BARE PHRASE STRUCTURE, LCA AND LINEARIZATION

http://vivaldi.sfs.nphil.uni-tuebingen.de/%7Ennsle01/SynVie.htm

1. STRUCTURE VS. PRECEDENCE

Syntactic principles are usually defined in such a way that they refer to structure (dominance and c-command), but not to order. If \( \alpha \) in (1) is e.g. in a designated, linguistically significant, relation with \( \beta \) - say, binding - then it does not matter whether \( \alpha \) precedes \( \beta \), or v.v.:

\[
(1) \quad \begin{array}{ll}
\text{a. } & \alpha \overset{X}{\longrightarrow} \beta \\
\text{b. } & \beta \overset{X}{\longrightarrow} \alpha 
\end{array}
\]

The irrelevance of ordering can be expressed in terms of the Dominance Hypothesis in (2).

(2) Dominance Hypothesis

Syntactic principles refer to c-command/dominance, but not to precedence relations.

The Dominance Hypothesis has been central in most attempts to derive syntactic locality:

1.1. LOCALITY

Movement is subject to syntactic locality constraints, descriptively grouped in terms of island conditions (Ross 1967). Islands come in two flavors:

Strong islands (Complex NP Condition in (3); subject/adjunct condition) result in strong violations and arise from the combination of certain nodes which are in a dominance relation (e.g. if CP is immediately dominated by IP, NP or VP).

\[
(3) \quad \text{Which book did John meet the [NP man [CP who wrote t1]]}
\]

combination of (upper segment of) NP and relative clause dominates the trace

Weak islands (wh-/factive/extraposition/negative islands) lead to mild unacceptability. Their signature is a c-commanding intervener between the head and the tail of the chain. These cases fall under Relativized Minimality (RM; Rizzi 1990; Cinque 1990).

\[
(4) \quad \text{Which book did John wonder [CP whether/who wrote t1?]}
\]

\( \text{wh-operator in SpecCP c-commands the trace} \)

The traditional perspective on how to define locality refers to dominance/c-command relations:

(5) Locality and dominance

a. Locality is defined in terms of dominance (strong islands) and c-command (weak islands).
b. c-command is defined in terms of dominance (containment).
c. \( \Rightarrow \) Locality is derived from dominance.
Recent studies challenge the Dominance Hypothesis, and attempt to derive (at least parts of) the restrictions on movement from precedence (Williams 1998, Fox & Pesetsky 2005; a.o). In what follows (see Handout #6 & #7), we will explore a particular implementation of (6) and its empirical consequences for the analysis of ellipsis phenomena.

(6)  
**Linearization Hypothesis**  
Syntactic locality conditions fall out from the way terminals are linearized, and not (only) from structural properties of the tree (i.e. notions such as c-command and/or dominance).

1.2. **Linearization**

A syntactic tree can be likened to a mobile, whose root is fixed but whose branches swing freely (Uriagereka 1999). By the point at which the derivation is transferred to PF, an ordering of the terminals has to be specified by mapping the tree onto a one-dimensional sequence that can be phonetically interpreted. In (1), for instance, the two-dimensional circle that results from letting $\alpha$ and $\beta$ rotate freely needs to be mapped to a one-dimensional string. Given that this information cannot come from any other source than from the lexicon and properties of the syntactic derivation (which are possibly restricted by ‘interface readability’), and given that the lexicon is inherently unordered, it follows that the tree somehow must also contain information about order, as expressed by corollary (7).

(7)  
**Corollary**: Trees encode dominance and precedence relations.  
**Question**: How can linear order be derived from the information encoded in a tree?

There are attempts to derive the *order* of terminals from hierarchical structure, more precisely the concept of asymmetric c-command (Kayne 1994; see section 3 for details).

(8)  
**Word order and dominance**
  a. *Word order* is defined in terms of asymmetric c-command  
  b. c-command is defined in terms of dominance (or containment).  
  c. $\Rightarrow$ *Word order* is derived from dominance.

Note that here, dominance is again - see (2) - the primary notion, from which precedence is derived. Thus, recent developments hint in two different directions. Locality conditions, which were previously thought to implicate structural conditions only, are reshaped to involve reference to precedence. And precedence, which was thought to be a primitive, has been reduced to asymmetric c-command. Does this mean that one of the notions can be defined in terms of the other? No. But it has become possible to (i) identify new types of restrictions, and (ii) to relocate some phenomena (in particular islandhood) to different domains.

