Sep 90 my copy
w/corrections
CB

CHRIS BARKER AND GEOFFREY K. PULLUM

A THEORY OF COMMAND RELATIONS*

0. INTRODUCTION

Since Langacker (1969) introduced the 'command' relation into the literature of syntax, such relations, which we can refer to generally as COMMAND RELATIONS, have increased greatly in importance. Yet the literature contains no systematic formal treatment of the general topic of command relations (notwithstanding a few articles focussed on the definitions of particular command relations, e.g , Aoun and Sportiche (1982), Saito (1984), Richardson and Chametzky (1985)). This paper sets out an explicit general definition that aims to capture the essential properties common to all command relations, and to allow specific relations to be distinguished in terms of different settings of a single parameter. We prove a number of theorems, some about command relations as a class, and some about particular command relations

The outline of the paper is as follows Section 1 presents a general definition of a command relation, and Section 2 discusses a number of command relations from the literature in light of this definition In Sections 3 and 4 we deal briefly with symmetric relations like 'clausemate', which are definable in terms of command relations, and 'government', which is a closely linked topic Section 5 generalizes the results of previous sections from trees to arbitrary labeled graphs Section 6 sketches a proposal for broadening the definition of a command relation to include certain proposals (in particular, the more elaborate of Reinhart's (1976) two versions of c-command) that are not encompassed by the framework given in Section 1. We close with a summary, and an appendix containing proofs of the results stated in the paper.

1. A GENERAL DEFINITION FOR COMMAND RELATIONS

This section presents our formal definition of 'command relation' We interpret the definition schematically, and we characterize the properties common to all command relations.

* We gratefully acknowledge the assistance and commentary provided by Robert Chametzkv, Gerald Gazdar, William Ladusaw Louise McNally, John Richardson, William Rounds, and several anonymous referees The Syntax Research Center at the University of California, Santa Cruz, provided logistical and other support for this research, and Calvin Pullum is responsible for the diagrams

Linguistics and Philosophy **13** 1–34, 1990

1.1. *The Definition*

Command relations are, of course, relations. Let N be some set (usually, for our purposes, the set of nodes in a tree or other graph). Then a RELATION R on N is a subset of $N \times N$, i.e., a set of ordered pairs of elements of N. If $\langle a, b\rangle \in R$, we say that a stands in the R relation to b, and we write aRb or $\neg aRb$, depending on whether $\langle a, b\rangle \in R$. A relation R on a set N is REFLEXIVE iff aRa for all a in N; IRREFLEXIVE iff $\neg aRa$ for all a in N; SYMMETRIC iff aRb implies bRa;[1] ASYMMETRIC iff aRb implies $\neg bRa$; ANTISYMMETRIC iff aRb and bRa implies $a = b$; and TRANSITIVE iff aRb and bRc implies aRc.

We proceed by developing a definition of the notion 'tree'. For the sake of familiarity, we adopt Wall's (1972, p 149) definition without substantive modification.

DEFINITION 1. A tree T is a 5-tuple $T = \langle N, L, \overset{D}{\geqslant}, \overset{P}{<}, \text{LABEL}\rangle$, where

> N is a finite set, the nodes of T,
> L is a finite set, the labels of T,
> $\overset{D}{\geqslant}$ is a reflexive, antisymmetric relation on N, the dominance relation of T,
> $\overset{P}{<}$ is an irreflexive, asymmetric, transitive relation on N, the precedence relation of T, and
> LABEL is a total function from N into L, the labeling function of T,

such that for all a, b, c, and d from N and some unique r in N (the ROOT NODE of T), the following hold·

(i) The Single Root Condition: $r \overset{D}{\geqslant} a$
(ii) The Exclusivity Condition: $(a \overset{D}{\geqslant} b \vee b \overset{D}{\geqslant} a) \leftrightarrow \neg(a \overset{P}{<} b \vee b \overset{P}{<} a)$
(iii) The Nontangling Condition: $(a \overset{P}{<} b \wedge a \overset{D}{\geqslant} c \wedge b \overset{D}{\geqslant} d) \rightarrow c \overset{P}{<} d$

To say that the relation LABEL is a total function merely requires that every node a have a ~~unique~~ label l such that $\langle a, l\rangle \in \text{LABEL}$. We shall see that many useful command relations can be defined independently of LABEL, but some, such as MAX-command, crucially refer to the syntactic category of a node as specified by the labeling function.

Note that the dominance relation $\overset{D}{\geqslant}$ is reflexive; it will occasionally be convenient to use $\overset{D}{>}$, the PROPER DOMINANCE relation, which will be just like $\overset{D}{\geqslant}$ but with all pairs of the form $\langle a, a\rangle$ removed We will refer to the

[1] Throughout the remainder of the paper, we will omit universal quantifiers where we feel the reader can readily supply them For example, the criterion for symmetry should be interpreted as $\forall a, b \in N[aRb \rightarrow bRa]$

nodes properly dominated by a node a as the DESCENDANTS of a, and to the nodes that properly dominate a as the ANCESTORS of a.

In this section we will define command relations as determined by properties of nodes, where a PROPERTY P on a set N is a subset of N. If $a \in P$ we say that a SATISFIES P, and we write $P(a)$. (In Section 6 we will consider a parallel approach in which command relations are generated not by properties, but by two-place relations.) For all command relations known to us, only those members of P which dominate a play a part in determining what a commands, we will make this generalization explicit by defining command relations in terms of the upper bounds of a with respect to P:

DEFINITION 2. The set of UPPER BOUNDS for a with respect to a predicate P, written $\mathrm{UB}(a, P)$, is given by

$$(1) \qquad \mathrm{UB}(a, P) = \{b \mid b \overset{D}{>} a \wedge P(b)\}$$

Thus b is an upper bound for a if and only if it properly dominates a and also satisfies P.

We now present our central definition, the general definition of a command relation between nodes in a tree.

DEFINITION 3. Let $T = \langle N, L, \overset{D}{\geqslant}, \overset{P}{<}, \text{LABEL}\rangle$ be a tree and let P be an arbitrary property over N. Then the COMMAND RELATION C_P induced by P on T is given by.

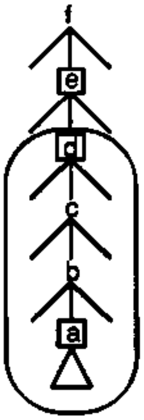
$$(2) \qquad C_P = \{\langle a, b\rangle \mid \forall x[(x \in \mathrm{UB}(a, P)) \rightarrow x \overset{D}{\geqslant} b]\}$$

We say that P GENERATES C_P. If aC_Pb, we say that a P-COMMANDS b, or, if the context permits reliable disambiguation, a COMMANDS b, the node a being the COMMANDER. We shall call the set of all nodes commanded by a the COMMAND DOMAIN of a (cf. the use by Reinhart (1976, 1981, 1983), Rouveret and Vergnaud (1980), Aoun and Sportiche (1982), and others of the term 'domain' in this sense). If R is a relation over N, then R is a command relation if and only if there exists some property P such that $R = C_P$.

1.2. *Interpreting the Definition*; *the Intersection Theorem*

Definition 3 says that a P-commands b iff every upper bound for a (with respect to P) dominates b. Thus there is a correspondence between properties of nodes on the one hand and command relations on the other. Assume that we have a tree T, a property P over nodes in that tree, and a node a in T and consider the tree schema in (3)

(3)



The boxed nodes represent members of *P* In (3), assume *a*, *d*. and *e* are in *P*; *d* and *e* are (the only) upper bounds for *a* The loop encloses the *P*-command domain of *a*; it contains all and only the nodes that are dominated by all the upper bounds for *a*

We have reconstructed the notion of a command relation in such a way that (immediately from Definition 3) every node commands itself: i.e., all command relations are reflexive under our account Thus, for example, *a* in (3) is dominated by *d* and *e*, hence is dominated by all of its upper bounds, hence is *P*-commanded by itself

Clearly *d* is the crucial node for fixing the command domain of *a*. Notice that *d* is the closest upper bound to *a*. in the sense that it dominates no other upper bound We formalize this sense of 'closest' as follows

DEFINITION 4. The set of MINIMAL UPPER BOUNDS for *a* with respect to *P*, written MUB(a, P), is given by

(4) $$\text{MUB}(a, P) = \{b \mid b \in \text{UB}(a, P) \wedge \forall x[(x \in \text{UB}(a, P) \wedge b \overset{D}{\geq} x) \rightarrow b = x]\}$$

The definition of the minimal upper bounds for *a* picks out a subset of *a*'s upper bounds. An upper bound is minimal only if there is no other upper bound on the path between it and *a*. For trees there is at most one minimal upper bound (but see Section 5 for a discussion of the general case)

Note that the command relation diagrammed in (3) is not transitive. We have shown that *a* commands *d*, and it is easy to see that *d* *P*-commands *e* But we have also established that *a* does not command *e*, since *e* is not dominated by all of *a*'s upper bounds From this counterexample it follows that command relations are not in general transitive under our account.

