certain constituents may be unordered with respect to one another. In particular, if a constituent x is part of another constituent y, there is no sense to asking whether x is pronounced before or after y. Instead, pronouncing x is part of pronouncing y. Thus, we need to assert the following additional condition:

(3) **Exclusivity condition**: $\forall x, y[(xPy \vee yPx) \leftrightarrow (\neg xDy \wedge \neg yDx)]$
"For any pair of nodes, they stand in the precedence relation if and only if they do not stand in the dominance relation."

There is yet another shortcoming of the formalization to this point. Since a node's precedence relations are determined independently of the nodes that dominate it (necessarily with the exclusivity condition, since they stand in a dominance relation), we have no way of guaranteeing that the subconstituents of w all precede the subconstituents of x when w itself precedes x. Assuming that this latter property holds of trees (but see McCawley (1982)), an additional condition is needed:

(4) **Non-tangling condition**: $\forall w, x, y, z[(wPx \wedge wDy \wedge xDz) \rightarrow yPz]$
"If node w precedes node x, the descendants of w also precede the descendants of x."

Taken together, the algebraic structure in (1) and the constraints in (2), (3) and (4) characterize a class of trees structures that are typically used in a theory of syntactic representations.[5] Thus, one might conclude that the foundations of syntactic theory are in order, so we can go about building the necessary theoretical edifice that constitutes the remainder of our grammatical theory. In the next section, we question whether this conclusion is as firmly established as we might hope.

3 Puzzles of Primitives

Apart from providing a formal characterization of the class of tree structures, the relations of the structure in (1), namely dominance and precedence, are intended to constitute the primitive terms out of which grammatical theory is constructed. As is well known from the study of definability in mathematical logic, different choices of primitives can substantially affect the range of properties or well-formedness constraints that can be expressed in a formal theory. Clearly, then, in building a grammatical theory, we must choose primitives that are sufficiently expressive so as to allow all necessary constraints on well-formedness to be stated. We do not want to suggest that the relations of dominance and precedence (together with information about node labels), when taken as the primitive lexicon of grammatical theory, are insufficiently expressive to allow all of the empirically significant structures and relations to be defined.[6] Yet, this choice of primitives leaves us with no explanation for why only a small portion of the rich class of tree structures that can be specified in dominance and precedence terms are exploited in the grammars of human languages. Typically, the grammatically irrelevant structures are ruled out not by the formal representational machinery of the grammar, but instead by separately stipulated substantive principles, for example X-bar theory and the like. We take it to be uncontroversial that it would be preferable if the need for such stipulations were reduced or eliminated (cf. Epstein (1999) and Reiss (2001)). One way in which this might be done is in terms of a relatively "weak" set of primitives, under which the grammatically uninhabited structures were not definable.

Related to this overabundance of definable trees is the fact that among the myriad of structural relations definable in terms of dominance and precedence, a great many of them appear to be uninteresting from the point of view of grammatical theory. One is left wondering, for example, why in contrast to the pervasive and fundamental role played by the particular derived relation of c-command, a hypothetical relation like p-command, defined below, has no grammatical importance in spite of its formal similarity.

(5) α c-commands β iff every node properly dominating α dominates β and neither dominates the other.

(6) α p-commands β iff some node properly dominating α precedes β and neither dominates the other.

We might hope that a less expressive choice of primitives might provide us with an explanation for the grammatical irrelevance of such a notion of p-command, in virtue of undefinability.[7]

Finally, a similar point can be made regarding certain constraints on grammatical derivations, e.g., that movement must apply cyclically, or that it must proceed to a position that c-commands the base position. There is nothing inherent in a dominance and precedence-based view of structure that would predict the existence of these derivational restrictions, and they must therefore be stipulated. Once again, it is imaginable that with the right choice of primitives, these stipulations might be eliminable, with these properties would follow as corollaries from the way in which structures and operations over them are defined.

We believe that these cases of over-expressiveness show that a formalization of syntactic structure in terms of dominance and precedence does not match well with the needs of grammatical theory. These primitives substantially underconstrain the possible structures and relations that play a role in human language, and thus do not play a significant role in limiting possible grammatical variation, one of the central questions in linguistic theorizing.[8] Though there will no doubt be aspects of the human language faculty that will not be reducible to formal properties, we take it to be nonetheless desirable if our choice of primitives were to determine a class of definable relations and representations to be one that corresponds more closely to those that are exploited in natural language. The closer one gets to such a formal characterization of the possible structures and relations, the more it allows one to dispense with additional stipulations. Also, the appropriate choice of primitive would have the metatheoretic advantage that the range of analytic possibilities would be limited to those things which could be defined.

One proposal along these lines is that of Kayne (1994). Kayne shows how precedence (among terminals) can be derived from hierarchical structure, thereby eliminating the need for a separate primitive of precedence.[9] In so doing, Kayne not only limits the class of possible ordering among elements (e.g., only head–complement ordering is permitted) but also restricts the range of possible hierarchical structures, in a manner that derives the effects of certain previously stipulated restrictions on phrase structure.

In the remainder of this paper, we explore the consequences of replacing the structural primitive of dominance with a primitive relation of c-command.[10] Our reason for considering c-command stems from the observation made above concerning this relation's fundamental role in grammatical theory.[11] Nonetheless, one might wonder whether c-command is sufficiently expressive on its own to characterize the crucial representational notion of constituency. In the next section, we take up the question of how this might be done.

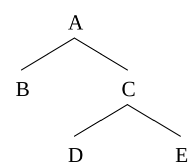
4 Changing the foundations: c-command as a primitive

As was suggested in the previous section, we would like to suggest that the role of dominance as a structural primitive be taken over by c-command. One might suppose that c-command is too impoverished as a source of structural information to adequately determine tree structure as normally understood. Though we appreciate this intuition, it does not appear to be true. Consider, for example the set of nodes and c-command relations among them given in (7).

(7) $N = \{a, b, c, d, e\}$
$C = \{(b, c), (c, b), (b, d), (b, e), (d, e), (e, d)\}$

Abstracting from ordering, it turns out that there is a single tree structure (with the same set of nodes) whose c-command relation is (7). That tree is the following:

(8)



One way of understanding what we are saying here is that the structure in (8), when understood in terms of its dominance relation, is the unique structure whose derived c-command relation is that in (7).[12] Thus, to specify the structure, it is sufficient, at least for this case, to give the c-command relation.

We can formalize this view of structure by defining a tree as specifying N, a finite set of nodes and C, the c-command relation on $N \times N$.[13] As before, we will need to impose additional conditions so as to guarantee that the primitive relation behaves in the expected way. For the time being, let us simply assume that the primitive c-command relation is one that could have arisen under a dominance-based formalization of trees given the following traditional definition:

(9) α c-commands β iff every node that properly dominates α also properly dominates β and neither dominates the other.[14]

We give this definition only to provide the reader with an understanding of what relation we intend by the name c-command. Specifying how this relation can be axiomatized without reference to a primitive dominance relation is beyond the scope of this work.

What we will focus on instead is the question of the degree to which taking c-command as a primitive allows us to satisfy the desiderata for phrase structure outlined earlier, concerning the representation of hierarchical relations and constituency. Though c-command embodies a particularly significant hierarchical relationship, it is not clear whether it allows us to express the notion of constituency. Recall that under a view of structure incorporating a primitive dominance relation, the constituent rooted in an element x is the set of nodes which were dominated by x. Alternatively, as domination defines a partial ordering among the nodes (as it is reflexive, antisymmetric and transitive), we can say that the constituent rooted in x is the set of nodes that are "greater than or equal to" x in the domination relation, taking the root to be the least element (i.e., the set of nodes at least as embedded as *x*). More formally:

(10) *constituent*$(x) \stackrel{\text{def}}{\equiv} \{y|xDy\}$

Let us try to construct an analogous definition of constituent using primitive c-command. The c-command relation on its own does not provide us with a partial ordering among the nodes, as it is neither transitive nor antisymmetric. Thus, we must build an ordering out of c-command. One way to do this might be using the following general schema for extracting an ordering $\preceq^R$ given an arbitrary relation R:

(11) $x \preceq^R y \stackrel{\text{def}}{\equiv} \forall z[zRx \rightarrow zRy]$

It is straightforward to prove that for any R, $\preceq^R$ is a relation with two of the three defining properties of a partial ordering, specifically reflexivity and transitivity. Reflexivity is trivial, as the two sides of the implication will be identical. Transitivity follows from the transitivity of implication. However, antisymmetry does not follow. Thus, while we do not obtain a partial-order, the relation we obtain is a *pre-order*. To ensure that we are dealing with a partial order, we can restrict ourselves to relations (and the corresponding structures) that meet conditions guaranteeing that $\preceq^R$ is antisymmetric. Alternatively, we can consider arbitrary relations, leaving $\preceq^R$ as a pre-order, but treating the elements x and y as equivalent when $x \preceq^R y$ and $y \preceq^R x$. This, of course, is the standard and natural way of making a pre-order into a partial ordering. Later in the paper, we will discuss the implications of each of these choices.

We can now examine the possibility of using the derived ordering relation to derive a notion of constituency. We first apply this schema to the c-command relation to yield the ordering $\preceq^C$:

(12) $x \preceq^C y \stackrel{\text{def}}{\equiv} \forall z[zCx \rightarrow zCy]$

"x is C-ordered before y iff every z that c-commands x also c-commands y."

Using this ordering, we next construct a definition of constituent analogous to that in (10):

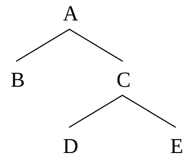
(13) *constituent*$(x) \stackrel{\text{def}}{\equiv} \{y|x \preceq^C y\}$

Although this characterization of constituent seems to have some right formal properties, its adequacy depends also on the degree to which it reflects the notion of constituent used in grammatical theory. To see how well it does this, let us apply the definitions of $\preceq^C$ and *constituent* to the c-command-based structure given in (7) above and depicted in (8) (both repeated here):

(14) a. $N = \{a, b, c, d, e\}$

$C = \{(b, c), (c, b), (b, d), (b, e), (d, e), (e, d)\}$

b.



c. $\preceq^C = \{(a,a),(a,b),(a,c),(a,d),(a,e),(b,b),(c,c),(c,d),(c,e),(d,d),(e,e)\}$

d. *constituent*$(a) = \{a,b,c,d,e\}$

constituent$(b) = \{b\}$

constituent$(c) = \{c,d,e\}$

constituent$(d) = \{d\}$

constituent$(e) = \{e\}$

Strikingly, this $\preceq^C$ relation is identical to the one we would have taken to be the primitive dominance relation for the structure depicted here under the standard conception of tree structure. As a result, our definition of constituent in terms of $\preceq^C$ behaves precisely like the definition in terms of dominance, yielding the standard constituents.

At this point, one might be tempted to label the relation $\preceq^C$ as dominance, since it appears to faithfully reconstruct the dominance relation.[15] We will follow this temptation, referring to $\preceq^C$ as a *derived* dominance relation. The reader should keep in mind that in our proposal it is the domination relation that is derivative and the c-command relation that is primitive. It is important to note as well that there is no conflict between our defining a derived domination relation and our elimination of primitive dominance from the grammar in favor of primitive c-command. The absence of primitive dominance does not exclude the possibility that a derived dominance relation might play a significant role in characterizing phrase structure, just as Kayne's (1994) work does not deny the importance of the temporal relation among constituents (i.e., precedence). Rather, we and Kayne are questioning whether these relations, as far as they are relevant, should be taken as primitives, or whether they should be derived from some other primitive relation(s). Our claim in the present work is that the only notion of domination that is grammatically significant is the substantially constrained version that can be derived from a primitive c-command relation.