Before considering how these systems work, some background assumptions need to be spelled out. More specifically, the discussion will proceed to the following topics:

- Bare Phrase Structure vs. X’-theory  
- The LCA as a mapping operation from 2-dimensional trees to 1-dimensional PF-objects  
- The LCA and Bare Phrase Structure  
- Consequences of the LCA for the treatment of movement (copies and traces)  
- Consequences of the LCA for the analysis of nominal modifiers (empirical limitations)
2. BARE PHRASE STRUCTURE

Goal: Derive properties of X'-theory (Chomsky 1994 and later)

In Bare Phrase Structure (BPS; Chomsky 1995), Merge is the basic structure building operation. Merge combines two objects $\alpha$ and $\beta$ into a set, and adds the label $L$:

\[
\{L, \{\alpha, \beta\}\}, \text{ where } \alpha \text{ is the label (Chomsky 1995)}
\]

On the assumption that the label is either $\alpha$ or $\beta$, this can be reduced to $\{\alpha, \{\alpha, \beta\}\}$ or $\{\beta, \{\alpha, \beta\}\}$, which in turn is equivalent to $<\alpha, \beta>$ and $<\beta, \alpha>$, respectively. (More precisely, $\{\alpha, \{\alpha, \beta\}\}$ is equivalent to $\{\{\alpha\}, \{\alpha, \beta\}\}$, which is the set-theoretic, so-called Wiener-Kuratowski notation for the ordered pair $<\alpha, \beta>$.) Suppose that by assumption, the first member of the pair provides the label. Then, Merge can be defined as in (10):

\[
\text{Merge } (\alpha, \beta) := \{\alpha, \{\alpha, \beta\}\} = <\alpha, \beta> \quad \text{(where } \alpha \text{ is the label of } \{\alpha, \beta\})
\]

Note that even though $<\alpha, \beta>$ is an ordered pair, this ordering does not encode information about the precedence relation between $\alpha$ and $\beta$. On current views, $\alpha$ and $\beta$ are ordered only in that it was assumed that the first category projects, i.e. provides the label of the resulting syntactic object. For instance, in (11), the verb like projects. Crucially, this does not mean that the verb precedes the object. The result would be exactly the same for OV languages such as German (modulo lexicon).

\[
\text{Merge (like, Mary) = } <\text{like, Mary}>, \text{ with like providing the label of } <\text{like, Mary}>
\]

Consequences of BPS: The BPS model has at least four advantages over X'-Theory (the discussion partially follows Hornstein, Nunes & Grohmann 2005: 196ff):

I. Diacritics that denote bar levels (° in X°, ‘’’ in X’, ‘P’ for XP) are not part of the lexicon. Only a system which has eliminated diacritics is therefore consistent with Inclusiveness.

(12) Inclusiveness
The computation does not introduce new items apart from lexical items and their features.

In BPS, the levels are identified by functional determination, instead. Just like the notion ‘object’ and ‘subject’ are relational notions, which are specified by the syntactic context (SpecTP vs. sister to a head), the phrase structural status of categories (maximal, minimal, intermediate) is determined by their respective position w.r.t. other nodes. (This idea originated with Muysken 1982).

(13) a. $\alpha$ is a maximal projection iff $\alpha$ does not project.
    b. $\alpha$ is a minimal projection iff $\alpha$ is selected from the numeration.
    c. $\alpha$ is an intermediate projection iff $\alpha$ is neither a maximal nor a minimal projection.

Question: What about segments (XP adjunct to XP)? If the lower segment projects, it does not count as maximal, if it does not project, the higher segment does not.

Solution: Only full categories are visible to syntax. Adjunction structures are complex categories made up of two segments (see also Kolb 1997).
II. BPS derives Endocentricity, which needs to be stipulated in X’-theory by stating that X° projects to X’, and not Y’ (where X ≠ Y). Endocentricity simply falls out from the format of Merge in (10).

III. BPS eliminates spurious, superfluous projections by getting rid of branching nodes that only dominate a single contentful node. Instead of being parsed as in (14), a name such as Mary is now reduced to (15)a (to be further simplified), while a VP is matched on the tree (15)b:

(14)    DP
        /   \ D’
       /     \\
D°   NP
     /   /\\
N°  N’
    / \\
    Mary

(15) a. N
     /  \\
   Mary

   V
     /  \\
   see D
     /  \\
   N
    /  \\
book

IV. Finally, BPS eliminates syntactic categories, thereby reducing a redundancy between the lexicon and the phrase structure component. More specifically, the fact that Mary is a noun is already stored in the lexicon. Thus, introducing a label N in the derivation does not add new information. Thus, (15)a is further simplified to (16)a, while (15)b is now transposed into (16)b.

(16) a. Mary a’. Mary
     /  \\
   see 
     /  \\
the book

As an additional bonus, rewriting (15)a as (16)a eliminates the distinction between lexical items (N in (15)a) and terminal nodes - they can now no longer be distinguished.

Question: Is (16)a’ also a well-formed object of BPS?

Answer: No, (16)a’ is not defined in BPS. In order to project, a category needs to Merge with another category. But there is no term that Mary can combine with in a meaningful way in (16)a’.

1. Mary cannot be merged with the empty set. This would yield <Mary, {}>, or (in set notation) \{Mary, \{\}\}. But combining Mary with {} is formally an object quite different from (16)a’.

2. Mary could merge with itself, resulting in <Mary, Mary>, or \{\{\}\}. This conception requires that there are two distinct occurrences of Mary (just as in The book that Mary’s brother bought did not appeal to Mary). But two names cannot be combined by any standardly sanctioned semantic composition principle. As a result, <Mary, Mary> is undefined.
Thus, BPS cannot even produce (16)a’. No such problems arise with (16)a, which contains the same syntactic information, but represents a tree theoretic object of its own (‘syntactic objects’).