It is easy to see how the command domain of *a* will vary with the membership of *P*. For instance, if we minimally extend *P* to include *c*, then c becomes an upper bound on the command domain of a In fact, it will be the minimal upper bound, and the command domain will contain only nodes dominated by c. If, on the other hand, we reduce the original membership of P so that it excludes d, then e becomes the minimal (and only) upper bound for a The command domain grows to encompass nodes dominated by e but not dominated by d

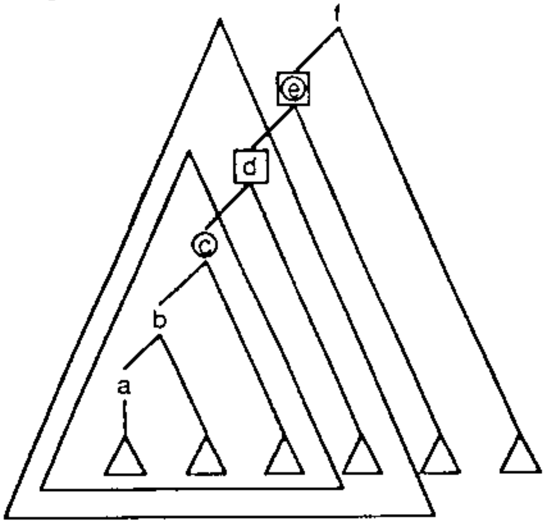
Thus the size of the generator property and the size of the command relation it generates varies inversely: enlarging the membership of the property P shrinks the resulting command relation, and making the conditions for membership in P more stringent causes the command relation to be larger. A central result of this paper, proved as Theorem 1 in the appendix, characterizes the nature of this relationship mathematically.

(5) Intersection theorem $C_P \cap C_Q = C_{P \cup Q}$
Intersection over command relations corresponds to union over their generating properties.

In words· taking the intersection of the set of pairs of nodes in one command relation C_1 with the set of pairs of nodes in another command relation C_2 always yields a third relation C_3 which is also a command relation. That is, the set of command relations is closed under intersection Moreover, C_3 is a specific relation determined by C_1 and C_2: it is the command relation generated by the union of the generator properties for C_1 and C_2 Thus if P generates C_1 and Q generates C_2, then $P \cup Q$ generates C_3

We represent some of the consequences of the intersection theorem schematically in (6), using circles to mark nodes in P and square boxes to mark nodes in Q

(6)



A node surrounded by both a circle and a square belongs to both P and Q The minimal upper bound for P is the closest circled node to a, and the minimal upper bound for Q will be the nearest boxed node. Thus the minimal upper bound with respect to $P \cup Q$ is the closest node to a marked either with a circle or a square (or both)

This makes it clear that the command domain induced by the union of the two generating properties will always be equal to the domain induced by either P or Q, whichever domain is smaller. Since one domain always entirely contains the other, the command domain associated with the command relation generated by the union of P and Q is the intersection of the domains with respect to the component properties.

This means, for example, that a linguist who finds some syntactic constraint is captured by requiring a pair of nodes to satisfy the definitions of two command relations simultaneously can be assured that there is a single command relation that captures the constraint; and further, this third relation can immediately be identified In Section 2 we discuss two actual cases from the literature where this fact about intersecting command relations is of relevance

Theorem 1 establishes that the set of all command relations forms a join semilattice (intersection being the join operation). Section 6 deals with the complementary question of what happens when we take the union rather than the intersection of two command relations.

1.3. *Characterizing Command Relations*

Definition 3 permits specification of a command relation merely by describing the membership of the generator property P Section 2 contains several examples of this technique Each of the reconstructed command relations in Section 2 will necessarily share those features it has by virtue of falling under our definition of a command relation, a relation so defined automatically behaves in a reasonable way in general and with respect to limiting cases. More specifically, Definition 3 guarantees that the induced command relation will exhibit the following characteristics common to all command relations.

(7)a. Ambidextrousness: command relations are insensitive to precedence.

b Boundedness: a root acts like an upper bound for every node it dominates.

c. Constituency. command domains correspond to constituents.

d Descent. if a commands b, then a commands b's descendants.

e. Embeddability: command relations on a subtree are insensitive to material external to that subtree.

f. Fairness: an upper bound for a node a is an upper bound for every node that a dominates.

These properties are given formal definitions in the appendix. Note that the properties listed in (7) hold of command relations in addition to the property of reflexivity and the constraint imposed by the intersection theorem

The six conditions listed in (7) describe situations in which restrictions on pairs of nodes can remove parts of a command relation in such a way that the result is no longer a command relation For instance, Langacker (1969) suggests that pronoun anaphora is sensitive to the complex relation 'precedes and commands'. A pair of nodes $\langle a, b\rangle$ stands in this compound relation if and only if a commands b and a precedes b. Ambidextrousness entails that the relation 'precedes and commands' is not a command relation.

This does not mean that Ambidextrousness prevents the generating property for a command relation from depending at least partly on precedence. For instance, we could choose for a generator the property of being a node which is preceded by no other node, that is, the set containing all and only those nodes of the leftmost edge of the tree. Call this property π. All nodes in a tree bear the command relation C_π to all nodes that precede them, though not necessarily to all nodes that follow them. But Ambidextrousness ensures that the sensitivity of C_π (or any other command relation) to linear precedence is due purely to the nature of the generator. To determine where a command relation fails to hold we examine only hierarchical relationships, and C_π is no exception to this, thus the formation of command relations from generators cannot be sensitive to precedence.

Boundedness describes the behavior of command relations for the limiting case in which a node a has no upper bounds. This will happen, for instance, when the generator property is empty. Boundedness says that in such a situation a commands every node in the tree. Thus a command domain is never empty

The remaining conditions can similarly be understood as constraints disallowing certain narrowings of command relations.

Constituency disallows lacunae in a command domain This means, for example, that the compound relation 'a commands b and neither a nor b

dominates the other' is not a command relation, it would give rise to a command domain consisting of two unconnected sets of nodes rather than a constituent (rooted subtree).

Descent excludes compound relations such as 'a commands b and a is not more than 5 nodes away from b', commanding a node entails commanding *all* its descendents.

Embeddability excludes a compound relation such as 'a commands b and no NP nodes dominate a', either a command relation holds in some subtree or it does not: embedding it under some new structure cannot change that.

Finally, Fairness excludes a compound relation that could be glossed 'a commands b or a is an NP node'; such a relation could include certain pairs $\langle x_1, x_2 \rangle$ where x_2 is not in the relevant command domain, the reason for the inclusion of the pair being simply that x_1 is labeled NP Fairness blocks such discriminatory treatment; the upper bound, whatever it is, has the same consequences for all nodes.

We discuss Constituency in Sections 2 1 and 5; Descent in Section 5; Embeddability in Section 5; and Fairness is mentioned again in Section 6

2 Command relations in the linguistic literature

In this section we survey some well-known command relations that have been defined (or at least roughly sketched) in the linguistic literature and used in some context for syntactic description We will discuss relations that we call 'S-command' (Langacker's (1969) *command*), 'NP-command' (defined here for convenience of reference), 'K-command' (Lasnik's (1976) *kommand*), 'MAX-command' (roughly what Chomsky (1986) calls 'm-command'), Klima's (1964) 'in construction with' relation, 'c-command' (Reinhart (1974)), and 'IDC-command' (Pullum (1986)). In Section 6 we mention a more elaborate definition of c-command due to Reinhart (1976), which is not a command relation at all according to Definition 3

Before investigating individual command relations we will make some more general comments which apply to nearly all the definitions found in the literature Consider the following generic informal definition of c-command

(8) A node a c-commands a node b if the first branching node dominating a also dominates b

Similar definitions have been common currency in linguistics over the last fifteen to twenty years. Yet terms such as 'branching node' and 'first' are almost always left undefined Both are clearly capable of more than one reasonable interpretation. As noted in Richardson and Chametsky (1985). for example, 'branching node' might mean 'node labeled with a category for which there are branching expansion rules in the grammar', and is taken to mean exactly that by McCawley (1984, p 231n), or it might mean 'node with at least two daughters'. The phrase 'first node' is likewise not transparent; it could mean 'first node down from the root', or 'first node from the left', or 'first node up from the commander', and so on

The normal usage takes a branching node to be one with more than one daughter, and 'first' to mean 'most deeply embedded', but this is not determined by any explicit statement in the literature.

In a similar vein, unless 'dominates' is taken to mean 'properly dominates', the branching node closest to the commander will often be the commander itself, which never seems to be the intended interpretation (it would reduce command and dominance to the same thing)

Furthermore, the use of 'if' leads only to a sufficient condition for the relation to hold, rather than necessary and sufficient conditions; clearly this has to be read as 'if and only if'.

Less trivially, consider a situation in which a node *a* has no branching ancestor Does the relation given in (8) hold of all pairs ⟨*a*, *b*⟩, since the condition is vacuously satisfied? Or does the relation always fail to hold by virtue of the ~~filaure~~ of the existence presupposition involved in the fail definite description 'the first branching node'? This is not resolved by any implicit convention or general understanding in the linguistic community, the definition in (8) leaves it completely open.