While (14) appears to suggest that the notion of constituent can be derived from c-command, we can examine whether the notion of the root of a structure can be derived as well. Recall that in the dominance-based view of tree, the root is defined as the unique node that dominates all other nodes. By observing the c-command relations in (14), we note that the root node A is also uniquely distinguished by the fact that no other node c-commands it. Thinking of the definition of $\preceq^C$, it is straightforward to observe that any node which is not c-commanded will also dominate every other node, in the sense of $\preceq^C$. It is also demonstrable that the converse holds, i.e., if a node dominates every other node, then it must not be c-commanded. Thus, exactly corresponding to the notion of root as that which dominates every node (from (2)), we have the following notion of a root:[16]

(15) **Rootedness condition**: $\exists r \forall y [\neg y C r]$

"There is a node r which is c-commanded by no other node."

Observing the c-command relations in (14), we note that the node A distinguishes itself in another way; it is the only node that does not c-command any other node. We will show later that while this second condition by itself does not yield the notion of root, it is nonetheless true that in the case of trees any node x that is not c-commanded by any node will also not c-command any node. Thus, for trees, it should not matter whether we take (15) as the definition of rootedness or whether we require both that a node not be c-commanded by any node and that it not c-command any node.[17]

Let us now return to the *order schema* given in (11). It turns out that this means of extracting the derived domination relation from c-command is not nearly so unusual in the context of syntactic theory as it seems as first blush.[18] Indeed, this schema has played a prominent role in previous syntactic theorizing, specifically in a definition of c-command assumed by Barker and Pullum (1990) among others, where D^{irr} is the irreflexive version of dominance, i.e., proper dominance:[19]

(16) $xCy \stackrel{\text{def}}{\equiv} \forall z[zD^{irr}x \rightarrow zD^{irr}y]$

"x c-commands y iff every node properly dominating x also properly dominates y."

It is immediately clear that this definition matches the form of our order schema exactly. Thus, we can restate Barker and Pullum's characterization of c-command as $\preceq^{D^{irr}}$. The fact that the transformation from dominance to c-command and back again is virtually identical is surprising indeed, and suggests a significant redundancy between these two relations.

Note that under the definition used by Barker and Pullum, and also by Jackendoff (1972) and Reinhart (1981), the c-command relation includes the domination relation (i.e., if x dominates y then x c-commands y). This instantiation of R in the order schema shows therefore that in general there can be overlap between the original relation R and the derived relation $\preceq^R$. However, under the most commonly assumed characterization of c-command (as given in (9)), we expect the domination relation (used to instantiate R in (16)) and the c-command relation (which we are attempting to derive from domination) to be non-overlapping relations. To capture the exclusion of some pairs, we can alter the order schema in the following way:

(17) $x \preceq^{R-} y \stackrel{\text{def}}{\equiv} \forall z[zRx \rightarrow zRy] \wedge \neg xRy$

"x is R^--ordered before y iff x is R-ordered before y and x is not R-related to y."

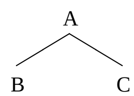
With this redefined schema, $\preceq^{D^{irr}}$ yields the notion of c-command used by Jackendoff, Reinhart, and Barker and Pullum, and $\preceq^{D^{irr}-}$ gives the traditional definition of c-command which does not include the domination relation.

Faced with these two variants of the order schema, we might now wonder whether derived dominance is better characterized by $\preceq^C$ or $\preceq^{C-}$. In fact, there are a number of considerations that push us in the direction of using the schema that excludes overlap between dominance and c-command, that is $\preceq^{C-}$. One of these concerns our ability to rule out ill-formed c-command relations. One simple way to do this, reminiscent of Kayne's (1994) LCA proposal, would require that the derived dominance relation behave like dominance, in the sense that it must satisfy the axioms usually imposed on primitive dominance as discussed in section 2. Consider a structure with 3 elements, say a, b and c, and the c-command relation $C = \{(a, b), (b, c)\}$. This c-command relation is illicit, since there is no tree which gives rise to exactly this c-command relation. Note, however, that if we apply the order schema $\preceq^R$ to this relation, we derive a dominance relation in (18).

(18) $\preceq^C$={(a,a), (b,b), (c,c), (a,b), (a,c)}

Taken together with an associated precedence relation $P = \{(b, c)\}$, this derived dominance relation is of a perfectly normal sort, as it satisfies all of the usual tree axioms and characterizes the following structure:

(19)



The problem here is that although the putative c-command relation C lacks the pair (c, b) but includes the pair (a, b), the derived relation $\preceq^C$ is the same as the one which would derived for the alternative c-command relation $C' = \{(b, c), (c, b)\}$, the one which we assume in fact characterizes the structure in (19). In fact, this problem is even more general as all of the following relations will give rise to the same derived relation when given as input to the order schema $\preceq^R$.

(20) a. $\{(a, b), (b, c)\}$

b. $\{(a, b), (b, c), (c, b)\}$

c. $\{(b, c), (c, b)\}$

d. $\{(a, c), (c, b)\}$

e. $\{(a, c), (b, c), (c, b)\}$

f. $\{(a, b), (a, c), (b, c), (c, b)\}$

Interestingly, this problem dissolves if we make use of the alternative order schema $\preceq^{R-}$ (from (17) and repeated here) that blocks overlap between R and the derived relation:

(21) $x \preceq^{R-} y \stackrel{\text{def}}{\equiv} \forall z[zRx \rightarrow zRy] \wedge \neg xRy$

Instead, only the relation in (20)c, the actual c-command relation for (19), gives rise to the derived dominance in relation in (18). Applying the alternative order schema to C yields instead the following derived dominance relation:

(22) $\preceq^{C-}$={(a,a), (b,b), (c,c), (a,c)}

This derived relation violates a number of properties usually attributed to dominance: there is no root and the nodes b and c violate exclusivity. Consequently, by adopting the alternative schema as the characterization of derived dominance, we can effectively rule out certain ill-formed c-command relations via constraints that we impose on the derived dominance relation.

Along a similar line, a number of properties of c-command can be deduced from constraints imposed on the derived dominance relation, so long as we adopt $\preceq^{C-}$ for this purpose rather than $\preceq^C$. For instance, by requiring $\preceq^{C-}$ to be reflexive, we can infer that C must be irreflexive. To prove this, suppose that it wasn't the case, that is, for some x,

$(x, x) \in C$. By the schema, it can't be the case that $x \preceq^{C-} x$, in contradiction to our assumption that $\preceq^{C-}$ meets the conditions of dominance, among which irreflexivity. Were we dealing with $\preceq^{C}$, this result would not follow, since the reflexivity of $\preceq^{C}$ could hold even for a reflexive c-command relation. The adoption of $\preceq^{C-}$ also allows us to derive the result that the existence of a c-command relation between two nodes implies the existence of a precedence relation between them. To see why, suppose that xCy. By the second part of the schema it can't be the case that $x \preceq^{C-} y$. Moreover, note that since xCy and not xCx, it can't be the case that $y \preceq^{C-} x$. Thus, there is no way for x and y to stand in the derived domination relation. Assuming again that $\preceq^{C-}$ satisfies the exclusivity condition, we can conclude that either xPy or yPx, as required. Clearly, this would not follow from $\preceq^{C}$, as it would not allow us to derive the lack of $x \preceq^{C-} y$ from xCy.

A further consequence of using the $\preceq^{R-}$ schema is that it allows us to distinguish between different conceptions of c-command. Recall that the traditional notion of c-command we are assuming, defined in (9) and repeated below, prevents x from c-commanding y if x stands in the domination relation with y.

(23) α c-commands β iff every node that properly dominates α also properly dominates β and neither dominates the other.

In contrast, the version of c-command assumed by Jackendoff (1972), Reinhart (1981) and Barker and Pullum (1990) makes no such prohibition:

(24) α c-commands β iff every node that properly dominates α also properly dominates β.

Under this definition, in fact, any two nodes that stand in a dominance relation will also stand in a c-command relation. Suppose we have a structure over which we have a traditional c-command relation C, and an alternative c-command relation C' that accords with the definition in (24). It turns out that for these two relations, $\preceq^{C}$ will always be identical to $\preceq^{C'}$. In contrast, if we assume that derived dominance is defined using the $\preceq^{R-}$ schema, the two types of c-command relations give rise to quite different results, with only those derived dominance relations produced from c-command relations of the traditional sort meeting the conditions exhibited by dominance.

We take these examples to suggest that the alternate schema is better suited for deriving dominance. Consequently, for the remainder of the paper, we assume that dominance is derived via $\preceq^{C-}$.

Before concluding this section, note that there are some properties of c-command can be deduced from either formulation of derived dominance. One example is the following property.

(25) **Inheritance**: $\forall x, y, z[xCy \wedge yDz \rightarrow xCz]$

"If x c-commands y, then x c-commands everything y dominates."

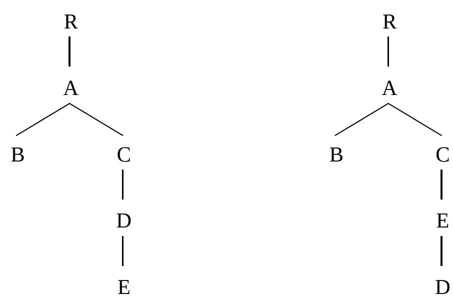
To prove inheritance, suppose that xCy and $y \preceq^{C-} z$, but not xCz. By the order schema, since xCy but not xCz, it cannot be the case that $y \preceq^{C-} z$, since there is something that c-commands y but not z, namely x. This yields a contradiction and the result follows.

5 Definability of Dominance

The results of the previous section rest heavily on the use of the possibility of defining dominance in terms of c-command. One might object at this point that if it were possible to use c-command to characterize exactly the same class of structures and express the same conditions that one can with dominance, we might be led to conclude that characterizations of trees in terms of c-command or dominance are essentially interdefinable variants. In this section, we will see that this is not the case.