NB: Trees are pairs <A, D>, where A is a set of nodes and D a set of pairs capturing the dominance relation. Trees can be trivial in that D is empty: <A, {}>. Thus, Mary in (16)a’ is actually a tree.

3. **The Linear Correspondence Axiom**

3.1. **Objectives**

**Goal A:** Derivation of basic X’-theoretic properties, among them the generalizations in (17) (Note that Kayne 1994 was written before the advent of Bare Phrase Structure, so order was still an issue.)

(17)  a. Specifiers universally precede heads  
     b. Heads do not take other heads as complements  
     c. Maximal projections do not take other maximal projections as complements

**Goal B:** Finding an algorithm for mapping structure (i.e. dominance relations) to order. The fact that the terminal α precedes the terminal β in a tree like (18) does not entail that α and β are actually pronounced in that order. If in the chemical representation C₆H₅OH for Ethanol, the symbol for carbon precedes the symbol for hydrogen, this does not mean that in the real world object - the alcohol molecules - we find the same order. Precedence expresses merely an accidental property of the medium used to write down the formula. (See also mobile metaphor in section 1.)

(18) \[
\begin{array}{c}
\text{U} \\
\text{V} \\
\alpha \quad \beta
\end{array}
\]

3.2. **The idea (informal)**

C-command as a partial order: In handout #1 it was seen that c-command does not induce an order on the terminals. Since trees are nothing else than the graph representation of partially ordered sets (together with some empirically motivated axioms), c-command can therefore not be used to generate syntactic trees on its own.

Suppose, however, that it were possible to restrict c-command in such a way that it would yield a partial order. Then, this modified version of c-command could be used as a primitive for defining trees (s.a. Frank & Vijay-Shanker 2001). This is the leading idea of Kayne (1994). More specifically, Kayne proposes that the ordering of the terminals can be read off the asymmetric c-command relations that hold between (a particular subset of) the non-terminals. This can be made explicit by demanding that trees satisfy the two conditions in (19) below. If a structure meets (19), it satisfies what Kayne calls the **Linear Correspondence Axiom** (LCA). (19) captures the essence of Kayne’s proposal, but substantially diverges from the original version; for details of the latter see 3.3.; assume that a container of a terminal x is a node non-reflexively dominating x.)
(19)  
   a. **ORDER BY ASYMMETRIC C-COMMAND**: Every terminal must have a container that is in an asymmetric c-command relation with the container of another terminal node. The terminal nodes that satisfy this relation are collected in ordered pairs of terminals.
   b. **A(NTI)SYMMETRY**: If (19)a includes a pair of terminals \(<x, y>\), then the tree must not contain a mirror image pair \(<y, x>\) such that this pair would also be generated by (19)a.

**Example**:

- Condition (19)a is fulfilled by tree (20) because each terminal has a container which either asymmetrically c-commands another container, or is asymmetrically c-commanded by another container:

\[
\text{(20)} \quad \begin{array}{c}
T \\
U \\
| \\
\alpha \\
| \\
W \\
| \\
\beta \\
| \\
Y \\
\end{array} \\
\text{T, V, X: non-terminal nodes, non-heads} \\
\text{U, W, Y: non-terminal nodes, heads} \\
\text{\alpha, \beta, \gamma: terminal nodes (= lexical items)}
\]

(21)  
   a. \(\alpha\) is dominated by \(U\), hence \(U\) is a container of \(\alpha\), and
   b. \(\beta\) is dominated by \(W\), hence \(W\) is a container of \(\beta\), and
   c. \(U\) asymmetrically c-commands \(W\)

\[
\Rightarrow \alpha \text{ has a container that asymmetrically c-commands another container}
\]

(22)  
   a. \(\alpha\) is dominated by \(U\), hence \(U\) is a container of \(\alpha\), and
   b. \(\gamma\) is dominated by \(X\), hence \(W\) is a container of \(\gamma\), and
   c. \(U\) asymmetrically c-commands \(X\)

\[
\Rightarrow \beta \text{ has a container that asymmetrically c-commands another container and} \\
\gamma \text{ has a container that is asymmetrically c-commanded by another container}
\]

\[
\Rightarrow \text{All terminals of (20) have a container, in satisfaction of (19)a.}
\]

(23)  
   (19)a generates three pairs of terminals: \(<\alpha, \beta>\), \(<\beta, \gamma>\) and \(<\alpha, \gamma>\).

- Second, (20) also observes the **antisymmetry clause** (19)b. (19)a generated \(<\alpha, \beta>\), \(<\beta, \gamma>\) and \(<\alpha, \gamma>\). In addition, there is no mirror image of any of these pairs in the tree which would also be generated by (19)a. (That is, the tree neither contains \(<\beta, \alpha>\), nor \(<\gamma, \beta>\), nor \(<\gamma, \alpha>\).)