Our formal definition resolves all of these indeterminacies, as we will show. We force predicates like 'branching' and 'first' to be given as defined sets of nodes; we give necessary and sufficient conditions for command relations to hold; and we resolve the question about situations in which no node meets the upper bound condition in favor of vacuous satisfaction. the Boundedness condition guarantees that when *a* has no upper bounds, the root acts as if it were an upper bound

Nearly all definitions of command relations in the literature are misformulated in one way or another, usually in several ways. The defects are usually not profound or difficult to remedy, though occasionally they do lead to significant misunderstanding or unclarities. We will not catalog the vagaries of each definition quoted below from the literature; rather, we will present a reconstruction of the apparently intended relation in a proposed unitary standard form for definitions of command relations, redefining each simply by setting the property parameter in our general definition. Where our version of a command relation diverges from what can be read literally in the source we cite, the consequences of our version are never less desirable, and in a number of cases are more desirable, than the original

We have three aims in this reformulation task: (i) to illustrate the intended use of our definitional framework using real examples that syntacticians wil be familiar with; (ii) to present the beginnings of a precise taxonomy of command relations as used in syntax; and (iii) to clarify and explicate the apparent intent of some familiar syntactic studies.

In each case, we will proceed by specifying a particular property of nodes in general When plugged into Definition 3, this property will generate (our reconstruction of) the desired command relation Throughout this section, the particular command relation under examination is assumed to be defined over a tree $T = \langle N, L, \overset{D}{\geq}, \overset{P}{<}, \text{LABEL}\rangle$, which we refer to when necessary.

2.1 *S-command* (*Langacker's 'Command'*)

In a discussion of the conditions under which personal pronouns can take intrasentential antecedents, Langacker (1969, p. 167) describes the following command relation (we ~~silently~~ adjust his notion to use early alphabet lower-case italic letters for nodes):

(9) We will say that a node *a* 'commands' another node *b* if
(1) neither *a* nor *b* dominates the other; and
(2) the S-node that most immediately dominates *a* also dominates *b*

Because of the reference to nodes labeled *S*, we will call this relation 'S-command'.

We have already chosen not to include Langacker's requirement (1) as a restriction on command relations in general Since this decision relates intimately to some deep properties of command relations (including reflexivity and the Constituency condition), we should comment on the decision at this point. Some authors in the subsequent literature have followed Langacker in stipulating that commanders neither dominate commanded nodes nor are dominated by them, and others have not. We are aware of no authors who give definite motivation for the exclusion, and nothing in Langacker's paper seems to hinge on it At least two works explicitly provide arguments in support of allowing command relations to overlap with dominance. Reinhart (1981, pp. 614–615; 1983, pp. 20–21), and Jackendoff (1972, p 312, n. 10) Unfortunately, the empirical argument in Reinhart (1981, 1983) is not strong (Richardson and Chametzky (1985) regard it as invalid), and we are unable to understand Jackendoff's remarks. It is therefore formal convenience rather than factual argumentation that leads us to follow these authors and allow command relations to overlap with dominance.

As a consequence of this assumption, if a situation were to arise in syntactic description in which nodes in a dominance relationship had to be treated differently from other nodes in some command relation, our framework requires that this fact should be made to follow from other considerations rather than from the definition of the command relation involved

To see that ignoring the antidominance stipulation does no harm in the case of S-command, note that Langacker himself adds ~~to~~ a second 7
requirement to nodes in an anaphoric relation He argues that a pronoun may not simultaneously command and precede its antecedent. If a node a precedes a node b, no dominance relation can hold between a and b (by the exclusivity Condition on trees: see Definition 1) If a pronoun node a linked to an antecedent node b fails to satisfy Langacker's constraint, then a precedes b, hence a does not dominate b, hence condition (1) of Langacker's definition is automatically satisfied. Langacker's requirement (1) is redundant, in other words, and it is clear that no harm results from removing it from the definition.

This leaves only requirement (2) to deal with. Most of the effect of the restriction is already built in to our definition schema; we need only to achieve the necessary reference to an 'S-node' by setting the property P of Definition 3 to be the set of nodes whose label is 'S'. Our definition of Langacker's notion of command is thus as follows:

DEFINITION 5. S-command is the command relation C_{P_1} where P_1 is given by

(10) $P_1 = \{a \mid \textsc{label}(a) = \mathrm{S}\}$

A node satisfies P_1 if and only if its label is S. We claim that, once it is seen that the burden of excluding nodes in a dominance relationship is taken up by the precedence condition in the constraint Langecker needs to state, S-command is precisely the relation that captures Langacker's intent.

2.2 *NP-command*

Purely for reasons of convenience in the next section, we now define a very similar relation that we can call NP-command by analogy with the definition of S-command We simply replace S by NP in the definition:

DEFINITION 6. NP-command is the command relation C_{P_2} where P_2 is given by

(11) $P_2 = \{a \mid \textsc{label}(a) = \text{NP}\}$

A node a satisfies the predicate P_2 if and only if its label is NP, and NP-commands all the nodes dominated by the lowest NP that properly dominates it

2.3. *K-command* (*Lasnik's kommand*)

Lasnik (1976, p. 15) proposes that the following command relation is the relevant one for the description of constraints on anaphora.

(12) *a* KOMMANDS *b* if the minimal cyclic node dominating *a* also dominates *b*.

What definition of 'cyclic node' is provided is not very important here; we may assume that any linguistic theory making use of the notion will incorporate a list of the labels that are definitive of cyclic node status (or more interestingly, provide some postulates from which this status will follow). Lasnik does not define 'minimal', but it is clear enough that the minimal node with property *P* should be taken to be the one that is dominated by more nodes satisfying *P* than any other ancestor of *a*, as in our definition of minimal bound.

We therefore reconstruct what we believe is the intent of Lasnik's definition as follows, renaming the relation K-command

DEFINITION 7. K-command is the command relation C_{P_3} where P_3 is given by

(13) $P_3 = \{a \mid \textsc{label}(a) \in \{\text{S}, \text{NP}\}\}$

A node satisfies P_3 if and only if its label is either S or NP.

Consider now one consequence of the intersection theorem. Since P_3 is clearly the union of P_1 and P_2 (the generator properties for S-command and NP-command), we have

(14) K-command = $C_{P_3} = C_{P_1 \cup P_2}$ = S-command ∩ NP-command

It is easy to see how to construct the analog of the K-command relation for any collection of labels declared to be cyclic: simply take the intersection over the command relations corresponding to the members of the set of cyclic node labels.

The further generalization that replaces 'cyclic node' with 'node labeled by a maximal projection' yields the next relation we discuss

2.4 MAX-*command*

Under the cluster of proposals referred to as X-bar theory (see Jackendoff (1977), Pullum (1985), Gazdar et al. (1988), Kornai and Pullum (1990), lexical and phrasal categories are taken to be linked by a relation usually called 'projection'. every phrasal category is a projection of some lexical category to some level of inclusiveness (where projections further from the lexical category potentially include more structure). Often the lexical categories in the inventory are represented as X^k for some choice of lexical category X and integer k, where k is called the BAR LEVEL, but this is irrelevant and dispensable (see Kornai and Pullum (1990)); what is crucial here is simply that there is a class of nodes defined as MAXIMAL PROJECTIONS, either by reference to a stipulation of some number m such that all nodes labelled X^m for any X are maximal projections, or in some more tree-dependent way.

For simplicity, let us suppose that there is simply a set MAX of labels of the form X^m, and all and only the nodes whose labels are in MAX are maximal projections. Then:

DEFINITION 8. MAX-command is the command relation C_{P_4} where P_4 is given by

(15) $$P_4 = \{a \mid \text{LABEL}(a) \in \text{MAX}\}$$

We intend MAX-command to be a reconstruction of the relation that numerous recent writers refer to as 'c-command in the sense of Aoun and Sportiche (1982)'. Aoun and Sportiche are concerned to define a notion of government based on this command relation [2] Section 4 below treats the role of MAX-command and other command relations used in the definition of government in more detail.

2.5. *C-command and In-construction-with*

The name 'c-command' (for 'constituent-command') was originally suggested by G. N Clements to denote the relation that Reinhart (1974) calls 'superiority' As Reinhart states (1974, p. 105, n. 3), her superiority relation is identical to the inverse of the 'in construction with' given by

[2] Aoun and Sportiche's paper, though frequently cited, is unusually fraught with errors The first key definition given (1982, p 224, (44)) is ambiguous, containing an anaphoric *it* and an unbracketed conjunct, later (p 228) an unambiguous but clearly non-equivalent definition is given

Klima (1964, p. 297). Klima's definition reads (excising exemplificatory references to a diagram).