Consider the following pair of structures, each with a sequence of 2 non-branching non-terminal nodes:

(26)



Clearly these two structures are distinguished by their dominance relation. In the structure on the left, D dominates E (and not vice versa), while in the structure on the right E dominates D (and not vice versa). Thus, in a dominance-based formalization, these constitute distinct representations, and therefore one would expect that the difference between them could prove grammatically relevant. The c-command relation, in contrast, does not distinguish between these phrase markers. That is, the two structures in (26) witness precisely the same set of c-command relations, namely:[20]

(27) $C = \{(b, c), (b, d), (b, e), (c, b)\}$

Unlike a formalization of phrase structure in terms of dominance, then, our c-command-based formalization does not distinguish between pairs of structures like those in (26), in which non-branching nodes are reversed in order. Our current perspective forces us to the conclusion, therefore, that the distinction between the two diagrams in (26) is merely artifactual, analogous to the obviously insignificant differences among phrase structure diagrams in the length of lines connecting nodes. Thus, we derive the result, correct we believe, that grammatical theory will never need to exploit the difference between the structures in (26) in order to characterize well-formedness.[21]

While this example raises the question whether grammatical theory will need to exploit differences between structures that are indistinguishable from the point of view of c-command relations, it also raises an interesting question of whether domination can be defined in structures specified in terms of c-command. (As precedence does not bear upon the result, we will put aside the issue whether it is available as a primitive relation.) In fact, we can show that it is not in general:

Theorem 1 *Domination is not definable from c-command (and precedence).*

Proof: Let N be the universe in a structure representing a set of nodes, and $C \subseteq N \times N$ represent the c-command relation. For C equal to the relation in (27), consider the mapping $f : N \to N$ which maps D to E, E to D, and maps every other node to itself. This mapping preserves c-command (and precedence) and hence is an automorphism. However, it does not preserve the domination relation (between D and E in either structure in (26)). As we have shown that there is an automorphism that preserves c-command (and precedence) but not domination, we have established that domination is not a definable property from c-command and precedence. ■

We can make some further observations on the basis of this example. The property of being a leaf node (i.e., a terminal node) is not definable either (as E is a leaf node but D is not in one but not the other of the structures in (26)). The property of being a root node is not definable either (shown by considering an automorphism for either of the structures in (26) which maps R to A, A to R, and maps every other node to itself). Note that the essential aspect of the proof of non-definability is that we have nodes such as D and E (and likewise R and A) that are indistinguishable by c-command, i.e., they are related to nodes in the structure by c-command in the same way. This is the reason why these mappings are automorphisms. [22]

It becomes interesting to consider whether domination (and properties of being a leaf node or root) can be defined when we consider subclasses of tree structures in which any pair of nodes participate in different c-command relations, i.e., nodes are distinguished by their c-command profile. The class of branching structures (i.e., where every non-root node has a sibling) is one class of structures that satisfy this additional requirement. In fact, we will now show that if we limit our attention to branching structures, the derived relation $\preceq^{C-}$ corresponds exactly to domination.[23]

Theorem 2 *In branching structures, domination is definable in terms of c-command and the domination relation is given by* $\preceq^{C-}$.

Proof: Consider a pair of distinct elements (nodes), A and B, in any branching structure. If A dominates B, then it follows that that for any node C, if C c-commands A then C would c-command B as well. Furthermore, it is not that case that A c-commands B. Thus, we have $\langle A, B\rangle \in \preceq^{C-}$.

Now suppose A does not dominate B. Then we have either that B properly dominates A or that this pair does not stand in the dominance relation at all. First, let us consider the case when A and B are not in the dominance relation. Then we must have a pair of sibling nodes, say C and D, which dominate A and B respectively. Then C witnesses $\langle A, B\rangle \notin \preceq^{C-}$. Now, let us consider the case where B properly dominates A, i.e., there is a child of B, say C, that dominates A. Since we are considering branching structures only, C must have a sibling, say D. Now D c-commands A but does not c-command B (as it is dominated by B). Thus, D witnesses $\langle A, B\rangle \notin \preceq^{C-}$. Thus, we have shown that if A does not dominate B then $\langle A, B\rangle \notin \preceq^{C-}$.

Therefore, we have established that domination relation and $\preceq^{C-}$ coincide in branching structures (where any pair of nodes can be distinguished on the basis of c-command relations). ■

Having shown that $\preceq^{C-}$ corresponds exactly to domination in all branching structures, we can now show that it is only in branching structures where $\preceq^{C-}$ is identical to domination.

Theorem 3 $\preceq^{C-}$ *coincides with the domination relation only in branching structures.*

Proof: Suppose we consider a structure which is not a branching structure. Then there must be a node A which has exactly one child, say B. Thus we have A dominates B but not vice versa. However, one can note that any node that c-commands B c-commands A as well and B does not c-command A. Thus, we would have $\langle B, A\rangle \in \preceq^{C-}$. Thus, in structures which are not branching structures, $\preceq^{C-}$ does not coincide with domination. ■

From these two results, we see that the class of branching structures exactly characterizes the contexts in which $\preceq^{C-}$ corresponds to dominance. This is the same class of structures that can be constructed in the derivational architecture of Chomsky (1995), where hierarchical structure reflects the merging of separate pieces of syntactic structure. The fact that these two systems characterize the same class of structures suggests that there may be a natural fit between them. Specifically, if we take applications of the merge operation to give rise to a c-command relation, an idea we explore further in section 7 (cf. also Epstein (1999) and Epstein et al. (1998)), theorem 2 ensures that we can unproblematically use $\preceq^{C-}$ to reconstruct the notion of constituency.

Clearly, the result that $\preceq^{C-}$ defines domination in all and only branching structures does not imply that domination is definable only in branching structures. Recall, our result on non-definability of domination crucially hinged on the fact that in certain forms of trees, there are pairs of nodes that are indistinguishable in terms of c-command (and hence allowing us to propose automorphisms that do not preserve domination relationships). Thus, to preserve domination between any pair of nodes, at the very least their participation in the c-command relation must be distinguishable, otherwise the c-command relation simply lacks sufficient information to reconstruct dominance. That is to say, a minimal condition for definability of dominance is that a c-command based structure must abide by the following condition:

(28) **Extensionality of** C: $\forall x, y[\forall z(xCz \leftrightarrow yCz) \wedge \forall w(wCx \leftrightarrow wCy) \rightarrow x = y]$
"If two nodes enter into the same set of c-command relations, they are identical."

Extensionality requires that nodes in a phrase marker are uniquely determined by their c-command profile.[24] As already observed, the class of branching trees are a subclass of the structures satisfying extensionality, and by Theorem 2, we know that dominance is in fact definable over these structures. However, non-branching structures such as the one given in (26) do not satisfy extensionality. This is because D and E (as well as the pair R and A) are indistinguishable by c-command. However, note C and D are distinguishable by c-command. These observations allow us to characterize the non-branching parts of structures that satisfy the extensionality condition. To give such a characterization, we first define the following notion of *node chain*:

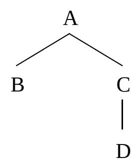
(29) We say nodes $A_0, A_1, \ldots A_n$ form a *node chain* of length n if A_i is the parent of A_{i+1} ($0 \leq i < n$) and A_j ($1 \leq j \leq n$) has no siblings.

Under this definition, a sequence of two nodes the higher of which is non-branching, such as the nodes R and A in (26), form a node chain of length 1, while a sequence of 3 such nodes, such as the nodes C, D and E in the same structures, form a node chain of length 2. We can then characterize branching structures as precisely those where there are no node chains of length greater than 0.

Let us now consider the nature of node chains in structures that satisfy the extensionality condition. Consider a node chain given by $A_0, \ldots A_n$, as specified in (29). As the A_j's ($1 \leq j \leq n$) can have no siblings, they will be indistinguishable by c-command. Hence by extensionality there can be at most one such A_j, and as a result the maximum length of a node chain is one. If A_0 has a sibling, then A_0 will c-command it but A_1 won't. Thus, we see that we can indeed have node chains of length one, as A_0 and A_1 are distinguished by their c-command profile. Such a node chain occurs in the following extensionality-satisfying structure:

(30) a. $C = \{(b, c), (b, d), (c, b)\}$

b.



In fact, such a node chain of length one can occur if and only if the first node in the node chain has a sibling. The only place where the top node of node chain would fail to have a sibling is at the root. Hence, at the top of a tree structure, we cannot even have a node chain of length one in structures satisfying extensionality: nodes R and A in the structures in (26) are in the same set of c-command relations, as neither c-commands nor is c-commanded by any other node.

It is straightforward to verify that in general $\langle A, B\rangle \in \preceq^{C-}$ iff A dominates B or A and B are in the same node-chain. Having noted that in structures satisfying the extensionality condition on c-command the maximum length of a node chain is one and that the root may only be part of a zero length node chain, we can derive domination in such structures in a straightforward and brute-force manner.

(31) A dominates B if and only if (a) $\langle A, B\rangle \in \preceq^{C-}$ and (b) if $A \neq B$ with $\langle A, B\rangle \in \preceq^{C-}$ as well as $\langle B, A\rangle \in \preceq^{C-}$ then there is a node that A c-commands but B does not.

Satisfaction of the antecedent in (b) implies A and B are in a node chain and the set of nodes c-commanding them are the same. In structures that satisfy extensionality, we have a node chain of length 1 and extensionality therefore requires them to be distinguished in terms of the nodes that are in c-command relation with them. The consequent in (b) simply says that if they are in a node chain, they are distinguished because the higher node in the node chain must have a sibling (notice that since it is a node chain of length one, A can not be the root of the tree) and hence it c-commands a node that the lower node in the node chain does not. Thus, we have shown that:

Theorem 4 *Domination is definable in exactly those structures that satisfy the extensionality condition on c-command.*

We take this result to provide a tantalizing connection between the primitive c-command hypothesis and tree structures, a class for which dominance is usually taken to be central in their characterization. The extensionality condition can be seen as representing the core intuition that we are exploring, namely that c-command is the basic relation out of which syntactic representations are built. If extensionality were to fail to hold of a primitive relation, it would mean that the primitive was not sufficiently expressive to express necessary structural differences. It is striking, then, that it is exactly over those structures for which c-command is extensional that dominance, and we take it tree structures, are definable in terms of c-command.[25]

We would like to suggest that the class of extensional structures is not simply of mathematical interest, but in fact corresponds to those which are required under Kayne's (1994) conception of phrase structure. Kayne proposes that precedence (among terminals) be derivable from the dominance relation in the following manner:

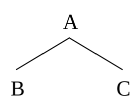
(32) A terminal element α precedes a terminal β iff some node which dominates α (perhaps α itself) asymmetrically c-commands β.

In order to ensure that the resulting relation be interpretable as a precedence relation, Kayne imposes a condition, what he calls the Linear Correspondence Axiom (LCA) the derived precedence relation.

(33) Linear Correspondence Axiom: The derived precedence relation must be a linear ordering on the terminals (i.e., it must be irreflexive, asymmetric, transitive and total).

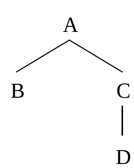
Kayne goes on to argue that the class of structures satisfying the LCA closely mirrors those permitted under constraints on phrase markers like X-bar theory. Consider for example what the LCA entails about the structure of complementation. In principle, the complement C to head B could either be a simple syntactic element with no internal structure, a head, or could possess internal structure, a phrase. The first of these possibilities is depicted in (34).

(34)



Applying the definition in (32) to this structure yields no order at all between the words dominated by B and C, as there is no node which asymmetrically c-commands any node dominating B or C. As a result, the resulting precedence relation violates the LCA as it is not total. In a structure with a complex complement, however, the situation is different:

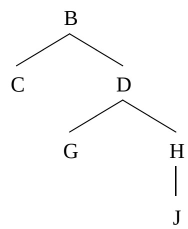
(35)



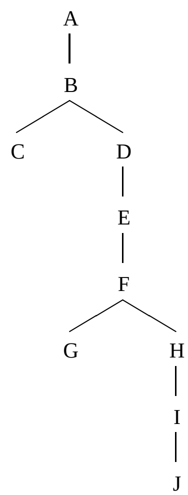
Here, Kayne's derived precedence relation satisfies the LCA: B precedes D since there is a node dominating B that asymmetrically c-commands D, namely B itself. Hence, the LCA derives the X-bar theoretic assumption that complements must be phrasal.