**Antisymmetry violations**: To see the effects of the antisymmetry clause consider the tree below:

(24)  
   a. \(\alpha\) is dominated by \(W\) and
   b. \(\beta\) is dominated by \(Z\) and
   c. \(W\) asymmetrically c-commands \(Z\)

\[
\Rightarrow \text{A container of } \alpha \text{ asymmetrically c-commands a container of } \beta
\]

(25)  
   a. \(\alpha\) is dominated by \(Y\) and
   b. \(\beta\) is dominated by \(X\) and
   c. \(X\) asymmetrically c-commands \(Z\)

\[
\Rightarrow \text{A container of } \beta \text{ asymmetrically c-commands a container of } \alpha
\]

\[
\Rightarrow \text{\textbf{X}Violation of antisymmetry clause (19)b}
\]
Asymmetric c-command violations: Trees that violate the asymmetric c-command condition (19)a are too small to contain enough mother nodes for the terminals to establish the required asymmetric c-command relations:

(26) \[ \begin{array}{c} \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ Y \\
\alpha & \beta \end{array} \]

(27) a. \( \alpha \) is dominated by \( Y \) and
b. \( \beta \) is dominated by \( Y \) but
c. \( Y \) does not asymmetrically c-commands \( Y \)

\[ \rightarrow \ \times \text{Violation of asymmetric c-command clause (19)a} \]

Note that even though a node may c-command itself if a reflexive version of c-command is adopted, no node can asymmetrically c-commands itself. This follows from the fact that reflexive relations are always symmetric.

- Similarly for (28) and (29). Again, in both structures there are no asymmetric c-command relations among the mothers:

(28) \[ \begin{array}{c} \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ X \\
Y & Z \end{array} \]

(29) a. \[ \begin{array}{c} \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ X \\
\alpha & Z \end{array} \]

b. \[ \begin{array}{c} \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ X \\
Z & \alpha \end{array} \]

Those are the core cases that the LCA is meant to exclude from the set of possible trees. For the moment, they look rather abstract, but once the variables are instantiated by actual labels, the consequences can be easily assessed. Intuitively, all structures have one of two properties: either they are ‘too small’, i.e. the terminals are not separated from the nodes at which they join the tree by enough structure. This situation leads to violations of the Totality clause. Alternatively, the trees can be too ‘symmetric’ (in a geometrical sense), triggering Antisymmetry violations.

Next:
- Formal definition of LCA (in two versions)
- Consequences for X*-theory
- Consequences of integrating LCA in Bare Phrase Structure

3.3. The LCA - Formal Definitions

The formal definition of the LCA has two parts to it (the initial part of the exposition below by and large follows Stabler 1997, and not Kayne’s original version, which is somewhat less intuitive. Note also that the definitions do not directly formalize the informal rendering of the proposal in (19).).
A. First, the procedure determining the well-formedness of syntactic trees collects all pairs of terminals which are dominated by non-terminals in an asymmetric c-command relation. This is done by the relation T in (31). T defines a set of ordered pairs. The member of T have the property that the first member of the pair is dominated by a node that asymmetrically c-commands a node which dominates the second member of the pair:

(30) a. Domains:
\[ N_T = \text{set of terminal nodes} \]
\[ N_{NT} = \text{set of non-terminal nodes} \]

b. Relations:
\[ \rightarrow = \text{reflexive dominance relation} \]

(31) Non-terminal-to-terminal relation T (adapted from Stabler 1997):

For any \( a, b \in N_T \), \( aTb = \text{Def} (\exists x \in N_{NT})(x \rightarrow a \land y \rightarrow b \land x \text{ asymmetrically c-commands } y) \)

(T collects all pairs of terminals which are dominated by non-terminals in asymmetric c-command relation)

B. The LCA states that the set defined by T has to satisfy the requirements of a total (weak) order, aka linear order\(^1\).

(32) Linear Correspondence Axiom (LCA)

\[ T \] is a linear ordering of \( N_T \), i.e. for any \( x, y, z \in N_T \) the following holds:

(i) \( x \# y \rightarrow xTy \lor yTx \) (Totality/Linearity)
(ii) \( xTy \land yTz \rightarrow xTz \) (Transitivity)
(iii) \( xTy \land yTx \rightarrow x=y \) (Antisymmetry)

Kayne’s original version looks as follows (adapted from Kayne 1994: 5f):

(33) Linear Correspondence Axiom (LCA)

\[ d(A) \] is a linear ordering of the terminals of the tree

(34) a. \( A = \{<X, Y> \in N_{NT} \times N_{NT} | X \text{ asymmetrically c-commands } Y \} \)

(A collects all pairs of non-terminals in asymmetric c-command relation)

b. The image \( d \) of a non-terminal node \( x \) is the set of terminals that \( x \) dominates

For any \( X \in N_{NT} \), \( d(X) = \text{Def} \{a \in N_T | X \rightarrow a \} \)

c. The image \( d \) of an ordered pair of nodes:

For any \( X, Y \in N_{NT} \), \( d(<X, Y>) = \text{Def} \{<a, b> \in N_T \times N_T | X \rightarrow a \land Y \rightarrow b \} \)

d. The image \( d \) of a set of ordered pairs of nodes:

For any \( S \in N_{NT} \times N_{NT} \), \( d(S) = \text{Def} \{ \cup(p) | p \in S \} \)