(16) A constituent is 'in construction with' another constituent if the former is dominated by the first branching node that dominates the latter

Reinhart's definition of 'superior to' (1974, p. 94) reads:

(17) A node *a* is 'superior to' a node *b* if the first branching node dominating *a* also dominates *b* (In this case, *b* is said to be *inferior* to *a*)

Although in-construction-with (= inferior-to) contains exactly the same information as superior-to (= c-commands), the former is not a command relation To see this, consider that if *a* is in-construction-with *b*, and *b* has a daughter *c*, then *a* is not necessarily in-construction-with *c*; but this is a violation of the Descent condition ((7d) in Section 1 above). (From this we have the result that the inverse of a command relation is not necessarily a command relation)

C-command is slightly more complex to represent than previous examples of command relations Since some writers (e.g., Aoun and Sportiche (1982), Chomsky (1981)) have used the name *c-command* for other relations, especially for MAX-command, we will occasionally refer to c-command as 'classical c-command' to avoid confusion

In constructing our formal definition of classical c-command, we do not need to take steps to ensure that the branching node dominates the commander or that it be the 'first' (maximally dominated) such node, since we have incorporated these requirements into the definition of a command relation We concentrate instead on defining what it means to be a branching node. It will be convenient to first define the mother relation over nodes:

DEFINITION 9. *M*, the MOTHFR relation, is given by

(18) $M = \{\langle a, b\rangle | a \overset{P}{>} b \wedge \neg\exists x[a \overset{P}{>} x \overset{P}{>} b]\}$

If *aMb* we say that *a* is the MOTHER of *b*, or that *a* IMMEDIATELY DOMINATES *b*. Notice that *M* is the smallest relation whose reflexive and transitive closure is the dominance relation. Definition 9 says that a node stands in the mother relation to another node if it is a minimal proper dominator of that node.

Now we can define c-command:

DEFINITION 10: C-command is the command relation C_{P_5} where P_5 is given by

(19) $P_5 = \{a \mid \exists x, y[x \neq y \wedge aMx \wedge aMy]\}$

P_5 represents the predicate 'is a branching node': a node satisfies P_5 if and only if it is the mother of two distinct nodes.

Although classical c-command is sometimes referred to as determining some kind of 'locality' (see, e.g., Berwick and Wexler (1982)), it is not local at all: nonbranching chains of nodes can be arbitrarily long, and a c-commanding node can be unboundedly far away from a node that it c-commands

A useful application of the intersection theorem arises from a proposal of Rouveret and Vergnaud (1980, p 111), who redefine c-command as follows (yielding a new relation, but one for which they keep the same name):

(20) The category *a* c-commands the category *b* if and only if *a* does not contain *b* and the lower of the two categories α, β dominates *b*, where α is the first branching category dominating *a* and β is the first maximal projection dominating *a*.

Because this definition mentions the lower rather than the higher of the two categories α and β, it picks out the intersection of classical c-command (if α is lower) and MAX-command (if β is lower) (The discussion of Rouveret and Vergnaud's relation in Aoun and Sportiche (1982, pp 213–214) contains vitiating errors and should not be consulted in this connection.) Given the intersection theorem, we do not have to do any investigation to determine whether Rouveret and Vergnaud's new relation is a command relation; we know immediately that it is, because the theorem entails that the set of command relations is closed under intersection. Moreover, we know what the generating property of the new relation is: it is $P_5 \cup P_4$, the union of the property 'is a branching node' and the property 'is labeled by a maximal projection'.

2.6 IDC-*command*

We now consider the relation called IDC-command in Pullum (1986). This is given in the literature by Langacker (1969, p. 174), where it occurs, oddly, as a misquotation of the inverse of Klima's 'in construction with'. It is identical to (the intent of) the definition of 'command' on metrical trees given by Kiparsky (1977, p 205), the definition of 'minimal c-command' given by Emonds (1985, p. 68), and the relation that must hold between two nodes (in one direction or the other) if they are to be affected by a 'strictly local transformation' in the sense of Chomsky (1965, p 215, n 18) or a 'local transformation' in the sense of Emonds (1976, p 242). The informal definition is straightforward·

(21) *a* IDC-commands *b* if and only if *a*'s mother dominates *b*.

By analogy with our treatment of the previous examples, we would choose a generator property that includes all mother nodes, that is, all nodes that immediately dominate some other node. This will include all nodes except for the terminals But since including the terminal nodes in the generator property does not affect the induced command relation, we can simply choose the maximally large property of nodes for a generator. In other words, we replace 'branching node' by 'node' in the definition of classical c-command. This works precisely because the restriction to the first proper dominator is built into the definition of a command relation.

DEFINITION 11. IDC-command is the command relation C_{P_6} where P_6 is given by

(22) $P_6 = N$

The property *P* here is the entire set *N* of nodes in the tree, i.e., the most general of all properties over *N*. By the intersection theorem, we know that increasing the size of the generator property restricts the size of the induced command relation. Thus, since IDC-command is generated by the maximally large property, it is the most restrictive (smallest) of all command relations. Another way to put this is that a pair $\langle a, b\rangle$ is in the IDC-command relation if and only if $\langle a, b\rangle$ is in every command relation.

3. Mate relations

Formal grammars often refer to co-membership in some syntactic domain. The most famous example is the clausemate relation. Two nodes are said to be CLAUSEMATES if and only if each is contained in every clause that contains the other

Like 'in construction with', the notion clausemate is an important relation on nodes in a tree, but it is not a command relation (e.g., it fails Descent). However, there is a very simple relationship between clausematehood and S-command the clausemate relation is the intersection of S-command with its inverse. In the case of the binary relations that we are concerned with, intersection with the inverse corresponds to the meaning of reciprocals in English, so that 'S-command each other' is a fair paraphrase for the mate relation generated by S-command. In fact, it is easy to see that on the definitions in Section 1, two nodes are clausemates if and only if they have identical S-command domains

More generally, we define the notion of a MATE RELATION as follows

DEFINITION 12. The MATE RELATION generated by a property P is given by

(23) $$M_P = C_P \cap C_P^{-1} = \{\langle a, b\rangle \mid \mathrm{UB}(a, P) = \mathrm{UB}(b, P)\}$$

In other words, the mate relation generated by a property P is the intersection of the command relation C_P with its inverse C_P^{-1}, which is the same thing as the set of all pairs $\langle a, b\rangle$ such that a and b have the same command domain with respect to P.

In addition to the clausemate relation (i.e., S-mate), MAX-mate is particularly common in definitions of government: see Section 4.

The inverse of S-command is not a command relation; this is shown by an argument exactly analogous to that given in Section 2.5 for 'in construction with' However, this is not because it is impossible for the inverse of a command relation to be a command relation. Consider, for instance, the command relation generated by the empty property. Since the maximally small property gives rise to the maximally large command relation (a corollary of Boundedness), this particular command relation holds of every pair of nodes in the tree. This relation is clearly symmetric, hence its inverse is itself, and its mate relation is itself and therefore is a command relation.

4 GOVERNMENT

The definitions of various command relations in the recent literature (see Aoun and Sportiche (1982, p 213) and Saito (1984, pp 402–403) for a selection of definitions) have very frequently been formulated with a view to use in a definition of government. The statement 'a governs b' normally means, first (and irrelevantly here), that a has bar level 0, i.e. is a lexical category like N, A, V, or P (and, in some works, has the feature +AGR if it is X^0 where X = INFL), and second, relevantly, that in addition a MAX-commands b

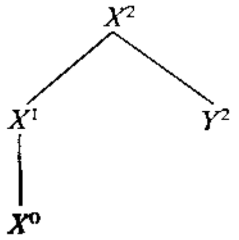
Consider a randomly chosen example, the definition of government in Davis (1987, p 311), which is said to be an adaptation of the definition of government in Aoun and Sportiche (1982), and is stated thus:

(24) a governs b in the configuration $[\cdots b \cdots a \cdots b \cdot\cdot]$ iff
a. $a = X^0$, (+AGR, if X^0 = INFL); and

b. where *P* is a maximal projection, *P* dominates *b* iff *P* dominates *a*

The phrase 'in the configuration $[\cdots b \cdots a \cdots b \cdots]$' is apparently intended to mean 'in a structure where a node *b* occurs either to the left or to the right of a node *a*', and '$a = X^0$' means that the label of node *a* is a category of bar level 0. Context suggests that (24), like the many similar definitions in the literature, is supposed to entail that in (25), X^0 governs Y^2, where categories with bar level 2 label maximal projections.

(25)



But this does not hold. X^0 does not govern Y^2 because Y^2 itself is a maximal projection that dominates Y^2 but does not dominate X^0 (recall that the usual interpretation of dominance is as a reflexive relation).

Davis's definition can be restated, quite informally but nonetheless accurately, as follows:

(26) A node *a* governs a node *b* iff
a. LABEL(*a*) has bar level 0; and
b. if LABEL(*a*) = INFL then LABEL(*a*) includes +AGR; and
c. *a* and *b* are MAX-mates, and
d. neither *a* nor *b* dominates the other.

Here 'includes' in clause (b) is simply set-thoeretic inclusion, assuming categories are sets of feature specifications. Clause (*d*) is needed because otherwise phrases would be defined as governed by their own lexical heads. This is not a stipulation that Davis' definition avoids; she stipulates it implicitly by stating that either *a* precedes *b* or *b* precedes *a* (the 'in the configuration $[\cdots b \cdots a \cdots b \cdots]$' condition).

Formulating the definition this way removes the necessity of any stipulated condition, such as many definitions of government contain, to the effect that *a* must not be separated from *b* by any distinct maximal projection *C*.

Aoun and Lightfoot (1984, p. 467) seek to develop a definition of government that is slightly broader than (26). Although they do not want '*a* governs *b*' to entail that *a* governs everything dominated by *b*, they do want the head of b to be governed Their definition does not achieve this, as noted by Postal and Pullum (1986). (The defense by Lightfoot (1986) is mistaken) An essential fact about the government relation that Aoun and Lightfoot envisage is that it is not simply a command relation with some additional conditions on the commander; it is the intersection of a command relation between a and b with an entirely different relation, best expressed as holding in the other direction, between b and a In fact, the second relation is the composition of H^* with MAX-command, where H is the relation 'is the head of' on which X-bar theory is founded, and '*' is an operator forming the reflexive transitive closure

That is, in order for the Aoun-Lightfoot notion of government to hold between a and b, a must MAX-command b, and b must bear to a the relation given by the composition of H^* with MAX-command.