Observe that if Kayne's proposal is on the right track, syntactic representations must include certain instances of non-branching node chains, contra the proposals of Chomsky (1994) or Lasnik and Kupin (1977). We might ask, however, whether there is any restriction on the class of node chains that Kayne's LCA allows. Since Kayne's LCA talks only about the ordering relation among the terminal nodes, it is necessarily insensitive to the addition of structure in the form of non-trivial node chains so long as such structure does not alter the existence of asymmetric c-command relations among nodes that are already present. That is to say, any node chain of length one can be expanded to a node chain of arbitrary length without affecting the resulting precedence relation.[26] Thus, from the point of view of the LCA, the following pair of structures are essentially equivalent, as both give rise to an identical derived precedence relation.

(36) a.



b.

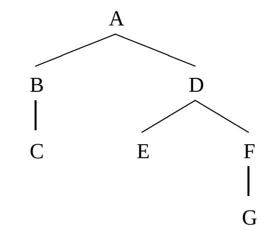


Assuming structures like (36)b to be linguistically anomalous, Kayne's assertion that the LCA derives, for example, X-bar theory seems incorrect. At the same time, the multiplication of the range of LCA-compatible structures through the process of expanding node chains strikes us as a sort of spurious ambiguity arising from an incorrect choice of representational primitives. Thus, a means of blocking structures like like (36)b would be welcome indeed.

In this connection, it is worth recalling that the class of structures satisfying the extensionality condition is limited to those containing node chains of length at most 1 (at most 0 at the root). The structure in (36)b is ruled out by extensionality, then, as the nodes E and F, or I and J are not distinguished by their c-command relations. This limitation on possible node chains does not however prevent their occurrence in contexts where the LCA requires them, in order to yield a well-formed derived precedence relation. Thus, although structures with longer node chains are tolerated by the LCA, one can show that any node chain in an LCA-compatible structure can be reduced to length 1 without affecting LCA-compatibility or the associated precedence relation.[27] Thus, the class of structures satisfying extensionality seems to exhaust the range of linguistically useful representations that are licensed by the LCA.

We have thus far focused only on head-complement configurations, and have omitted mention of specifier-head structures like the following:

(37)

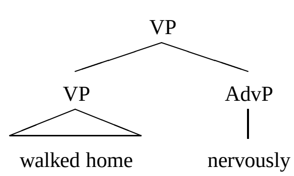


As Kayne (1994) notes, these structures are ruled out by the antisymmetry clause of the LCA: C should precede E, since B (which dominates C) asymmetrically c-commands E, while at the same time E should precede C, since D (which dominates E) asymmetrically c-commands C. Kayne's solution to this conundrum is to assume that specifiers stand in a structural relation with the phrase to which they attach distinct from those which we have considered. In particular, specifiers constitute a case of an adjunction structure. The addition of the possibility of adjunction introduces a host of questions about the nature of phrase markers, questions about which our our c-command-based proposal has much to say. In the next section, therefore, we explore the implications of primitive c-command for adjunction structures.[28]

6 Adjunction

The view of phrase structure we have been exploring retains the essential ideas underlying traditional syntactic representations. Yet scattered throughout the syntactic literature are proposals to modify this picture in a number of ways. One especially prominent modification concerns the phrase structure underlying adjunction. So-called Chomsky adjunction is assumed to differ from substitution in that it does not produce a constituent whose bar level is greater than that of the constituent that is projected. Instead, the target of adjunction is projected with no change in its bar level. Thus, in contrast to the substitution of a phrasal element, say NP, into the complement position of a V^0 head to produce a V' projection, the adjunction of an adverbial to a VP (=V'') produces a structure with iterated VP nodes:

(38)



If nothing further is said, one would expect these two instances of VP to behave just as any other pair of nodes where one is the projection of the other. However, May (1985) and Chomsky (1986) argue on quite distinct empirical grounds that the structure underlying adjunction differs in a fundamental way from that involved in substitution. Specifically, they argue that the element which is adjoined, the AdvP in the example above, has an intermediate status with respect to the question of whether it is within the projection of the element to which it is adjoined. On the one hand, both May and Chomsky take an adjoined element A to form a constituent with the element B to which it is adjoined. Yet they do not want this fact to interfere with the possibility of A entering into structural relations with elements outside of B's projection. As an illustration of the kinds of empirical arguments that have been put forward in favor of this conclusion, consider the following example discussed by May:[29]

(39) Who does everyone like?

The puzzle raised by this example concerns the ambiguity in scope between the wh-phrase *who* and the quantifier *everyone*. May suggests that prior to LF, quantifiers adjoin to some maximal projection containing them, yielding an LF like the following:

(40) $[_{\text{CP}}$ who$_i$ does $[_{\text{IP}}$ everyone$_j$ $[_{\text{IP}}$ t_j like t_i]]]

May argues that an element A can have scope over another one B if A commands B, i.e., every node that dominates A also dominates B. Since *everyone* is adjoined to IP, May suggests that IP does not dominate *everyone* in the relevant sense. Consequently, *everyone* is able to enter into a command relation with wh-phrase within the CP.[30]

To achieve this result, May suggests that the iterated nodes deriving from adjunction be treated as a certain type of unit. Thus, in addition to the usual nodes that constitute phrase markers, which Chomsky labels *segments*, May adds another type of object to the phrase structural ontology, the *category*. To clarify terminology, in the structure in (38) we say that there are two segments of a single VP category. In contrast, the AdvP category consists of a single segment. As before, segments (nodes) are assumed to stand in a dominance relation. Thus, the top segment of VP dominates the sole AdvP segment as well as the lower VP segment. In addition, Chomsky and May assume a separate relation of dominance by a category. This is defined (in terms of domination by segment) as follows:

(41) x category-dominates y iff every segment of x segment-dominates y.

The sense in which the AdvP in (38) is both within and without the VP should now be clear: it is dominated by one of the VP segments, but is not dominated by the VP category. If command is now defined using category-domination, the quantifier *everyone* will not be dominated by the IP, and will therefore be able to command the wh-element within CP.

Since the "classical" view of trees does not include a distinction between segments and categories, it needs to be extended in some way to handle this proposal. One way to do this involves extending the primitive structural ontology to include two sets of objects corresponding to segments and categories, and replacing the single dominance relation among nodes with two distinct dominance relations, one among segments and one among categories.[31] Of course, the set of segments must bear a specific relationship to the set of categories (e.g., the latter must determine an equivalence relation over the former), and the dominance relation over segments must bear a tight relationship to the dominance relation over categories. Thus, we will need to impose additional conditions so as to guarantee that these relationships are well-behaved. This approach strains a bit under the weight of Occam's razor: the representational machinery is dramatically expanded to accommodate adjunction structures.

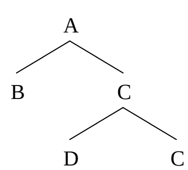
A less ontologically promiscuous approach to the segment/category dichotomy, and the one that May and Chomsky adopt, treats one of the types of phrase structural objects (and its associated dominance relation) as primitive, and defines the other in its terms. May and Chomsky take segments to correspond to the traditional nodes, and define a derivative notion of category as a set of segments. Domination by a segment is taken to be the standard dominance relation among nodes, while domination by a category is defined in terms of this primitive relation, as in (41).[32] This view of structure, though less expansive in its ontological commitments than the previous suggestion, does pose certain puzzles, similar in spirit to those discussed in section 3. First of all, it seems to be the case that syntactic processes manipulate only categories and not segments; that is, it is a category and not a segment that undergoes syntactic movement or serves as a landing site of such movement. If segments are the primitive objects, one wonders why it is not manipulated by the syntax? Further, from an interpretive point of view, it is the category rather than its constituent segments which is assigned a thematic roles. To repeat the by now familiar refrain, if segments are the primitive object of our representation, why is it apparently invisible for the process of interpretation?

Assuming the importance of a structural difference between adjunction and substitution, we might wonder whether there is any way to represent this difference without resort to segments and categories. In fact, we believe that there is and it depends crucially on the idea of a primitive c-command relation. As mentioned above in our discussion of (40), May's account of scopal relations rests crucially on the assumption that c-command is sensitive to the notion of category-dominance. It is in this way that the adjunction of the quantifier has its distinctive effects. Let us consider in further detail how c-command is defined once the notion of structures is extended to allow for the possibility of adjunction. The following definitions come from Kayne (1994):

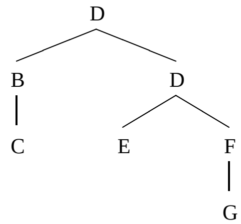
(42) a. α c-commands β iff every category that *properly* dominates α *properly* dominates β and α excludes β.[33]
b. α excludes β iff no segment of α dominates β.

For pure tree structures, this relation among categories is identical to the c-command relation among nodes that we have been considering thus far. However, in the context of adjunction structures, there is an interesting structural asymmetry between the the adjoined element and the category to which it has adjoined. In the structure in (43), the adjoined element D as well as the category C to which it is adjoined both c-command B: the only category dominating either of them is A, which in turn dominates B. In addition, D c-commands the category C: the only category dominating D, namely A, also dominates C, and D excludes C. Yet, the converse relation does not hold, that is, C does not c-command D: although A dominates both C and D as before, C does not exclude D.

(43)



This asymmetry in c-command between the adjunct and the target of adjunction provides a elegant solution to the problem mentioned at the conclusion of the previous section concerning specifiers and the LCA. Specifically, Kayne (1994) suggests that (phrasal) specifiers are adjoined to their hosts to yield structures like the following:

(44) 

Unlike the problematic structure in (37), this structure satisfies the LCA: terminal C within the specifier precedes the head E since B asymmetrically c-commands E. E no longer (improperly) precedes C, since the adjunction configuration prevents D from (asymmetrically) c-commanding C.

Beyond this benefit for Kayne's proposal, the asymmetry of c-command in adjunction suggests a novel way of characterizing such structures directly without resorting to the machinery of segments and categories.[34] Roughly, when we have two sister nodes which are related through unidirectional c-command, we understand this as an instance of adjunction. Under this conception, then, there is nothing at all unusual about "adjunction structures". They are simply those in which there is a unidirectional, as opposed to bidirectional, c-command relation between certain nodes. In the next section, we will turn to making this idea more precise.

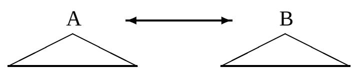
7 C-Command and Grammatical Derivations

Up to now, we have adopted a representational view of grammar, where well-formedness derive from constraints that apply to completely specified. In the remainder of the paper, we will explore the implications of primitive c-command for a derivational model, where well-formedness derives, at least in part, from properties of the derivational operations. Such a derivational view will allow us to introduce in an especially simply way the conception of adjunction we suggested in the previous section.

Let us assume a model of grammar in which the phrase marker is composed from smaller pieces during the course of a derivation, generalized transformations in the terminology of Chomsky (1955) (see also Chomsky (1995), Frank and Kroch (1995), Epstein et al. (1998)). We follow Chomsky (1995) in supposing that there are two combinatory operations that can be used to compose sentential material, namely substitution and adjunction. Chomsky defines these operations in terms of their effects on the set theoretic objects that he assumes constitute syntactic representations, what he calls *terms*. Specifically, two terms α and β can be combined via substitution to produce a third term $\gamma = \{\mathrm{H}(\beta), \{\alpha, \beta\}\}$, where the singleton element of this term, $\mathrm{H}(\beta)$, called the *label* of the term, represents the fact that it is (the head of) β that projects its features to γ. Adjunction of a term α to a term β behaves in an almost identical fashion, to produce the term $\{\langle H(\beta), H(\beta)\rangle, \{\alpha, \beta\}\}$. The only novel property of this adjunction-derived term, as compared to the substitution-derived term seen above, is in the form of the label: in this case it is an ordered pair, while above it is simply a bundle of features. This difference is intended merely as a minimal variation in notation so as to represent the difference between adjunction and substitution. As nothing about the set-theoretic structure of terms would have led us to expect the difference between substitution-derived and adjunction-derived terms, the existence of these modes of combination is essentially stipulated.