(35) Generalized union of \( M \) (Notation: ‘\( \cup M \)’)

a. \( \cup M = \text{Def} m_1 \cup m_2 \cup ... \cup m_n \) (for \( 1 \leq i \leq n \), and \( m_i \in M \))

b. \( \cup M = \text{Def} \{x | \exists i[x \in m_i]\} \) (for \( 1 \leq i \leq n \), and \( m_i \in M \))

\(^1\)Kayne uses the term ‘ordering’, not ‘order’. If precedence is taken to be irreflexive, the relation should also be required to be a strong order (irreflexive and asymmetric (see Handout #1).
Example:

(36)

C
  |    
  d_1  d_2  e_1  e_2
  c_1

(37)  

a.  \( A_{(36)} = \{<C, D>, <C, E>\} \)

b.  \( d(<C, D>) = \{<c_1, d_1>, <c_1, d_2>\} \)

c.  \( d(<C, E>) = \{<c_1, e_1>, <c_1, e_2>\} \)

d.  \( d(A_{(36)}) = d(\{<C, D>, <C, E>\}) = \{\cup d(p) | p \in \{<C, D>, <C, E>\}\} = \)

= \( d(<C, D>) \cup d(<C, E>) = \{<c_1, d_1>, <c_1, d_2>\} \cup \{<c_1, e_1>, <c_1, e_2>\} = \)

= \{<c_1, d_1>, <c_1, d_2>, <c_1, e_1>, <c_1, e_2>\} = \)

= \{<c_1, d_1>, <c_1, d_2>, <c_1, e_1>, <c_1, e_2>\} = \)

3.4. SAMPLE APPLICATION OF LCA

Consider how the LCA derives the well-formedness of the tree in (38):

(38)  

TP

  DP_1  VP

      John  V°  DP_2

          saw  N°

          Mary

First, a set is assembled - the T-set - which collects pairs of terminals. In particular, these terminals need to fulfill the condition that they are dominated by non-terminals in an asymmetric c-command relation.

(39)  

<John, saw> \in T_{TP} because

a.  DP_1 dominates John

b.  V° dominates saw and

c.  DP_1 asymmetrically c-commands V°

(40)  

<John, Mary> \in T_{TP} because

a.  DP_1 dominates John

b.  DP_2 dominates Mary and

c.  DP_1 asymmetrically c-commands DP_2

(41)  

<saw, Mary> \in T_{TP} because

a.  V° dominates saw

b.  N° dominates Mary and

c.  V° asymmetrically c-commands N°

Nothing else is in the T-set for TP. Thus, the extension of T_{TP} is as in (42).

(42)  

T_{TP} = \{<John, saw>, <John, Mary>, <saw, Mary>\}
Second, it is verified that the T-set observes the properties of a linear ordering. This is the case: the set in (42) is total because all terminals are members of at least one set. It is transitive, as can be easily verified, and it is antisymmetric. The latter holds because there are no conflicting ordering statements of the form \(<\alpha, \beta>\) and \(<\beta, \alpha>\) in \(T_{TP}\).

Thus, the tree in (38) is sanctioned by the LCA.

Next, consider some of the inadmissible trees that are weeded out by the LCA. These consequences of the LCA translate into generalizations about possible X'-structures.

3.5. CONSEQUENCES FOR X'-THEORY

Complements of heads: Complement of a head cannot be a head. More generally, two heads can never be sisters.

(43) a. \(*V'\) b. \(\checkmark V'\)

\[
\begin{array}{c}
V^o & N^o \\
| & | \\
saw & Mary \\
\end{array}
\quad \begin{array}{c}
V^o \\
| \\
saw \\
\end{array}
\quad \begin{array}{c}
DP \\
\end{array}
\quad \begin{array}{c}
N^o \\
| \\
\end{array}
\quad \begin{array}{c}
boring \\
\end{array}
\]

(44) \(T = \{\}\) \(\Rightarrow X\text{LCA} \quad \text{(totality violation)}\)

Adjunction: At first sight, it seems as if adjunction is inadmissible, because AP asymmetrically c-commands N°, and NP₂ asymmetrically c-commands A°:

(45) \(\begin{array}{c}
V^o \\
| \\
AP \\
| \\
A^o \\
| \\
boring \\
\end{array}
\quad \begin{array}{c}
NP_1 \\
| \\
NP_2 \\
| \\
N^o \\
| \\
book \\
\end{array}
\)

(46) \(<\text{boring, book}> \in T_{TP}\) because
a. AP dominates boring
b. N° dominates book and
c. AP asymmetrically c-commands N°

(47) \(<\text{book, boring}> \in T_{TP}\) because
a. NP₂ dominates book and
b. A° dominates boring and
c. NP₂ asymmetrically c-commands A°

(48) \(T_{NP} = \{<\text{boring, book}>, <\text{book, boring}>\} \Rightarrow X\text{LCA} \quad \text{(antisymmetry violation)}\)
Kayne avoids this undesirable result by altering the definition of c-command. The new definition explicitly exempts segments from nodes that can enter into a c-command relation:

**Assumption:** Segments do not count for the computation of c-command.

\[
\alpha\ c\text{-commands}\ \beta \iff \begin{array}{ll}
(i) & \alpha\ and\ \beta\ are\ categories\ (not\ just\ segments\ thereof) \\
(ii) & \text{no segment of } \alpha\ \text{contains } \beta\ (= \alpha\ excludes \beta) \\
(iii) & \text{every category dominating } \alpha\ \text{also dominates } \beta. \\
\end{array}
\]

As a consequence, the tree in (45) satisfies the LCA. Crucially, NP₂ no longer asymmetrically c-commands A°:

\[
\text{<book, boring> } \notin T_{TP} \because \begin{array}{ll}
a. & \text{NP₂ dominates } book \text{ and} \\
b. & \text{A° dominates } boring \text{ but} \\
c. & \text{NP₂ is a segment, and therefore does } not \text{ enter c-command relations} \\
\end{array}
\]

\[
T_{NP} = \{<\text{boring, book}>\} \rightarrow \checkmark\text{LCA}
\]

The revised version of c-command also admits head adjunction. Note that V₁ does not c-command V₂ because V₁ does not exclude V₂.

\[
\text{<has, played> } \in T_X \because \begin{array}{ll}
a. & \text{V₂ dominates } has \text{ and} \\
b. & \text{V₁ dominates } played \text{ and} \\
c. & \text{V₂ asymmetrically c-commands V₁} \\
\end{array}
\]

\[
T_{V₁} = \{<\text{has, played}>\} \rightarrow \checkmark\text{LCA}
\]

**Specifiers:** The reason why the prenominal adjunction structure (45) initially failed to conform with the LCA was that two XPs (the adjoined AP and an NP segment) were construed as sisters. More generally, such configurations are always blocked by the LCA, as the LCA restricts sisterhood to cases in which one sister is a head, and the other a non-head. This property of the LCA entails interesting consequences for the way in which specifiers are parsed.

To begin with, consider the tree in (56), which contains an intermediate projection T’. Such structures violate the LCA, because the specifier and the intermediate node are (i) in a symmetric c-command relation and (ii) contain at least one non-terminal node each:
(56) \[ \begin{array}{c}
TP \\
\downarrow \\
DP \\
\downarrow \\
N^\circ \\
\downarrow \\
John \\
\downarrow \\
-s \\
\uparrow \\
T' \\
\downarrow \\
T^\circ \\
\downarrow \\
VP \\
\downarrow \\
\text{smoke}
\end{array} \]

(57) \( <\text{John}, -s> \in T_{TP} \) because  
   a. DP dominates John 
   b. \( T^\circ \) dominates -s 
   c. DP asymmetrically c-commands \( T^\circ \)  

(58) \( <-s, \text{John}> \in T_{TP} \) because  
   a. \( N^\circ \) dominates John 
   b. \( T' \) dominates -s 
   c. \( T' \) asymmetrically c-commands \( N^\circ \)  

(59) \( T_{NP} = \{<\text{John}, -s>, <-s, \text{John}>, \ldots\} \rightarrow \text{XLCA} \) (antisymmetry violation) 

**COROLLARIES:**  
   o There are no intermediate \( X^\prime \)-levels. 
   o All specifiers are adjuncts. 

An alternative (Chomsky 1999): LCA only sees \( X^\circ \) and XP, but not \( X' \).  

**Number of adjuncts:** The LCA has a final important consequences: it prohibits multiple adjunction of XPs as well as of heads. Interestingly, the reasons why these two types of trees are blocked are not the same.  
   o *Multiple XP-adjunction* is barred due to the fact that segments are ignored for c-command. 

(60) \[ \begin{array}{c}
\text{DP} \\
\downarrow \\
\text{NP}_1 \\
\downarrow \\
\text{AP}_1 \\
\downarrow \\
A_1^\circ \\
\downarrow \\
\text{long} \\
\downarrow \\
\text{NP}_2 \\
\downarrow \\
\text{AP}_2 \\
\downarrow \\
A_2^\circ \\
\downarrow \\
\text{boring} \\
\downarrow \\
\text{NP}_3 \\
\downarrow \\
\text{N}^\circ \\
\text{book}
\end{array} \]

(61) \( <\text{long}, \text{boring}> \in T_{TP} \) because  
   a. \( \text{AP}_1 \) dominates *long* 
   b. \( A_2^\circ \) dominates *boring* and 
   c. \( \text{AP}_1 \) asymmetrically c-commands \( A_2^\circ \) (first category to dominate \( \text{AP}_1 \) and \( \text{AP}_2 \) is DP) 

(62) \( <\text{boring}, \text{long}> \in T_{TP} \) because  
   \( \text{AP}_2 \) asymmetrically c-commands \( A_1^\circ \) (first category to dominate \( \text{AP}_2 \) and \( \text{AP}_1 \) is DP) 

(63) \( T_{NP} = \{<\text{boring}, \text{book}>, <\text{book}, \text{boring}>\} \rightarrow \text{XLCA} \) (antisymmetry violation) 

**Solution:** \( \text{AP}_1 \) adjoins to \( \text{AP}_2 \). But this makes wrong predictions for *former French president.*
Multiple X°-adjunction is finally blocked by the totality - and not by the antisymmetry - requirement:

(64)  will have seen...

\[
\begin{array}{c}
\text{VP} \\
\text{V}_1^° \\
\text{DP} \\
\text{V}_3^° \\
\text{V}_2^° \\
\text{V}_1^° \\
\text{will} \\
\text{have} \\
\text{seen}
\end{array}
\]

\(V_1^°\) does not c-command any of the other heads, because it does not exclude them. But \(V_3\) and \(V_2\) are in a symmetric c-command relation: for both, the first dominating category is VP, which in turn also dominates the other head. It follows that the two heads will and have cannot be ordered.