The refinement to the definition of θ-government given in Chomsky (1986, p. 19) is closely analogous. Basically, the structural requirement is that the θ-governor and the governed node must be sisters Chomsky's modification is to add a disjunct clause, 'or b is the head of a sister of a' to his definition This addition is intended to ensure that heads of θ-governed categories (and heads of heads, and so on) are also θ-governed. In parallel with the MAX-command case above, the relevant stipulation is that b must bear to a the relation given by the composition of H^* with IDC-command

Both the last two examples sugest a general characterization of a government relation as $C \cap (H^* \circ C^{-1})$ where C is a command relation, H is the head-of relation, and '$\circ$' gives the composition of two relations.[3]

5 Generalizing to arbitrary labeled graphs

The properties of command relations as defined in Section 1 depend upon the properties of trees. Many alternative proposals for the graph-theoretic representation of sentence structure have been made (see, e.g., Morin and O'Malley (1969), Sampson (1975), Perlmutter and Postal (1977), Johnson and Postal (1980), McCawley (1982), Ojeda (1987)) Since experimentation with such alternatives is becoming commonplace in recent syntactic (and phonological) theory, it is important for us to determine the extent to which our characterization of command relations depends upon our particular definition of trees, as opposed to other types of graph. In this section we will explore this question by generalizing command

[3] The composition of R_1 and R_2 is $R_1 \circ R_2 = \{(a,b) | \exists x[aR_1x \wedge xR_2b]\}$

relations to the domain of all graphs The results for graphs will then hold for any linguistic framework which involves graphs or representations equivalent to them.

A graph is simply a set of nodes with a relation defined on it. Since we wish to retain the ability to state command relations which are sensitive to the syntactic category of the nodes involved, we must augment the basic notion of a graph to permit labeling. This suggests the following defintion'

DEFINITION 13 A labeled graph G is a 4-tuple $G = \langle N, L, \overset{D}{\geq}, \textsc{label}\rangle$, where

N is a finite set, the nodes of G,
L is a finite set, the labels of G,
$\overset{D}{\geq}$ is a reflexive, transitive relation over N, the dominance relation of G, and
LABEL is a total function from N into L, the labeling function of G.

There is no mention of a precedence relation, and no further restrictions are placed on these graphs. In fact, it is non-standard even to require that the relation defining the graph be reflexive or transitive. But since the relation is intended as a model of dominance, we have taken the liberty of referring only to the transitive closure of the usual basic relation for the sake of convenience.

This definition differs from the earlier definition of trees in failing to stipulate either the Nontangling Condition or the Single Root Condition (the Exclusivity Condition is irrelevant in the absence of a precedence relation). The alternative proposals mentioned above usually place themselves in between the standard tree and this extreme position by weakening or removing one ~~of~~ the other of the Nontangling Condition or the Single Root Condition

Another partially independent requirement commonly placed on treehood is the Single Mother Condition (Sampson (1975)).

DEFINITION 14. A graph $G = \langle N, L, \overset{D}{\geq}, \textsc{label}\rangle$ satisfies the SINGLE MOTHER CONDITION iff

(27) $(aMb \wedge cMb) \rightarrow (a = c)$

Here M is the mother-of relation as defined in (18) The Single Mother Condition guarantees that a node has at most one mother. This is a separate stipulation in Sampson's definition of a tree, but it is a theorem of Wall's definition and therefore of our definition in Section 1. As for the other constraints, we do not require that the Single Mother Constraint hold for graphs

We now consider the results of applying the definition of command relations given in Section 1 to graphs which are not trees: graphs which have, for instance, more than one root, or nodes with more than one mother, or cycles (dominance chains in which a node has a descendant that is also an ancestor)

Surprisingly, most of the results obtained for trees continue to hold. Most importantly, the intersection theorem goes through without modification· union of generating properties corresponds to intersection of command relations, and conversely.

(28)

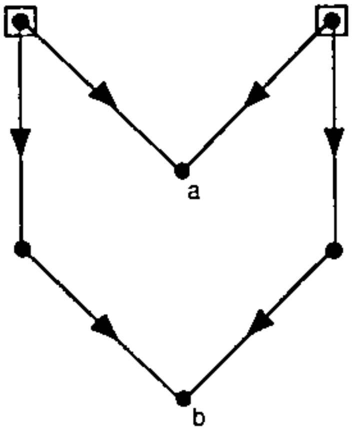
Characterizing Property	*Trees*	*Labeled Graphs*
Intersection	yes	yes
Ambidextrousness	yes	yes
Boundedness	yes	yes
Constituency	yes	no
Descent	yes	yes
Embeddability	yes	no
Fairness	yes	yes

The only two characteristics of command relations which fail to generalize to labeled graphs are Constituency and Embeddability

5.1 *Constituency and the Connected Ancestor Condition*

Constituency fails on some graphs which fail to satisfy the Single Mother Condition, as in (29)

(29)



In (29), *a* IDC-commands only *a* and *b*. There is no node in (29) which dominates exactly *a*'s command domain; thus Constituency fails.

Just how much of the restrictiveness provided by Constituency must we give up when we move to arbitrary labeled graphs? Descent, in effect, is a weaker but more general form of Constituency. Recall that Descent says that if *a* commands *b*, *a* commands all of *b*'s descendants. Both theorems constrain the degree to which a command domain may be fragmented. Constituency makes the most restrictive statement conceivable: a command domain must form a single coherent constituent. Descent, by comparison, requires only that command relations are closed under composition with dominance. That is, if *C* is a command relation, Descent guarantees that *C* composed with $\overset{D}{\geq}$ is equal to *C*.

The constraints on trees given in Section 1 entail Constituency. But there is a more general class of graphs than the set of trees which supports Constituency, viz., those graphs which satisfy the Connected Ancestor Condition.

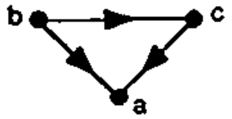
DEFINITION 15. A labeled graph $G = \langle N, L, \overset{D}{\geq}, \textsc{label}\rangle$ satisfies the CONNECTED ANCESTOR CONDITION iff

(30) $(b \overset{D}{\geq} a \wedge c \overset{D}{\geq} a) \rightarrow (b \overset{D}{\geq} c \vee c \overset{D}{\geq} b)$

This condition requires there to be a dominance path connecting any pair of nodes that share a descendant. The proof that the Connected Ancestor Condition guarantees Constituency appears in the appendix in Theorem 7.

Note that the Connected Ancestor Condition is more general than the Single Mother Condition. Consider (31)

(31)



Here *a* has two mothers, violating the Single Mother Condition, but the Connected Ancestor Condition is satisfied, since *b* and *c* are in a dominance relationship. However, the Single Mother Condition does entail the Connected Ancestor Condition, which means that Wall's (1972) definition of tree given in Section 1 also entails the Connected Ancestor Condition. Hence command relations on trees obey Constituency.

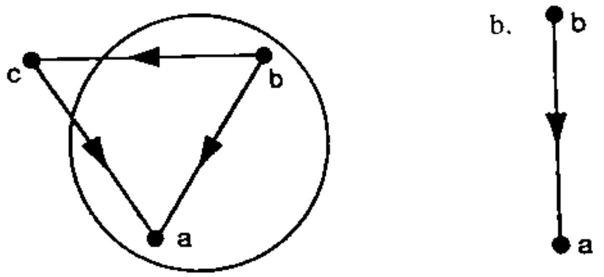
5.2. *Embeddability*

We now turn to Embeddability, the second of the two characteristics of command relations which fail when we move to arbitrary graphs. We must first present a definition of what it means to embed one graph within another

DEFINITION 18 A graph $G_1 = \langle N_1, L_1, \overset{D}{\geqslant}_1, \text{LABEL}_1\rangle$ is said to be EMBEDDED within a graph $G_2 = \langle N_2, L_2, \overset{D}{\geqslant}_2, \text{LABEL}_2\rangle$ iff $N_1 \subseteq N_2$, $L_1 \subseteq L_2$, $\overset{D}{\geqslant}_1 \subseteq \overset{D}{\geqslant}_2$, and $\text{LABEL}_1 \subseteq \text{LABEL}_2$.

What additional conditions must be satisfied in order for Embeddability to hold? Just as for Constituency, the Connected Ancestor Condition is a necessary condition for Embeddability to succeed. But we must also place two restrictions on how a sensible embedding should behave. Consider the graphs in (32)

(32)a



The subgraph embedded in (32a) that lies entirely within the circle is redrawn in (32b) Assume that of the three nodes a, b, and c, only c is a member of the generating property P. It is easy to see that a commands b in (32b). But in (32a), c is an upper bound for a which fails to dominate b, so a fails to command b in the larger graph This is despite the fact that (32a) obeys the Connected Ancestor Condition

The embedding depicted in (32) is peculiar, since b has a child in (32a) that it does not have in (32b) (namely, c) In effect, the embedding operation has inserted new material into the embedded subgraph. We propose to exclude such unnatural embeddings by adopting the Integrity condition on embedding.