Thinking in terms of c-command based structures, however, the situation is quite different. Suppose we have two structures α and β, with roots A and B respectively. Assuming that the grammar may manipulate only the primitive c-command relation, there are exactly three ways in which α and β may be combined (at the root): through the assertion of mutual c-command between A and B, through one way c-command from A to B, or through one way c-command from B to A.[35] The first of these, mutual c-command, we will take to be substitution:

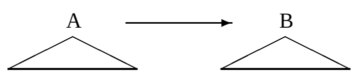
(45)



The conflation of mutual c-command with substitution should strike the reader as unsurprising: heads and complements, the canonical instance of substitution, stand in precisely this relation.

The other combinatory possibilities, where there is only one-way c-command between A and B, we take to be cases of adjunction. In particular, when A c-commands B but not conversely, this operation corresponds to the adjunction of A to B (and conversely for the other case).

(46)



In the context of our discussion of structure (43), the reason that we identify this type of combination with adjunction should be clear. It is precisely in the context of adjunction structures that this c-command asymmetry arises, given the version of c-command derived from dominance using the definitions in (42). However, instead of taking the asymmetry to derive from the configuration of segments and categories in adjunction structures, we take it as definitional of an adjunction structure.

These intuitions can be formalized as follows: Let S be a structure whose node set is N, whose c-command relation is C, and whose root is $R(S)$ (as determined according to the rootedness condition). Similarly, let S' be a structure with node set N' (disjoint from N), c-command relation C', and root $R(S')$.[36]

(47) a. Merging S and S' via substitution yields the smallest structure satisfying the structural well-formedness conditions where:

$$\begin{aligned} N_{\mathrm{SUBST}(S,S')} &\supseteq N \cup N' \\ C_{\mathrm{SUBST}(S,S')} &\supseteq C \cup C' \cup \{(R(S), R(S')), (R(S'), R(S))\} \end{aligned}$$

b. Adjunction of S to S' is the smallest structure satisfying the structural well-formedness conditions where:

$$\begin{aligned} N_{ADJ(S,S')} &\supseteq N \cup N' \\ C_{ADJ(S,S')} &\supseteq C \cup C' \cup \{(R(S), R(S'))\} \end{aligned}$$

Seen from the perspective of a c-command-based view of structure, the existence of these distinct modes of combination is unsurprising, as substitution and adjunction virtually exhaust the range of logically possible combinatory operations.[37] Indeed, the fact that natural language seems to exploit all of the possible ways that our formal machinery makes available for structural combination suggests that our choice of formal primitives is correct. Furthermore, the specific nature of the distinction between substitution and adjunction need no longer be stipulated using the machinery of segments and categories or ordered pair labeled terms, as it follows instead from the primitive representational tools that the grammar provides.

7.1 Rootedness and Constituency Revisited

From the very simple characterization of the combinatorial operations in (47), we would like to be able to derive the entire set of properties that these operations are usually taken to have. First of all, in addition to the bidirectional or unidirectional c-command relations that are asserted by the definitions in (47), we also ought to be able derive other c-command relations that obtain after structural combination takes place. Specifically, we must explain why when the root of S c-commands the root of S', whether as a result of substitution or adjunction, it also c-commands every node within S'. Consider a situation in which r and r' are the roots of two structures that are combined, and where a is a node dominated by r'. By assumption, r c-commands r' in the resulting structure. If we assume that subconstituency that is established at some point in a derivation is preserved throughout, it follows that r must c-command a: if it did not, then r' would no longer dominate a.[38]

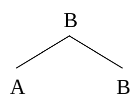
Perhaps the most significant difference between adjunction and substitution lies in the fact that the latter, but not the former, requires that a new instance of a node (category) be projected upon combination (but see Epstein (1998) for a different view). Given the view of adjunction and substitution we have just sketched, this difference can be shown to follow directly from the rootedness condition in (15) above. Consider the case of substitution. Here, the former roots of the structures that are combined will both be c-commanded by some other node, namely one another. Consequently, the smallest structure that includes all of the c-command relations specified in (47)a must add at least one node to those in N and N', so as to satisfy the rootedness condition. The case of adjunction works differently. Following the application of adjunction, one of the former roots will still satisfy the rootedness condition in (15): the asymmetric character of the adjunction operation will have the effect of leaving the root of the adjoined element not c-commanded by any other node. Hence, we correctly deduce that the adjunction operation, as defined in (47)b, requires the introduction of no new nodes.[39]

The discussion in the previous paragraph addressed the issue of when a new node needs to posited, but did not address the issue of the categorial identity or label of such nodes. Instead of viewing projection as a well-formedness requirement on structures, we can view it as the process of assigning an identity to a node at its point of insertion into the structure. In the case of substitution, we need only say that the inserted node must be the projection of one of the roots of the combined structures (perhaps for reasons similar to those adduced by Chomsky (1995, ch.4). The substitution operation itself, as defined in (47)a, is symmetrical with respect to the two objects that are combined, and thus either one may in principle project. The case of adjunction is rather different as the operation defined in (47)b is asymmetrical. Moreover, as we are dealing with a structure in which no new node (category) is projected, we should expect that the categorial identity of the combination should be determined by the label of the target of adjunction. Though this would be the ideal resolution, there remains the problem of identifying which node in an adjunction structure is the root. Under our characterization of root in the rootedness condition in (15), we should conclude that it is the root of the adjoined element that serves as the root following an instance of adjunction. Unfortunately, this is contrary to fact, as it appears to be the root of the target of adjunction that determines the distribution (and hence the label) of the combined structure.

A related problem concerns the issue of (derived) dominance and constituency. We have already explored these questions for structures arising from substitution, and have seen how dominance and constituency can be faithfully reconstructed using the relation $\preceq^{C-}$ derived from the order schema. Consider the following simple adjunction structure:

(48) a. $C = \{(a, b)\}$

b.



What should the derived dominance relation on such a structure be? Since we have eliminated reference to segments and are taking categories to be atomic entities, the standard interpretation of category domination in (41) dictates that neither A nor B (qua categories) dominates the other. Applying the definition of $\preceq^{C-}$, we see that this is exactly the correct result: neither A $\preceq^{C-}$B nor vice versa. A$\preceq^{C-}$B does not obtain in (48) because A c-commands B, while B$\preceq^{C-}$A fails because A does not c-command itself.[40]

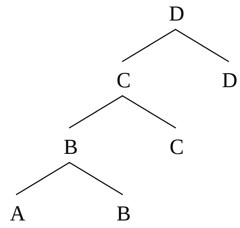
Now consider the problem of the root's identity in adjunction structures. In particular, suppose that we adopt the definition of root standardly assumed when dominance is primitive, that is, the least node in the ordering provided by dominance. When stated in terms of derived dominance, this is as follows:

(49) A node n is a root of structure S iff there is no node $m \in S$ such that $m \preceq^{C-} n$.

In the context of structures built using substitution alone, this definition yields the same results as those obtained from the characterization of root provided by the single root condition. In adjunction structures, however, the results diverge. Take for example the structure in (48). Here, the $\preceq^{C-}$ relation does not determine a unique minimum. Neither A$\preceq^{C-}$B nor B$\preceq^{C-}$A, so both are minima. Consequently, both A and B count as roots according to this characterization. In cases of iterated adjunction, this can yield an unbounded number of minima as shown in the following structure:

(50) a. $C = \{(a, b), (a, c), (a, d), (b, c), (b, d), (c, d)\}$

b.



Here, A, B, C and D are all minima according to $\preceq^{C-}$, as none dominates (in the sense of $\preceq^{C-}$) any of the others. *There are however two minima which may be distinguished, and which have important grammatical roles.* The first is the node that is not c-commanded, i.e., A. This node, which was picked out by our previous notion of root, does have one property that we ascribe to roots, namely it is not embedded (in the sense of Epstein (1999). Further, this node serves as the only possible locus for further adjunction (Kayne, 1994). The other distinguished minimum is the node which c-commands nothing, i.e., D. It is this node which determines the categorial label of the adjunction structure. We believe that this split of properties among the two minima suggests that the notion of root actually conflates a number of distinct concepts that are separated only in adjunction structures.[41] At present, however, we do not have any explanation for why these properties should disperse among these two distinguished minima as they appear to.

The presence of multiple $\preceq^{C-}$-minima in adjunction structures also allows us to derive the presence of certain c-command relations under the adjunction operation. In particular, consider the case in which a simple adjunction structure like that in (48) is combined using substitution with a structure with unique minimum C. The desired result is that both A and B should c-command C and everything within it: for example, a head A moved from within C should be able to c-command its trace. So far nothing we have said will produce this result. However, in the context of a conception of root as an $\preceq^{C-}$-minimum, it makes sense to reformulate the operations of substitution and adjunction in (47) so as to specify c-command relations among the minima of the structures according to the $\preceq^{C-}$ ordering, even when there are multiple such minima. Thus, substitution will posit bidirectional c-command between all the minima of one structure and all the minima of the other, while adjunction will posit unidirectional c-command among these minima. In essence, this builds certain aspects of transitivity into the combinatorial operations and avoids the necessity of invoking fairly complex transitivity conditions that would otherwise be necessary if we were take, say, B as the unique root of the adjunction structure. In the case in question, this will have the effect that both A and B will c-command C (and vice versa).[42] C-command relations between A and nodes within C will follow from inheritance as before.

Returning again to the question of constituency in adjunction structures, we can now construct a notion of constituent as in (13) based on $\preceq^{C-}$ (essentially category dominance). Doing this, we get the result that A and B in (48) form separate units, i.e., *constituent*$(A) = \{A\}$ and *constituent*$(B) = \{B\}$, and that there is no single constituent that contains them both. This result is unsatisfying in that there is a clear sense in which we would like A and B to form a unit. (Note that this result is obtained because of the use of this definition of constituent in terms of category dominance, regardless of whether category dominance were derived from c-command or not.)

One way out of this dilemma is to exploit an alternative hierarchical relation to (derived) category dominance in defining constituency. Indeed, we can get A and B to form a constituent if we make the notion of constituent sensitive to a hierarchical relation similar though not identical to dominance, namely inclusion, a relation defined as follows:

(51) x includes y iff x does not exclude y.

Recalling the definition of exclusion from (42)b, we see that a category x includes y when some segment of x dominates y. B includes A in (48), then, because the upper segment of B dominates A. Altering the definition of constituent in (13) to make reference to inclusion yields the result that *constituent*$(B) = \{A, B\}$, while *constituent*$(A) = \{A\}$. This is a plausible assignment of constituency to this structure, since, as noted earlier, the lower segment of an adjunction structure, excluding the adjoined element, does not seem to engage in any syntactic or interpretive processes. We cannot immediately incorporate this new conception of constituency into our primitive c-command proposal, however: the definition of exclusion in (42)b, the relation crucial in defining inclusion, makes reference to segments, objects we have excluded from our structural ontology. It is however possible to define a relation that seems to match inclusion closely, without recourse to segments and categories, making use instead of the c-command relation over categories and our order schema. This is done as follows:

(52) α includes β iff $\forall\gamma[\gamma D^{irr}\alpha \rightarrow \gamma D^{irr}\beta] \wedge \neg\alpha C\beta$.