(65)  \(T_{V_1} = \{<\text{have}, \text{seen}>\} \rightarrow \mathbf{XLCA}\)  (totality violation - will and have are not ordered)

3.6. REDUCING DIMENSIONS (GOALB)

Recall that the definition of c-command is inherently blind to precedence. If a hypothetical T relation defines e.g. the set \{<a,b>, <b,c>, <a,c>\}, this does not say anything about whether T translates as precedence or subsequence or some other, more complex linearization relation. Imagine e.g. a backwards-talking competition among kids, where The book sucks has to be rendered as sucks book the. In such a competition, the speakers would presumably still create the strings in the normal order first, and only map them to the backwards order afterwards.

The potential point of confusion here is that linear order is a technical (mathematical) term, which describes properties of certain relations - it does not make any commitment as to temporal precedence at the PF interface. (Note again that Kayne uses ordering instead of order, but this, if anything, adds to the confusion.)

Linearization: Kayne suggests that left to right temporal sequencing is an inherent property of language (not the strongest motivation). Assuming that the mapping from the linear order established by the T-set to order in the tree is indeed defined in terms of precedence (i.e. the T-relation is be mapped onto precedence relation) the following consequences for phrase structure trees defined by the LCA ensue:

(66)  Consequences of LCA

Hierarchically higher categories precede lower ones. More specifically,

a. specifiers precede heads
b. heads precede complements
c. modifiers precede the heads of the categories they modify
3.7. **EPILOGUE: URIAGEREKA (1999)**

Uriagereka (1999) proposes an alternative way for deriving order relations in terms of *Multiple Spell Out*. Recall again the reason why X'-levels must not be part of the representations in the LCA. X' would asymmetrically c-command Y°, creating an ordering conflict:

\[
(67) \quad \text{YP} \xrightarrow{3} \text{X'} \xrightarrow{1} \text{Y°} \xrightarrow{1} \beta \\
\quad \alpha \quad T = \{<\alpha, \beta>, <\beta, \alpha>, \ldots\} \Rightarrow \text{X} \subseteq \text{LCA} \quad \text{(antisymmetry violation)}
\]

Uriagereka (1999) suggest to keep intermediate projections and offers an interesting way to derive linearization. The account relies on two ingredients:

- Specifiers are spelled out separately and are treated like terminals. (Otherwise, they could not be ordered.) This effectively collapses YP into the terminal string α.

- C-command is only defined for terminals. As a consequence, the new representation observes the LCA, and at the same time derives the desired word order α > β. (For details see also exposition Lasnik & Uriagereka 2005: 48ff)

\[
(68) \quad \text{XP} \xrightarrow{3} \alpha \xrightarrow{1} \text{X'} \xrightarrow{1} \beta
\]

4. **EXTENSIONS**

4.1. **BARE PHRASE STRUCTURE AND THE LCA**

Combining the LCA with or integrating the LCA into BSP leads to various incompatibilities. The BPS representation below violates the LCA, as the two heads cannot be ordered (totality):

\[
(69) \quad \text{X} \quad \text{(where X is the label - possibly saw)}
\]

\[
\text{saw} \quad \text{Mary}
\]

**Solution** (Chomsky 1995; Hornstein et. al 2005: 229): Assume that the LCA serves as a PF-condition which only applies to overt categories, ignoring traces and silent nodes.

**Option A:** Add silent structure inbetween verb and object.

\[
(70) \quad \text{X} \quad \text{(saw})
\]

\[
\text{DP} \quad \text{D} \quad \text{Mary}
\]
**Option B:** Chomsky (1995): Movement of one of the lexical items to a higher position, resulting in an asymmetric c-command configuration.

(71) \[ X \]

\[ \text{saw} \quad Y \]

\[ t_i \quad \text{Mary} \]

Note that the trace of the verb does not induce an LCA violation on the assumption that the LCA operates as a PF-condition which ignores traces.

**Problem I:** If the LCA only sees overt nodes, how can single terminal utterances such as *Go!* or *祂θε* (‘S/he came’) ever be linearized without inducing a violation of the totality clause?

**Problem II:** The assumption that only overt categories are visible to the LCA does not align well with the Copy Theory of movement. More precisely, at the moment at which the LCA applies, the copies/traces are not distinguished from their antecedents - they are just different occurrences of one and the same category. Ideally, the fact that all but one copy end up as silent should be derived, and not assumed as a primitive. Thus, ignoring all silent occurrences only offers a suboptimal solution to the problem noted in connection to (71).