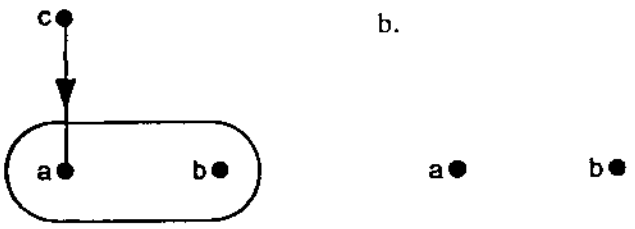
DEFINITION 16. An embedding of $G_1 = \langle N_1, L_1, \overset{D}{\geqslant}_1, \text{LABEL}_1\rangle$ within $G_2 = \langle N_2, L_2, \overset{D}{\geqslant}_2, \text{LABEL}_2\rangle$ obeys INTEGRITY iff

(33) $(a \in N_1 \wedge a \overset{D}{\geqslant}_2 b) \rightarrow a \overset{D}{\geqslant}_1 b$

Notice that $a \overset{D}{\geqslant}_1 b$ entails that $b \in N_1$. Thus Definition 16 says that if a node is part of the subgraph after embedding, it is part of the subgraph in isolation.

But Integrity alone is not sufficient to guarantee embeddability

(34)a



b.

In (34b), *a* (trivially) IDC-commands *b*, but in (34a), it does not, assuming that *c* satisfies the generating property We therefore also require that an embedded subgraph be rooted

DEFINITION 17. A graph $G = \langle N, L, \overset{D}{\geqslant}, \text{LABEL}\rangle$ IS ROOTED iff

(35) $\exists x \forall a[x \overset{D}{\geqslant} a]$

Informally, a graph is rooted if and only if there is some node which dominates the entire graph.

In sum, in order to guarantee Embeddability, we must assume that the graphs over which command relations are to be defined must satisfy the Connected Ancestor Condition, the subgraph to be embedded must be rooted, and the embedding must obey Integrity. These stipulations are natural enough· the Connected Ancestor Condition is much more general than the full complement of requirements usually built in to the definition of a tree, and as for the constraints on embedding, they will be desirable for the same reasons to the same extent that Embeddability seems like a natural and desirable constraint on the behavior of command relations

We conclude that the intersection theorem and the properties of Ambidextrousness, Boundedness, Descent, and Fairness (though see Section 6) constitute properties of command relations quite independently of whether the graphs we consider are trees, but Constituency and Embeddability will be in evidence only when command relations are defined on graphs which obey the Connected Ancestor Condition. Since trees as defined in Section 1 entail the Connected Ancestor Condition, of course, all six of the characterizing properties hold for command relations defined over trees

Having made these observations, we now return to discussing command relations over trees, for reasons of familiarity and (for the present) wider applicability.

6. Relations as generators

Up to this point we have generated command relations from properties of nodes We have also investigated the consequences of generating command relations from relations between nodes rather than from properties (the method for describing command relations employed by Pullum (1986)), and we briefly summarize our results here.

Under the definition of an upper bound given in (1), each generator property can be thought of as specifying a set of upper bound candidates. A candidate is an upper bound for a particular commander if and only if it properly dominates that commander. The proposed modification omits mention of any property, but requires instead that all upper bound candidates must stand in a certain relation to the commander. In effect, the set of candidates can vary arbitrarily depending on the commander.

In order to use relations as generators, we do not need to modify the definition of a command relation We simply define upper bound with respect to relations instead of properties.

DEFINITION 19. The set of UPPER BOUNDS for *a* with respect to a relation *R*, written UB(*a*. *R*). is given by

(36) $\mathrm{UB}(a. R) = \{b \mid b \overset{D}{\cancel{>}} a \wedge bRa\}$ $k \; \boxed{\overset{D}{>}}$

This definition is exactly parallel to Definition 2 Thus *b* is an upper bound for *a* with respect to *R* if and only if it properly dominates *a* and also stands in the *R* relation to *a*

Clearly this increases the expressive power of the framework for specifying command relations All previously definable command relations are still definable, since a command relation generated by a property *P* can be generated instead by the relation $\{\cancel{\langle a, b\rangle} \mid P(a)\}$ But the set of command $k \; \langle b, a \rangle$ relations is expanded in two ways: (i) Fairness no longer holds (see Theorem 6 and preceding discussion in the appendix). and (ii) the set of command relations becomes closed under union, so that for any two command relations. there is some relation over nodes which generates their union (see Theorem 9 and preceding discussion in the appendix)

Fairness fails for command relations generated from relations precisely because upper bounds vary from node to node independently. It is possible, for instance, for the command domains of NP nodes to coincide with the command domains expected for c-command at the same time that all other nodes have command domains appropriate for MAX-command Thus it is possible for a node to have a larger command domain than its mother, this is not possible when generating from properties.

The idea of using relations as generators is not motivated simply by mathematical posssibility. We are aware of at least two command relations in the syntactic literature which can be characterized only by using relations rather than properties as generators One is the relation that is called c-command in Reinhart (1976; 1981, p 612; 1983, p. 23).

(37) Node a c(onstituent)-commands node b iff the branching node x_1 most immediately dominating a either dominates b or is immediately dominated by a node x_2 which dominates b, and x_2 is of the same category type as x_1

Despite the name Reinhart gives it, this is very different from classical c-command. In effect, (37) allows the minimal upper bound to be replaced by another node, one node closer to the root than would be possible under classical c-command depending on a property of the commander (namely, what the label is on the branching node that most immediately dominates it). There is a way of defining this using relations as generators (which we omit here, as it is complex and unrevealing), but there is no way to characterizing it using properties.

Similarly, we believe that Chomsky's (1986) definition of government in terms of 'exclusion' and 'barrierhood' requires recourse to command relations generated from relations. Chomsky's government relation seems to be the intersection of a command relation with the inverse of a second command relation whose definition is based on barriers (we show in Section 4 above that recent definitions of government are generally intersections of command relations with inverses of somewhat more complex relations). The concept 'is a barrier for' seems to be a particular instance of our general notion 'is an upper bound for', and Chomsky explicitly observes that his definition of 'barrier' is 'relational a category α will be a barrier for β for certain choices of β but not for others' (1986, p. 12). This entails that the command relation induced by barrierhood does not satisfy Fairness, since the set of upper bound candidates varies depending on the choice of commander.

7. Summary and conclusion

The schema for defining command relations given in Section 1 has two main virtues. First, it facilitates the presentation of convenient yet precise and reliable working definitions of particular command relations By way of demonstration, Section 2 reconstructs a wide range of command relations proposed in the literature merely by specifying a simple property of nodes in a tree. In each case the general schema guarantees specific reasonable behavior in the limiting cases which the less formal statements of the relations found in the literature leave undetermined. Furthermore, we suggest that other important classes of relations on trees such as mate relations (Section 3) and government relations (Section 4) are best understood as complex objects built from command relations.

Second, our approach provides a foundation for a mathematical characterization of command relations. We show that the intersection of two command relations is always a command relation (Theorem 1). Moreover, there is a smallest, maximally restrictive command relation, namely, IDC-command. In addition, we consider two dimensions along which a theory of command relations can vary: whether command relations are defined over trees in the sense of Wall (1972), or, more generally, over arbitrary labeled graphs (Section 5), and whether command relations are generated from properties of nodes or from relations over nodes, as required in order to encompass certain more complicated relations (Section 6). No matter what the values of these two metatheoretic parameters, command relations all observe the intersection theorem, so that requiring a pair of nodes to stand in two different command relations simultaneously is always equivalent to requiring that they stand in a single command relation (the one derived from the first two by intersection). Command relations also respect the Ambidextrousness, Boundedness, and Descent conditions. When they are defined over rooted graphs which obey the Connected Ancestor Condition (as do trees), they also exhibit Constituency and Embeddability. Furthermore, when generated from properties rather than relations they satisfy the Fairness criterion. The set of command relations generated from relations is closed under set union, but the set of command relations generated from properties is not.

APPENDIX: PROOFS OF THEOREMS

This section contains proofs of the results mentioned in the previous sections. Specifically, we show that the set of all command relations is closed under intersection; that IDC-command is the smallest command relation; that command relations as defined in Section 1 exhibit Ambidextrousness, Boundedness, Constituency, Descent, and Embeddability, that command relations generated from properties satisfy the Fairness criterion; and that command relations generated from relations are closed under set union.

We will prove the five fully general theorems assuming that command relations are defined on arbitrary graphs, and that they are to be generated from properties rather than relations. There are proofs similar to the one presented here for the other settings of these two parameters. but we do not include them here.