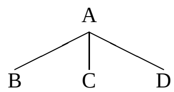
We take D^{irr} to be irreflexive (i.e., proper) category dominance. As we have just seen, category dominance can be defined as $\preceq^{C-}$, so (52) provides a definition of inclusion in terms of c-command. Since the definition of inclusion in (52) has essentially the form of an instance of our order schema, we can restate it in the following form (where D^{irr} is the irreflexive version of categorial dominance, D, that is given by $\preceq^{C-}$):

(53) α includes β iff $\alpha \preceq^{D^{irr}} \beta \wedge \neg\alpha C\beta$.

What is going on here is that we are defining a relation essentially analogous to Barker and Pullum's (1990) definition of c-command, a hierarchical relation which includes both the usual dominance and c-command relations, and removing the usual c-command relation. This turns out to leave exactly the portion of the dominance relation that is appropriate for defining constituency in adjunction structures. Thus, in (48), although neither A nor B dominates the other, only B includes A and not vice versa since A c-commands B.

Frank, Hagstrom, and Vijay-Shanker (1999) suggest an alternative conception of constituency in a primitive c-command model that avoids the complications of inclusion. They adopt a view of constituents that preserves the conceptual idea behind the view taken in dominance-based trees, where the constituent determined by a node is set of nodes it is less than in the dominance ordering, but depart from the details of this proposal by using c-command rather than dominance as the relevant ordering. Roughly, they propose that constituents are the maximal set of nodes c-command by some particular node, which they call the governor. This simple idea is insufficient for non-binary branching structures like the following, since here, the B should be able to function as the governor for the non-constituent consisting of C and D.

(54)

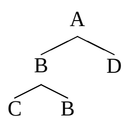


To avoid this problem, Frank et al. incorporate a notion of root of a constituent, a node which mutually c-commands the governor, and require that the root c-command no other node within the constituent, the latter condition being violated in the putative constituent {B,C} in (54), since a root B would c-command C (or conversely). This yields the following definition:

(55) Constituent(r)= $\{n | gCn \text{ and } \neg rCn\}$ for a node g such that rCg and gCr.

In the case of a tree like that in (56), this revised notion of constituent derives the desired result that the complex {B,C} together form a constituent with root B and governor D.

(56)



Space prevents us from exploring the numerous other consequences that follow from this revised notion of constituent, for example involving the structures of multiple adjunction. See Frank, Hagstrom, and Vijay-Shanker (1999) for further details.

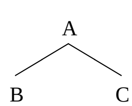
7.2 The Extension Requirement and Derivational Monotonicity

Chomsky (1993) argues that substitution differs from adjunction in an additional respect. He suggests that, as we have been assuming thus far, substitution can take place only at the root of a structure, as in (45), and cannot combine one structure with another at an internal node. We will call this strict cyclicity condition the extension requirement, as it demands that substitution must "extend" the structures being combined.[43] Adjunction need not obey the extension requirement, however, according to Chomsky. There are a number of arguments for this relaxation, the most straightforward stemming from head movement.[44] If movement must be to some c-commanding head, it necessarily will not target the root of the structure (but see Bobaljik and Brown (1997)).

Why should adjunction and substitution differ in this fashion? Under the characterization of these operations as combinations of set-theoretic terms, there is no easily available reason (but see Kitahara (1995) for an explanation in terms of economy conditions). However, the c-command based view of structure allows us to understand the reason for the existence of the extension condition and why it should apply only to substitution. Suppose we have the the following structure:

(57) a. $C = \{(b, c), (c, b)\}$

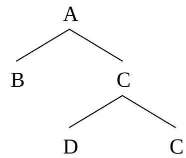
b.



If we attempt to adjoin a structure consisting of a single node D at C, we get the following:

(58) a. $C' = \{(b,c),(b,d),(c,b),(d,c)\}$

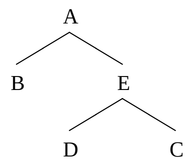
b.



If instead, we were to perform this combination by substitution, the desired result would be as follows:

(59) a. $C'' = \{(b,c),(b,d),(b,e),(c,d),(d,c),(e,b)\}$

b.



The relation between C and C', the adjunction-derived structure, differs from the relation between C and C'', the substitution-derived structure, in one crucial respect: C is a subset of C' while C is not a subset of C''. Specifically, C and C' both include the pair (c,b), while C'' does not.

In the context of a natural assumption concerning the character of grammatical derivations, this observation forms the foundation of our derivation of the extension requirement. This assumption takes the following form:

(60) Derivational Monotonicity: Syntactic derivations must monotonically increase the set of c-command relations.

Derivational monotonicity is conceptually related to the widely assumed constraint of the recoverability of deletion. If structural relations, as determined by c-command, could be retracted, their well-known import in interpretation would be lost.[45] Note also that derivational monotonicity can be seen as the driving force behind the copy theory of movement. The structure that is moved from a certain position cannot be literally deleted during the derivation, as c-command relations between nodes outside of the moved element and nodes within it would be severed. For these reasons, let us assume that derivational monotonicity holds.

Let us return now to the issue of the substitution/adjunction contrast. As we just noted, substitution of one structure to another at an internal node forces the retraction of a c-command relation, while adjunction does not. Stated in another way, internal adjunction satisfies derivational monotonicity, while internal substitution does not. In contrast, both substitution and adjunction, when applied cyclically at the root node, satisfy derivational monotonicity. The reason adjunction satisfies this is as before. Cyclic substitution satisfies derivational monotonicity because there is no c-command relations to be disturbed: in the non-cyclic case, the targeted node, C in our example, was lowered so that it no longer c-commands its original sister, B in (57). When substitution applies at the root, there is no such sister, and hence no c-command relation to retract.[46]

We conclude therefore that the extension requirement is a simply a reflex of the derivational monotonicity, and has no independent status. This is a welcome result, since there seemed to be little independent conceptual motivation for the extension requirement with its substitution/adjunction contrast. In contrast, derivational monotonicity is firmly conceptually grounded. We can make two further comments at this point. The first is the condition of derivational monotonicity offers us a new way of conceptualizing adjunction, in addition to that discussed above. Specifically, adjunction can now be seen as the combinatory operation that maximally preserves surrounding c-command relations, rather than as some exotic operation that introduces multiple segments or adds ordered pair-labeled terms. As we have seen, this has as a consequence that it is permitted to apply non-cyclically. Our second comment concerns the role of the c-command-based view of structure in the derivation of the extension requirement from derivational monotonicity. Note that if syntactic structure were characterized in terms of dominance and precedence, the difference between substitution and adjunction we are crucially exploiting collapses. All of the dominance and precedence relations in (57) continue to hold in both (58) and (59). Thus, a version of derivational monotonicity that constrained the dominance and precedence relations would leave the extension requirement unexplained.[47]

8 Conclusion

In this paper we have argued that c-command should be taken as the structural primitive in place of the usually assumed relation of dominance. To review, we have shown how this assumption substantially constrains possible syntactic structures in what we believe is a linguistically natural way. In section 4, we introduced a general schema for deriving a notion of hierarchy from an arbitrary relation, and showed in section 5 that applying this schema to c-command yields a relation that is identical to dominance in precisely the class of branching tree structures. We went on to prove that although dominance is not generally definable from c-command, the class of structures in which it can be defined corresponds precisely to those in which c-command is extensional, that is, where the each node is distinguishable from every other by its c-command profile. We then argued that this class of extensional structures has an interesting linguistic characterization as well: it is just those structures that are needed under Kayne's (1994) LCA-based conception of phrase structure.

When we turned our attention to adjunction in section 6, we found that the c-command-based view of structure provides us with a radically simplified conception of adjunction structures, allowing us to dispense entirely with segment-category distinctions, while retaining a structural difference between substitution and adjunction. In section 7, we in fact showed how the structural difference between substitution and adjunction is an expected consequence of a derivational system that manipulates primitive c-command relations, as these two operations correspond to the only two natural ways of combining pieces of phrase structure. A number of consequences arise from this conception. For instance, we deduce why a new node (category) must be projected in substitution though not in adjunction. Further, we see why, in the case of iterated adjunction, we obtain several roots or nodes at the top level in the sense that they all may c-command outside of the category to which they are adjoined. Of these roots, we noted that two can be distinguished, and observed that these two play important and distinctive grammatical roles, concerning determination of categorial label and locus of further attachment in a cyclic derivation. This observation implies that the traditional idea that structures possess a unique root conflates a number of distinct properties that are separated only in adjunction structures. Finally, we considered the distinction between adjunction and substitution as concerning derivational cyclicity, i.e., that the extension condition applies only to substitution. We observe that this stipulated difference has no independent basis and is simply a reflex of a more general requirement of derivational monotonicity.

In related work, we have pursued a c-command-based view of phrase structure in the context of the derivational system of the Tree Adjoining Grammar (TAG) formalism (Kulick, Frank, and Vijay-Shanker, 2000; Frank, Kulick, and Vijay-Shanker, 2000). There, we demonstrate that characterizing the structure of elementary trees and the effects of the TAG adjoining operation in terms of c-command resolves certain well-known syntactic puzzles for this formalism, and also allows a number of previously stipulated properties of the adjoining operation to derive naturally from the same monotonicity condition that we formulated in section 7.2.

It must be stressed that this investigation is preliminary and we do not claim to have resolved the issues that we have raised. Our current work is continuing to refine the notion of constituency with extensions to movement (Frank, Hagstrom, and Vijay-Shanker, 1999), studying possible axiomatizations of c-command based structures (obtained using the operations of adjunction and substitution), and considering the relation between such an axiomatization and Kayne's derivation of precedence from c-command. We believe, however, that we have already demonstrated some important advantages of conceptualizing phrase structure in terms of c-command. Thus, it seems to us that this path merits continued formal and empirical investigation.

Notes

[*]Thanks to the following for stimulating discussion: Paolo Acquaviva, John Chen, Paul Hagstrom, Richard Kayne, Tony Kroch, Seth Kulick, Fero Kuminiak, Peter Ludlow, Mike Parker, Paul Portner, Charles Reiss, Jim Rogers, Paul Smolensky, Juan Uriagereka, and Scott Weinstein. We have also received a wealth of useful commentary on this work both in written form from our anonymous reviewers and in oral form during presentations at CUNY, Universidade Estadual de Campinas, Universidade Federal de Santa Caterina, Università di Padova, University of Delaware, University of Edinburgh, University of Maryland, University of Pennsylvania and at the 18th GLOW Colloquium at the University of Tromsø. Finally, we gratefully acknowledge the financial support of NSF grants SBR-97-10247 and SBR-97-10411.

[1]In this respect, our investigations differ from those of previous works that have formally investigated c-command relations and their role in syntactic structure (Barker and Pullum, 1990; Kracht, 1993; Grefe and Kracht, 1996). Rather than attempting to formalize the usual notion of c-command defined on top of a standard domination-based formalization of tree structure, our goal is to study how taking c-command alone as a hierarchical primitive restricts the class of definable structures.

[2]Trees can be depicted using a variety of equivalent notations, for example as graphical tree notations or as bracketed strings. Such differences in notation, however, do not reflect any difference in the underlying formal structure that is the focus of discussion here.