An alternative solution presents itself, though: copies are internally complex, hence contain structure. This property (together with the remarks in 4.1.) might help to construe the relation between the head and the copy as asymmetric.

### 4.2. COPIES AND THE LCA

Assume, alternatively, that copies are visible to the LCA. Then, another problem arises: the relation among the copies triggers LCA violations, this time in offense of the symmetry clause.

Suppose that, as shown in (72), \( \beta \) moves across an attracting head \( \alpha \). Prior to movement, \( \alpha \) asymmetrically c-commands the lower copy of \( \beta \), while subsequent to raising of \( \beta \) to the left of \( \alpha \), the higher occurrence of \( \beta \) asymmetrically c-commands \( \alpha \). Thus, the T-set contains the symmetric pairs \( <\beta, \alpha> \) and \( <\alpha, \beta> \), and the terminals embedded inside the movement copies cannot be ordered.

(72) \[ T = \{<\beta, \alpha>, <\alpha, \beta>, \ldots\} \]

\[ \times \text{LCA} \quad \text{(antisymmetry violation)} \]

\[ \beta \quad \text{higher copy} \]

\[ \alpha \quad \text{Attractor} \]

\[ \beta \quad \text{lower copy} \]

NB: If the order relation is assumed to be a weak order (transitive, total, reflexive), *string vacuous* movement is allowed by the LCA to leave copies.

Analyses that address this problem lead to interesting new insights into how the LCA can be employed in the derivation of properties of movement.
I. Determining which copy is pronounced: Nunes (1996, 1999) uses the above complication for the LCA as a starting point for deriving the fact that only one copy is pronounced in movement chains. Concretely, he assumes that the LCA is a condition on linearization, which only sees overt categories. Thus, foregoing Spell-Out (or eliding) a copy - \( \beta_{lower} \) in (72) - resolves the antisymmetry conflict.

The analysis is further developed into an account for why the higher, and not the lower, copy must delete. In a nutshell, features that drive movement are eliminated in higher copies. Thus, higher copies bear fewer features than lower ones. Assuming that deletion of fewer features represents a more economical option, higher copy delete. Naturally, this presupposes the existence of some additional - presumably not cost-free - mechanism that establishes a correspondence between the higher and lower copies, transferring the information that the features on the higher copy have been checked to the lower copy. This consideration, among others, led to the current conception of dislocation as remerge or internal merge (Chomsky 2004). Note incidentally that Nunes proposal is not directly compatible with the internal merge model, because on such theories, all occurrences of a category (if there are more than one in the first place - multiple dominance) are identical, thus an imbalance between number of features does not arise.

II. Locality and phases: The fact that (72) fails to observe the LCA can also be interpreted as a sign that the LCA only evaluates parts of the syntactic derivation at a time, and needs to be relativized to subparts of the derivation, e.g. phases. Such approaches can, among others, account for the locality of A-movement (Superraising) and Superiority effects (Lechner 2004; see also Richards 2001). The idea in short is as follows. Suppose that each phase has its own T-set. Then, movement may cross other LCA visible material as long as it leaves the local phase, as in (73)a. If movement needs to pass through the edge of the phase, though, as in (73)b, the initial short movement induces an LCA-violation inside the lower phase.

(73) a. Licit movement

\[
\begin{align*}
T_{lower Phase} &= \{<\beta, \gamma>\} & \checkmark \text{LCA} \\
T_{higher Phase} &= \{<\beta, \alpha>\}
\end{align*}
\]

\[
\begin{align*}
\beta & \quad \alpha_{\text{Attractor}} \\
\gamma & \quad \text{PHASE BOUNDARY}
\end{align*}
\]

b. Illegitimate Violation

\[
\begin{align*}
T_{lower Phase} &= \{<\beta, \gamma>, <\gamma, \beta>\} & \times \text{LCA} \\
T_{higher Phase} &= \{<\gamma, \alpha>\}
\end{align*}
\]

\[
\begin{align*}
\beta & \quad \gamma & \quad \alpha_{\text{Attractor}} \\
\gamma & \quad \text{PHASE BOUNDARY}
\end{align*}
\]

First, this setup ensures that a (phase-external) head can only attract the highest suitable category in the subordinate phase (\( \beta \) in (73)). This prohibition constitutes the core of the Minimal Link
Condition, which demands attraction of the closest feature-compatible candidate. Second, in such a system, all violations are triggered locally, inside a phase. As a result, the principles evaluating syntactic locality no longer have to have access to the internal make up of higher/lower phases, as e.g. in Chomsky (2001: 14) version of the Phase Impenetrability Condition, where phases are evaluated once the derivation reaches the next higher phase. (For further details see Lechner 2004).

4.4. REPAIR STRATEGIES FOR LCA

In sum, it was seen that there are various ways to suspend LCA violations. These strategies include:

- Addition of (functional, empty) structure
- Movement of a category out of a symmetric c-command relation
- PF-deletion of all but one copy (Nunes 1999)
- Multiple Spell-Out (Uriagereka 1999)
- Relativization of the LCA to subdomains (Richards 2001; Lecher 2004)

REFERENCES


Richards, Norvin. 2001. A Distinctness Condition on Linearization. Ms., MIT.