We adopt the definitions set out in Section 1. In particular. let $G = \langle N, L, \overset{D}{\geqslant}, \text{LABEL}\rangle$ be a labeled graph and let P and Q be properties over N which induce the command relations C_P and C_Q. Since the choice of P and Q is arbitrary, theorems true of them will hold for all command relations. Similarly, let a, b, c, and d be arbitrary nodes in N. Since the choice of these nodes is arbitrary, boolean predicates which hold of these nodes will hold for all choices of nodes from N

THEOREM 1 $C_P \cap C_Q = C_{P \cup Q}$ Intersection over command relations corresponds to union over the properties that induce them

Proof. Assume there is some $\langle a, b\rangle \in C_P \cap C_Q$ such that $\langle a, b\rangle \notin C_{P \cup Q}$. Then there exists some $c \in P \cup Q$ that properly dominates a but does not dominate b, by the definition of a command relation If $c \in P$, then a does not P-command b, contrary to assumption; but $c \in Q$ leads to a similar contradiction Therefore $C_P \cap C_Q \subseteq C_{P \cup Q}$. In the other direction, assume there is some $\langle a, b\rangle \in C_{P \cup Q}$ such that $\langle a, b\rangle \notin C_P \cap C_Q$. Then there must be some upper bound c in either P or Q that dominates a but not b But then $c \in P \cup Q$, and $\langle a, b\rangle \notin C_{P \cup Q}$, contrary to assumption. Therefore $C_P \cap C_Q \supseteq C_{P \cup Q}$, and the theorem follows immediately. ☒

THEOREM 2. IDC-command $= \cap\{C | C$ is a command relation$\}$. IDC-command is the intersection over all command relations; it is the smallest. most restrictive command relation posssible

Proof Recall that IDC-command is generated from the maximal property, the property that contains every node – that is. N Then. by Theorem 1, IDC-command $\cap$ $C_P = C_N \cap C_P = C_{N \cup P} = C_N =$ IDC-command. Hence IDC-command $\subseteq C_P$. Since the choice of P is arbitrary. IDC-command is a subset of every command relation Therefore it is a subset of the intersection over all command relations. Furthermore. since IDC-command is itself a command relation, it must contain the intersection of all command relations The theorem follows immediately. ■

THEOREM 3 Command relations are ambidextrous. Linear precedence is irrelevant to command

Proof. The ambidextrousness of command relations follows from Definition 3. since it fails to mention the precedence relation. ■

Theorem 3 means that whether two nodes stand in a command relation will never depend solely on their linear order

The next theorem gives a special property of root nodes A root node, of course, is any node which dominates every node in N.

THEOREM 4. Boundedness. If r is a root node, then $C_P = C_{P \cup \{r\}}$. A command relation is not altered by the addition of a root to its generating property.

Proof. We have $C_P \supseteq C_{P \cup \{r\}}$ by Theorem 1. In the other direction, assume there exists some $\langle a, b \rangle \in C_P$ such that $\langle a, b \rangle \notin C_{P \cup \{r\}}$ Then there exists some upper bound $c \in P \cup \{r\}$ that is not in P and which properly dominates a but does not dominate b Clearly, r is the only candidate But r is a root node; therefore r dominates b, and we have a contradiction ■

THEOREM 5. Descent: $(aCb \wedge bDc) \rightarrow aCc$. The descendants of a commanded node are also commanded

Proof. Assume aCb and bDc. We wish to show that aCc. Assume otherwise Then there is some upper bound d for a which fails to dominate c. But d dominates b, since a commands b. hence d also dominates c, by the transitivity of dominance, and we have a contradiction Therefore a must command c. ■

As discussed in Section 6, we must assume that command relations are generated only from properties rather than from relations before we can prove Theorem 6 (though command relations generated from relations can accidentally satisfy Fairness). The formal statement of Fairness is somewhat difficult to penetrate. It is easier to understand when expressed in terms of upper bounds, but the formal statement must refer only to the command relation in question, since Fairness is a property of relations independent of the means by which the relation is specified. Roughly, Fairness characterizes a class of situations in which information concerning the command domain of one node can be deduced by examining the command domain of another node. That is, if a command relation is fair, then the command domains of all nodes are shaped by the same impartial generating property In contrast, when a command relation is generated from an arbitrary relation, there is no guarantee that the command domains of different nodes will have any connection at all

THEOREM 6 Fairness: $(aCb \wedge bCc \wedge \neg aCc) \rightarrow (aCd \rightarrow b \overset{D}{\geqq} d)$ An upper bound is an upper bound for every node it properly dominates

Proof. Assume $aCb \wedge bCc \wedge \neg aCc$. We wish to show that b dominates every node that a commands. Since $\neg aCc$, there must be some upper bound for a that does not dominate c. Since aCb, this upper bound x must dominate b Furthermore, since b commands c, either x dominates c or $x = b$. But x does not dominate c, or aCc would hold. Hence $x = b$, i e., b is an upper bound for a. Therefore b dominates every node that a commands ■

REMARK Fairness can be strengthened if we assume that the underlying graphs are rooted and obey the Connected Ancestor Condition. Specifically, we have the condition $(aCb \wedge bCc \wedge \neg aCc) \rightarrow (aCd \leftrightarrow bDd)$.

The preceding theorems are true for command relations defined on arbitrary (transitive) graphs. The remaining theorems each require additional stipulations. Specifically, we must assume that the graphs in question satisfy the Connected Ancestor Condition, and that they are rooted. In addition, for Embeddability, we must assume the restrictions on embedding discussed in Section 5.

Since our definiton of a tree entails rootedness as well as the Single Mother Condition, and since the Single Mother Condition entails the Connected Ancestor Condition, the remaining three theorems will hold for any command relation defined on a tree.

THEOREM 7 Constituency: $\exists x[aCb \leftrightarrow x \overset{D}{\geqslant} b\}$ Command domains correspond to constituents.

Proof We will show the Connected Ancestor Condition entails that a node has at most one minimal upper bound Assume othewise: let b and c be minimal upper bounds for a such that $b \neq c$. Since b and c share a descendant, either $b \overset{D}{\geqslant} c$ or $c \overset{D}{\geqslant} b$ by the Connected Ancestor Condition. If $b \not\overset{D}{\geqslant} c$, b is not a minimal upper bound; but if $c \not\overset{D}{\geqslant} b$, then c is not minimal Therefore a node a has at most one minimal upper bound. Since G is rooted, there is some node r which dominates the entire graph. If $a = r$, the theorem is trivial. Assume, therefore, that r properly dominates a. By Boundedness, we can assume that r is an upper bound for a. Therefore a has at least one minimal upper bound b, and hence exactly one minimal upper bound By the definition of a command relation, a commands only those nodes dominated by b; by Descent, a commands all of the descendants of b ■

For proving Embeddability, we assume that the graph $G_1 = \langle N_1, L_1, \overset{D}{\geqslant}_1$, LABEL$_1\rangle$ is rooted, that the embedding of G_1 within the graph $G_2 = \langle N_2, L_2, \overset{D}{\geqslant}_2$, LABEL$_2\rangle$ obeys Integrity, and that G_2 satisfies the Connected Ancestor Condition (see Section 5)

THEOREM 8. Embedding preserves commands relations. A node a commands a node b in G_1 if and only if a commands b in G_2.

Proof Assume a commands b in G_1. We wish to show that a also commands b in G_2. Assume otherwise. Then there must be some node c in N_2 which properly dominates a but which does not dominate b By the fact that G_1 is rooted, there is some node d in N_1 which dominates both a and b By the fact that G_2 obeys the Connected Ancestor Condition, either $d \overset{D}{\geqslant}_2 c$ or $c \overset{D}{\geqslant}_2 d$ If $d \overset{D}{\geqslant}_2 c$, Integrity entails that $c \in N_2$; but then a would fail to command b in G_1, contrary to assumption. But if $c \overset{D}{\geqslant}_2 d$, then $c \overset{D}{\geqslant}_2 b$ by the transitivity of dominance. Since we have arrived at a contradiction. a must command b in G_2. In the other direction. assume aCb in G_2. We wish to show that aCb in G_1 as well. Assume otherwise. Then there must be some c in N_1 which dominates a but not b. In order for aCb in G_2 we must have $c \overset{D}{\geqslant}_2 b$. But Integrity entails that $c \overset{D}{\geqslant}_1 b$, and we have a contradiction ■

We now set out to prove that the set of all command relations when generated from relations over nodes is closed under set ~~intersection~~ union At this point it will be convenient to define the notion of a maximal generator. Intuitively, a generator relation is maximal if it is impossible to increase its membership without altering the command relation it generates. To begin to make this notion more precise, notice that the definition of a command relation induces an equivalence relation of the set of relations over N.

DEFINITION 20 Two relations R and S are said to be COMMAND-EQUIVALENT. written $R \sim S$, if $C_R = C_S$.

Two relations are command-equivalent if and only if the command relations they induce are equal. This does not imply that the two relations are identical For instance, Boundedness entails that any relation is command-equivalent to the union of itself and any set of pairs $\langle a, b\rangle$ where a is a root node, which gives rise to a family of distinct but command-equivalent relations

DEFINITION 21. Let R be a relation. Then $\widehat{R}$, the maximal generator for C_R, is given by

(38) $\widehat{R} = \bigcup \{S | S \supseteq R \wedge S \sim R\}$

Informally, $\widehat{R}$ is the largest relation that is equivalent to R under $\sim$. Notice that $\widehat{R} \supseteq R$ and $\widehat{R} \sim R$ It is easy to see if c dominates some upper bound b. $\langle c, a\rangle \in \widehat{R}$, since non-minimal upper bounds do not affect the value of the command relation.