[3]Recall that a relation R is *asymmetric* iff for all x, y, it is not the case that both xRy and yRx hold. A relation is

antisymmetric iff for all x, y, if both xRy and yRx hold, then $x = y$.

[4]We will not present arguments here for the reflexivity of dominance. We simply carry this assumption over from Partee et al.'s definition. It turns out that dominance may safely be assumed to be either reflexive or irreflexive, either choice having consequences for later definitions.

[5]In fact, the set of axioms given here is not a complete characterization of the set of tree structures so much as a definition of a class of models: There are structures that are consistent with these constraints that are not trees. Backofen, Rogers, and Vijay-Shanker (1995) show that a first-order axiomatization of the set of finite tree structures is impossible, but do provide a first-order axiomatization of the theory of finite trees, that is, a set of axioms that characterizes all of the properties that hold in finite trees. The differences between Backofen et al.'s axiomatization and that of Partee, ter Meulen, and Wall (1993) are not especially pertinent to our current concerns, so we do not introduce the additional complications entailed by the complete axiomatization.

[6]See the discussion of adjunction structures in section 6, however, for a case in which dominance and precedence may in fact be too weak.

[7]Even if definable, we might nonetheless understand a contrast in importance of two relations from the relative complexity of their definitions in terms of some set of primitives.

[8]We should point out here that our argument about constraining linguistic variation does not have roots in learnability issues, as the requisite stipulated substantive restrictions on possible structures or relations may nonetheless form part of our innate language endowment. Instead, our point stems from considerations of explanatory power and of language design. Concerning the first of these, we believe that ceteris paribus it is preferable if grammatical phenomena, i.e., perceived patterns of language date, can be explained through reference to properties of the formal basis of the theory, as this is likely to provide more general grounding for such an explanation than one rooted in stipulated conditions. Of course, this is an issue which must be decided on a case by case basis: if such derivation requires substantial complication of the underlying formalization, it might suggest that the explanation of a phenomenon more properly resides in a parochial condition.

Turning to the question of language design, it seems to us plausible that the structure of a grammatical theory might to a certain degree be derivable from the formal structure out of which the theory is built, minimizing additional constraints on it so much as possible. One sees similar considerations arising in the construction of theories of physical phenomena. Thus, the equations of general relativity arise virtually necessarily out of the structure of manifold theory under the assumption that no reference frame has a privileged status (i.e., the theory should be invariant under general coordinate transformations). These formal assumptions led to a number of astonishing and subsequently verified predictions, for example the existence of black holes.

[9]As Richard Kayne (p.c.) points out to us, his proposal is not explicitly stated in this way. Rather, he suggests that precedence and hierarchy are mutually constraining, and argues that the necessity for asymmetric configurational relations derives from the asymmetry of ordering. However, since under his proposal precedence is completely determined

by hierarchy (and not vice versa), it is natural to interpret the proposal in the way we have done in the text.

[10]In this work, we remain agnostic on the question of whether primitive precedence should be eliminated in turn in favor of c-command as well, following Kayne. Either resolution of this question is compatible with what follows.

[11]It is beyond the scope the current work to provide an explanation for why it is c-command as opposed to some other relation that is primitive. It is important to note, however, that the results of our investigation are not obviously incompatible with attempts to derive c-command from something more basic, perhaps along the lines of suggestions by Chametzky (1996), Epstein (1999), and Epstein et al. (1998). If c-command is derived from something more basic, though, it is important that the dominance relation not also be similarly derivable in general, as that would sacrifice the limitations on definability we observe in section 5.

[12]For convenience, in this paper, we will use the same symbols for the nodes and the labels in tree diagrams, using lower case for node names and upper-case for node labels.

[13]As our current exercise is to investigate whether the primitive of domination can be replaced by c-command, we do not explicitly consider the precedence relation. Therefore, for now, we remain agnostic to whether this relation must be taken to be a primitive.

[14]By proper dominance we mean the irreflexive version of the dominance relation.

[15]Note that if such a translation from c-command to dominance were possible generally, it might suggest that dominance-based and c-command-based views of structure are equivalent in some sense. We return to this issue in section 5 below.

[16]Observe that the root has an additional distinguishing property with respect to c-command, namely that it is a node that c-commands nothing. Thus, we might consider the additional rootedness condition:

(i) **Rootedness condition II**: $\exists r \forall y[\neg rCy]$
"There is a node r which is c-commands no other node."

Unlike the condition in (15), this property does not necessarily uniquely characterize roots for the class of structures we will be considering in this paper, and therefore we will not consider replacing (15) by (i). In section 6, we will explicitly take up the possibility that the roots as defined by these two conditions are distinct.

[17]When we consider adjunction structures in section 6, the divergence between these two properties will become clear.

[18]Nor is this schema unusual in the context of other formal theories. The definition of subset is identical to the order schema applied to the member-of relation, i.e., $\subseteq$ is identical to $\preceq^{\in}$.

[19]As Barker and Pullum note, it is crucial to use the D^{irr} relation as opposed to reflexive dominance in defining c-command, as this definition otherwise yields a relation identical to dominance.

[20]We crucially assume here the the primitive c-command is interpreted in the usual fashion, that is as blocking c-

command between nodes that would stand in a dominance relation (what we referred to as $\preceq^{D^{irr}-}$ at the conclusion of the previous section).

[21]This non-definability of non-branching structure recalls the proposals of Lasnik and Kupin (1977) and Kupin (1978) who derive the non-distinctness of structures like those in (26) from their monostring formalism. We leave open the relationship between their proposal and ours. See Richardson and Chametzky (1985) and Chametzky (1996) for a possible connection.

[22]Alternate definitions of c-command might not allow us to obtain this result. In the definition of c-command used by Barker and Pullum (1990), for instance, in which any pair of nodes in the domination relation are related by c-command as well, every pair of nodes is distinguished by c-command. In the definition of c-command involving branching nodes (Reinhart, 1976), again we have the situation that D and E are indistinguishable in terms of c-command. Under this definition, then, we can also arrive at the conclusion then that domination is not definable from c-command.

[23]The same holds for $\preceq^{C}$, since as we will see below it produces results identical to $\preceq^{C-}$for the class of tree structures when the input relation is c-command.

[24]In the context of the classical view of trees, no analogous extensionality requirement is typically asserted for the dominance relation, even though it holds for dominance as well. The reason for this is that there can be no pair of distinct nodes that enter into an identical set of dominance relations, since this situation is blocked by by the assumption that dominance is a partial order, and specifically antisymmetric. In contrast, as we will see directly below, the extensionality condition on c-command has significant force in limiting the class of well-formed structures, beyond the other constraints we are considering.

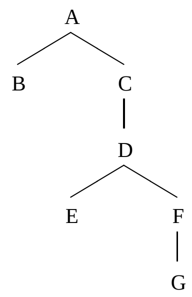
[25]Earlier, we remarked that the notion of root is not definable from c-command (and precedence) alone. This result was obtained when we considered trees in which there is a pair of nodes that is indistinguishable by c-command. By placing the extensionality requirement on c-command, however, such a situation is no longer possible. Yet as we have seen, non-trivial node chains can nonetheless arise in structures satisfying extensionality, like that in (30)b. However, even in this context, we have seen that non-trivial node chains cannot occur at the root of a structure. Thus, we can conclude that there can be only one node in an extensionality-satisfying tree structure that dominates every node (in the sense of $\preceq^{C-}$); the uniqueness of the root thereby follows from extensionality. Moreover, the notion of root is now definable, as the fact that every node other than the root is c-commanded by some node prevents the construction of automorphisms of the sort necessary to show non-definability.

[26]At the root, a node chain of length zero can be expanded to a node chain of arbitrary length.

[27]In Kayne's discussion, the only explicit reference he makes to non-trivial node chains concerns head-complement structures at the bottom of a phrase marker, like that in (35). As noted in the text, the LCA tolerates non-trivial node chains much more generally. Though the extensionality condition we propose blocks many of the node chains Kayne's LCA allows, it does tolerate node chains of length 1 internal to a phrase marker. Thus, the following is a well-formed

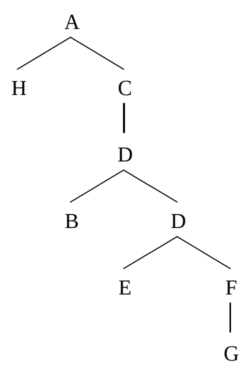
representation:

(i)



Such an internal node chain (created by the addition of node C) is not necessary in case B is combined with the constituent rooted in D via substitution. However, such internal node chains are crucial in allowing the embedding of a phrase to which a head has adjoined. Thus, unless we create the node chain consisting of nodes C and D in (ii), the phrase to which head B adjoins cannot be further embedded, as B and H could not be ordered. (cf. Kayne (1994, pp.30-32)):

(ii)



Whether such adjunction is permitted is an open empirical question, but see Kayne (1991) for a proposal that rests on this possibility.

[28]Frank and Kuminiak (2000) present an alternative conception of extensionality, stated as a condition on the asymmetric c-command relation, which derives constraints on the structure and occurrence of specifiers without resort to adjunction.

[29]Chomsky's (1986) empirical motivation for taking adjunction to produce a distinctive sort of structural configuration concerns locality conditions on syntactic movement. Kayne (1994) and Hornstein (1995, pp.120-122) exploit the distinctive properties of adjunction structures in the analysis of instances of quantifier-pronoun binding that do not seem to respect c-command.

[30]The reader will no doubt have noticed that under current structural assumptions, the view of c-command we have been assuming does not actually give the desired result: there is a node which dominates *everyone* but does not dominate *who*, namely C′, thereby blocking the possibility of c-command. May assumes that the necessary command relation is instead that of m-command:

(i) A m-commands B iff every maximal projection dominating A also dominates B, and neither A nor B dominates the other.

In the framework we have been developing, it would be unfortunate to have to import an additional command relation into the machinery, and without a notion of projection, there is little hope that m-command could be defined in terms of c-command. Instead, we will assume that the relevant structural relation is not between *everyone* and *who*, but rather between *everyone* and the C^0 head, which we assume contains interpretable wh-features. In this way, it is sufficient that *everyone* c-commands C^0.

[31]Presumably such an extension would also be required for the precedence relation, though we are aware of no cases in which differences in precedence arising from adjunction as compared to substitution are relevant.

[32]Under this more conservative approach, there must still be some mechanism for determining which segments together constitute a single category. One could imagine that this determination could be made by adding an equivalence relation over segments to the underlying formal structure, or else that it is derivable from identity of node labels under a certain configuration.

[33]Kayne takes domination to be irreflexive. However, as we take domination to be reflexive as well, we have added the modifier *properly*.

[34]Indeed under the assumption that the segments of a single category enter into identical c-command relations, the extensionality condition prevents us from even considering the use of segments and categories.

[35]The fourth logical possibility, namely that there is no c-command between A and B would not constitute a mode of combination. Note also that we are ignoring the possibility of combining α and β at any node other than the root. We turn to this possibility in section 7.2 below.

[36]What is meant by structural well-formedness conditions here is the set of conditions that we impose on the c-command relation to guarantee that it corresponds to a well-formed tree. Though we do not as yet have a complete set of such conditions, we might assume here the proposal tentatively suggested in section 4 in which the standard dominance conditions are applied to the derived relation $\preceq^{C-}$.

[37]One might argue that the third logical possibility of not asserting any c-command relation between the two roots is ruled out on conceptual grounds, as it would not constitute a case of combination. Moreover, we can show that the resulting structure is ill-formed. If no c-command relation is asserted between two roots r and r', the resulting derived relation $\preceq^{C-}$ will have both $r \preceq^{C-} r'$ and $r' \preceq^{C-} r$. Under the assumption that derived dominance must be antisymmetric, the impossibility follows.

[38]Another case of c-command relations whose existence we need to explain concerns those between adjoined element and all the nodes c-commanded by the target of adjunction. Such c-command relations are needed if, for example, an adjoined head is to c-command its trace. We return to this problem below.

[39]By taking Kayne's LCA to be one of the conditions that applies at each point in the derivation, we can understand

how the sorts of node chains discussed in section 5 can arise during a derivational process that uses only the derivational operations in (47). Specifically, suppose that substitution is applied to two heads. As discussed earlier, the presence of a mutual command relation between these heads will yield a lack of ordering among them in violation of the LCA. Just as a new root node is created when a structure no longer satisfies rooted, so too can we imagine that an additional node is created to enrich the structure of the complement so that it is LCA-compliant. This addition of structure can be viewed as a repair strategy of sorts, similar in spirit to the proposals of Chomsky (1995, ch.4) and Moro (1997; 2000), where LCA violations are rescued via movement.

[40]Note that $\preceq^C$ does not provide the appropriate answer in this case, as it would yield that A$\preceq^C$B and B$\preceq^C$A. As we noted above, $\preceq^{C-}$ yields results identical to $\preceq^C$ in the case of pure trees (derived with substitution alone), but, as we now see, diverges in the case of adjunction structures. In section 4, however, we argued for using $\preceq^{C-}$, on the basis of the fact this notion of derived dominance, and not $\preceq^C$, allows us to capture the difference between different conceptions of c-command in terms of constraints on the derived dominance relation. Furthermore, it is this version of the order schema that is relevant in the standard definition of c-command, where nodes that stand in the dominance relation are explicitly blocked from standing in the c-command relation. It is therefore interesting to note that once we move to the richer class of adjunction structures, it is once again the $\preceq^{C-}$ version of the derived dominance relation that functions correctly.

[41]See Frank, Hagstrom, and Vijay-Shanker (1999) for further discussion.

[42]Getting this to follow from the operations given requires that head A combines with head B prior to combination with C, an interarboreal derivation of the sort envisioned by Bobaljik and Brown (1997). A perhaps less elegant alternative to this might define an alternative operation of internal adjunction that explicitly states the necessary c-command relations between the adjunct and the sisters of the target.

[43]In fact, Chomsky (1993) argues that the extension requirement holds only for those cases of substitution applying prior to SPELL-OUT. Chomsky (1995, ch.4), however, suggests that all movement after SPELL-OUT is in fact adjunction. Following this line, substitution obeys the extension requirement uniformly. It is this that we will assume.

[44]The argument Chomsky provides involves anti-reconstruction effects of the sort discussed by Freidin (1986) and Lebeaux (1988).

[45]This raises the interesting question (brought to our attention by Paolo Acquaviva) of whether interpretations may be composed from c-command relations alone. This would require considerable rethinking of the usual ways of assigning meanings to interpretations, but we believe that it may have some attractive consequences.

[46]Note that this leaves open the possibility that we could perform sister adjunction at an internal node, as that would not disturb the surrounding c-command relations. In current syntactic theory, such an operation is not usually assumed, and indeed, we could rule it out through the imposition of either local asymmetry or local transitivity of c-command (both of which have the effect of allowing only binary branching structures). In any case, we note that sister adjunction would be consistent with a conception of derivations incorporating derivational monotonicity.

[47] Marcus, Hindle, and Fleck (1983), Gorrell (1995) and Weinberg (1993) suggest that precisely this kind of monotonicity requirement (stated in terms of dominance and precedence) plays an important role in modeling human performance in sentence processing. Frank, Vijay-Shanker, and Chen (1996) and Frank and Vijay-Shanker (2000) argue, however, that even in this domain, a monotonicity constraint on c-command has even greater explanatory power. Furthermore, c-command allows us to provide a linguistic basis for the kind of underspecified attachments that had been assumed in these previous dominance-based parsing models.

References

Backofen, Rolf, James Rogers, and K. Vijay-Shanker. 1995. A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language and Information*, 4(1):5–39.

Barker, Chris and Geoffrey K. Pullum. 1990. A theory of command relations. *Linguistics and Philosophy*, 15:1–34.

Bobaljik, Jonathan David and Samuel Brown. 1997. Interarboreal operations: Head movement and the extension requirement. *Linguistic Inquiry*, 28(2):345–356.

Chametzky, Robert. 1996. *A Theory of Phrase Markers and the Extended Base*. State University of New York Press, Albany.

Chomsky, Noam. 1955. *The Logical Structure of Linguistic Theory*. Distributed by Indiana University Linguistics Club. Published in part by Plenum, New York, 1975.

Chomsky, Noam. 1986. *Barriers*. MIT Press, Cambridge, MA.

Chomsky, Noam. 1993. A minimalist program for linguistic theory. In Kenneth Hale and Samuel Jay Keyser, editors, *The View from Building 20*. MIT Press, Cambridge, MA, pages 1–52.

Chomsky, Noam. 1994. Bare phrase structure. MIT Occasional Papers in Linguistics 5, Department of Linguistics and Philosophy, MIT.

Chomsky, Noam. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.

Epstein, Samuel David. 1998. Overt scope marking and covert verb-second. *Linguistic Inquiry*, 29(2):181–227.

Epstein, Samuel David. 1999. Un-principled syntax and the derivation of syntactic relations. In Samuel David Epstein and Norbert Hornstein, editors, *Working Minimalism*. MIT Press, Cambridge, MA, pages 317–345.

Epstein, Samuel David, Erich M. Groat, Ruriko Kawashima, and Hisatsugu Kitahara. 1998. *A Derivational Approach to Syntactic Relations*. Oxford University Press, New York.

Frank, Robert, Paul Hagstrom, and K. Vijay-Shanker. 1999. Roots, constituents and c-command. *GLOW Newsletter*, 42:26–27. Paper presented at the 22nd GLOW Colloquium.

Frank, Robert and Anthony Kroch. 1995. Generalized transformations and the theory of grammar. *Studia Linguistica*, 49(2):103–151.

Frank, Robert, Seth Kulick, and K. Vijay-Shanker. 2000. Monotonic c-command: A new perspective on tree adjoining grammar. *Grammars*, 3(2/3):151–173.

Frank, Robert and Fero Kuminiak. 2000. Primitive asymmetric c-command derives X′-theory. In M. Hirotani, A. Coetzee, N. Hall, and J-Y. Kim, editors, *Proceedings of the 30th Annual Meeting of the North East Linguistics Society*, volume 1, pages 203–217. Graduate Linguistics Students Association, University of Massachusetts, Amherst.

Frank, Robert and K. Vijay-Shanker. 2000. Lowering across languages. In Marica De Vincenzi and Vincenzo Lombardo, editors, *Cross-Linguistic Perspectives on Language Processing*. Kluwer, Dordrecht, pages 63–87.

Frank, Robert, K. Vijay-Shanker, and John Chen. 1996. Dominance, precedence and c-command in description-based parsing. In Carlos Martin-Vide, editor, *Proceedings of XII Congreso de Lenguajes Naturales y Lenguajes Formales*, pages 61–74, Barcelona, Spain. PPU.

Freidin, Robert. 1986. Fundamental issues in the theory of binding. In Barbara Lust, editor, *Studies in the Acquisition of Anaphora*, volume 1. D. Reidel, Dordrecht, pages 151–188.

Gorrell, Paul. 1995. *Syntax and Parsing*. Cambridge University Press, Cambridge.

Grefe, Carsten and Marcus Kracht. 1996. Adjunction structures and syntactic domains. Technical Report 80, Sonderforschungsbereich 340, Universität Stuttgart and Universität Tübingen.

Hornstein, Norbert. 1995. *Logical Form: From GB to Minimalism*. Basil Blackwell, Oxford.

Jackendoff, Ray. 1972. *Semantic Interpretation and Generative Grammar*. MIT Press, Cambridge, MA.

Kayne, Richard. 1991. Romance clitics, verb movement, and PRO. *Linguistic Inquiry*, 22:647–686.

Kayne, Richard. 1994. *The Antisymmetry of Syntax*. MIT Press, Cambridge, MA.

Kitahara, Hisatsugu. 1995. Target alpha: Deducing strict cyclicity from derivational economy. *Linguistic Inquiry*, 26(1):47–77.

Kracht, Marcus. 1993. Mathematical aspects of command relations. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 240–249, Utrecht.

Kulick, Seth, Robert Frank, and K. Vijay-Shanker. 2000. Defective complements in tree adjoining grammar. In Alexander Williams and Elsi Kaiser, editors, *Penn Working Papers in Linguistics*, volume 6(3). University of Pennsylvania, Department of Linguistics, pages 35–73.

Kupin, Joseph. 1978. A motivated alternative to phrase markers. *Linguistic Inquiry*, 9(2):303–308.

Lasnik, Howard and Joseph Kupin. 1977. A restrictive theory of transformational grammar. *Theoretical Linguistics*, 4(3):173–196.

Lebeaux, David. 1988. *Language Acquisition and the Form of the Grammar*. Ph.D. thesis, University of Massachusetts, Amherst, MA.

Marcus, Mitchell, Donald Hindle, and Margaret Fleck. 1983. D-theory: Talking about talking about trees. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 129–136, Cambridge, MA.

May, Robert. 1985. *Logical Form*. MIT Press, Cambridge, MA.

McCawley, James. 1982. Parentheticals and discontinuous constituent structure. *Linguistic Inquiry*, 13(1):91–106.

Moro, Andrea. 1997. Dynamic antisymmetry: Movement as a symmetry-breaking phenomenon. *Studia Linguistica*, 51(1):50–76.

Moro, Andrea. 2000. *Dynamic Antisymmetry*. MIT Press, Cambridge, MA.

Partee, Barbara, Alice ter Meulen, and Robert E. Wall. 1993. *Mathematical Methods in Linguistics*. Kluwer, Dordrecht.

Reinhart, Tanya. 1976. *The Syntactic Domain of Anaphora*. Ph.D. thesis, MIT, Cambridge, MA.

Reinhart, Tanya. 1981. Definite NP anaphora and c-command domains. *Linguistic Inquiry*, 12(4):605–635.

Reiss, Charles. 2001. The ocp and nobanana or philosophical and empirical reasons to ban constraints from linguistic theory. Manuscript, Concordia University.

Richardson, J.F. and Robert Chametzky. 1985. A string based redefinition of c-command. In Steve Berman, Jae-Woong Choe, and Joyce McDonough, editors, *Proceedings of the 15th Annual Meeting of the North East Linguistics Society*, pages 332–361. Graduate Linguistics Students Association, University of Massachusetts, Amherst.

Weinberg, Amy. 1993. Parameters in the theory of sentence processing: Minimal committment theory goes east. *Journal of Psycholinguistic Research*, 22(3):339–364.

Robert Frank
Department of Cognitive Science
Johns Hopkins University
3400 North Charles Street
Baltimore, MD 21218
rfrank@cogsci.jhu.edu

K. Vijay-Shanker
Department of Computer and Information Science
University of Delaware
Newark, DE 19716
vijay@cis.udel.edu