THEOREM 9 $C_R \cup C_S = C_{\hat{R} \cap \hat{S}}$. Union over command relations corresponds to intersection over the maximal generators that induce them.

Proof Let $\langle a, b\rangle \in C_R \cup C_S$. Then either $\langle a, b\rangle \in C_R$ or $\langle a, b\rangle \in C_S$. Assume $\langle a, b\rangle \in C_R$ We wish to show that $\langle a, b\rangle \in C_{\hat{R} \cap \hat{S}}$. Assume otherwise; then $\langle a, b\rangle \notin C_{\hat{R} \cap \hat{S}}$, and there must be some c such that $\langle c, a\rangle \in \widehat{R} \cap \widehat{S}$ and c properly dominates a but c does not dominate b It follows that

$\langle c, a\rangle \in \widehat{R}$, and $\langle a, b\rangle \notin C_{\hat{R}}$, contrary to assumption. But $\widehat{R} \sim R$, so $C_R = C_{\hat{R}}$, and $\langle a, b\rangle \notin C_R$. We arrive at the same contradiction if we assume $\langle a, b\rangle \in C_S$. Therefore $C_R \cup C_S \subseteq C_{\hat{R}\cap\hat{S}}$. In the other direction, assume $\langle a, b\rangle \in C_{R\cap S}$ but $\langle a, b\rangle \notin C_R \cup C_S$. Then there are nodes c and d which both properly dominate a such that $\langle c, a\rangle \in R$ and $\langle d, a\rangle \in S$, and neither c nor d dominates b. Since c and d share a descendant and G obeys the Connected Ancestor Condition, either c dominates d or vice-versa. Assume that c dominates d. Then $\langle c, a\rangle \in \widehat{S}$, since c is a non-minimal upper bound for a From the fact that $\widehat{R} \supseteq R$ we have also that $\langle c, a\rangle \in \widehat{R}$. Therefore $\langle c, a\rangle \in \widehat{R} \cap \widehat{S}$, which entails that $\langle a, b\rangle \notin C_{\hat{R}\cap\hat{S}}$, contrary to assumption. If we assume that d dominates c we arrive at a similar contradiction Therefore $C_R \cup C_S \supseteq C_{\hat{R}\cup\hat{S}}$, which completes the proof ■

Theorem 9 establishes that if we generate command relations from relations on nodes, the set of all command relations is a lattice, with set intersection for a meet operation and set union for a join. The command relation generated by the empty relation ($C_{\{\}} = N \times N$) is the top element, and IDC-command, as shown in Theorem 2, is the bottom.[4]

References

Aoun, Joseph and David W Lightfoot 1984 'Government and Contraction', *Linguistic Inquiry* 15 465–509

Aoun, J and Sportiche D 1982 'On the Formal Theory of Government', *Linguistic Review* 2, 211–236

Banfield, A 1973, 'Grammar of Quotation Free Indirect Style, and Implications for a Theory of Narrative', *Foundations of Language* 10, 1–39

Berwick, R C and Wexler K 1982, 'Parsing Efficiency, Binding, and c-command' in D P Flickinger, M Macken, and N Wiegand (eds), *Proceedings of the First West Coast Conference on Formal Linguistics*, Stanford University, Stanford, California, pp 29–34

Chomsky, Noam 1955, *The Logical Structure of Linguistic Theory*, manuscript published by Plenum Press, New York 1975, republished by the University of Chicago Press, Chicago 1985

Chomsky, Noam 1965, *Aspects of the Theory of Syntax*, MIT Press, Cambridge, Massachusetts

Chomsky, Noam 1981, *Lectures on Government and Binding*, Foris, Dordrecht Holland

Chomsky, Noam 1986, *Barriers*, Linguistic Inquiry Monographs 13, MIT Press, Cambridge, Massachusetts

Davis Lori J 1987 'Remarks on Government and Proper Government', *Linguistic Inquiry* 18 311–321

[4] We believed at the time we presented an earlier version of this paper at the 1987 Annual Meeting of the Linguistic Society of America that the set of command relations formed the domain of a boolean algebra We have since discovered that the complement operation we had in mind does not obey the necessary laws We leave open the question of whether there is a boolean algebra on the set of command relations

Emonds, Joseph E. 1976, *A Transformational Approach to English Syntax*, Academic Press, New York.

Emonds, Joseph E. 1985, 'A Unified Theory of Syntactic Categories', *Studies in Generative Grammar* **19**, Foris, Dordrecht, Holland.

Gazdar, Gerald, Geoffrey K. Pullum, Robert Carpenter, Ewan Klein, Thomas E. Hukari, and Robert D. Levine 1988, 'Category Structures', *Computational Linguistics* **14**, 1–19.

Huck, Geoffrey J. and Almerindo E. Ojeda (eds.) 1987, *Discontinuous Constituency*, Syntax and Semantics **20**, Academic Press, New York.

Jackendoff, Ray S. 1972, *Semantics in Generative Grammar*, MIT Press, Cambridge, Massachusetts.

Jackendoff, Ray S. 1977, *X-bar Syntax: A Study of Phrase Structure*, MIT Press, Cambridge, Massachusetts.

Johnson, David E. and Paul M. Postal 1980, *Arc Pair Grammar*, Princeton University Press, Princeton, New Jersey.

Kiparsky, Paul 1977, 'The Rhythmic Structure of English Verse', *Linguistic Inquiry* **8**, 189–247.

Klima, Edward 1964, 'Negation in English', in J. Fodor and J. J. Katz (eds), *The Structure of Language*, Prentice Hall, Englewood Cliffs, pp. 246–323.

Kornai, András and Geoffrey K. Pullum 1990, 'The X-bar Theory of Phrase Structure', *Language* **66**, 1.

Langacker, Ronald W. 1969, 'On Pronominalization and the Chain of Command', In D. Reibel and S. Schane (eds.), *Modern Studies in English*, Prentice-Hall, Englewood Cliffs, New Jersey, pp. 160–186.

Lasnik, Howard 1976, 'Remarks on Coreference', *Linguistic Analysis* **2**, 1–22.

Lightfoot, David 1986, 'A Brief Response', *Linguistic Inquiry* **17**, 111–112.

McCawley, James D. 1982, 'Parentheticals and Discontinuous Constituent Structure', *Linguistic Inquiry* **13**, 91–106.

McCawley, James D. 1984, 'Anaphora and Notions of Command', in Claudia Brugman et al. (eds.), *Proceedings of the Tenth Annual Meeting of the Berkeley Linguistics Society*, pp. 220–232, University of California, Berkeley.

Morin, Yves Charles and Michael H. O'Malley 1969, 'Multi-rooted Vines in Semantic Representation', in *Papers from the Fifth Regional Meeting*, Chicago Linguistic Society, Chicago, Illinois, pp. 178–185.

Ojeda, Almerindo E. 1987, 'Discontinuity, Multidominance, and Unbounded Dependency in Generalized Phrase Structure Grammar: Some Preliminaries', in Huck and Ojeda (1987), pp. 257–282.

Perlmutter, David M. and Paul M. Postal 1977, 'Toward a Universal Characterization of Passivization', in K. Whistler et al. (eds), *Proceedings of the Third Annual Meeting of the Berkeley Linguistics Society*, Berkeley, California, pp. 394–417.

Postal, Paul M. and Geoffrey K. Pullum 1986, 'Misgovernment', *Linguistic Inquiry* **11**, 104–110.

Pullum, Geoffrey 1985, 'Assuming Some Version of X-bar Theory', *CLS* **21**, 323–353.

Pullum, Geoffrey 1986, 'On the Relations of IDC-command and Government', *West Coast Conference on Formal Linguistics* **5**, 192–206.

Reinhart, Tanya 1974, 'Syntax and Conference', *Proceedings of NELS 5*, Graduate Linguistic Student Association, University of Massachusetts, Amherst, pp. 92–105.

Reinhart, Tanya 1976, *The Syntactic Domain of Anaphora*, Doctoral dissertation, MIT, Cambridge, Massachusetts.

Reinhart, Tanya 1981, 'Definite NP Anaphora and C-Command Domains', *Linguistic Inquiry* **12**, 605–635.

Reinhart, Tanya 1983, *Anaphora and Semantic Interpretation*, University of Chicago Press, Chicago, Illinois.

Richardson, J F and Robert Chametzky: 1985, 'A String Based Redefinition of c-command', *Proceedings of NELS 15*, Graduate Linguistic Student Association, University of Massachusetts, Amherst, pp 332–361

Rouveret, Alain and Jean-Roger Vergnaud 1980, 'Specifying Reference to the Subject French Causatives and Conditions on Representations', *Linguistic Inquiry* **11**, 97–202

Saito, Mamoru 1984 'On the Definition of c-command and Government', *NELS* **15**, 402–417, Graduate Linguistic Student Association, University of Massachusetts, Amherst

Sampson, Geoffrey 1975, 'The Single Mother Condition', *Journal of Linguistics* **11**, 1–11

Wall Robert· 1972 *Introduction to Mathematical Linguistics*, Prentice Hall Englewood Cliffs, New Jersey

Board of Studies in Linguistics
University of California, Santa Cruz
Santa Cruz, CA 95064
U.S.A.