1. Main idea

linear order reflects the order in which elements are merged

(1) equivalence

\[ \langle x, \langle y, z \rangle \rangle = / x y z / \]

replaces

(2) Linear Correspondence Axiom (LCA)

Linear order is a function of asymmetric c-command relations (Kayne 1994)

2. The LCA

(3) More exactly:

Given a set \( T \) of terminals of a phrase marker \( P \) and an asymmetric c-command relation among the non-terminals of \( P \), the dominance relation from non-terminals to terminals \( d(A) \) yields a linear ordering of \( T \)

(4) General correctness of the structure-order correspondence
a. subject-predicate order
b. extracted element precedes its remnant/trace

(5) Problems of the LCA (as stated)

a. global (representational) rather than local (derivational)
b. violates bare phrase structure requirement (Chomsky 1995, section 4.8)

Ad (5a)

(6) specifiers: no linear order of \( x \) and \( y \)

solution

YP and YP are ‘segments’, lower segment does not c-command
Ad (5b)

(7) bare complements: no linear order of $y$ and $z$

solution
nonbranching
structure:

(8) Bare Phrase Structure (Chomsky 1995): structure is a function of merge alone
- no segments
- no nonbranching structures
- no restrictions on number of adjunctions

Both problems are solved if linear order can be made a function of merge

3. Merge

(9) Merge
1. select 2 elements $x, y$ from a numeration (N)
2. combine $x$ and $y$ yielding $P$

(10) Problems
a. why 2?
b. no designated output (multiple tree creation, interarboreal operations)
c. recursion: select targets $P$ in all instances of merge except the first step
d. move: select may target a term of $P$, but only for one of the two elements to be merged (= extension condition)

(11) Simplification
Merge assigns an element from the Numeration to the Derivation

(12) Addressing the problems
a. only 1
b. Derivation = designated output (no multiple tree creation, no interarboreal operations)
c. Merge = iterative (recursion = output of a derivation may appear in the next numeration)
d. No move (bottom up derivation: remerge from Numeration + ‘copy’ deletion; top down derivation: merge only once, leaving a gap)

(13) Asymmetry
Temporal asymmetry between a newly merged element and already existing structure (Jaspers 1998)
4. Top-down derivation (split-merge)

(14) **Derivation** (**D**)  
1. \( N = \{ a, b, c, d, e \} \) and \( D = \emptyset \)  
2. Select \( a \), yielding \( N = \{ b, c, d, e \} \) and \( D = \langle a, N \rangle \)  
3. Select \( b \), yielding \( N = \{ c, d, e \} \) and \( D = \langle a, \langle b, N \rangle \rangle \)  
4. Select \( c \), yielding \( N = \{ d, e \} \) and \( D = \langle a, \langle b, \langle c, N \rangle \rangle \rangle \)  
5. Select \( d \), yielding \( N = \{ e \} \) and \( D = \langle a, \langle b, \langle c, \langle d, N \rangle \rangle \rangle \rangle \)  
6. Select \( e \), yielding \( N = \emptyset \) and \( D = \langle a, \langle b, \langle c, \langle d, \langle e, N \rangle \rangle \rangle \rangle \rangle \rangle \)

(15) **Merge**  
Select \( x \in N \), yielding \( N = N - x \) and \( D = \langle x, N \rangle \)

(16) **Constituent**  
P is a constituent if it is the output of Merge (i.e. \( N \) or \( D \))

(17) **Syntactic position**  
The pair \( \langle x, N \rangle \) defines the syntactic position of \( x \)

(18) **Grammatical relation**  
A grammatical relation between \( x \) and \( y \) exists iff \( D = \langle x, y \rangle \)

(19) **Linear order**  
\( \langle a, \langle b, c \rangle \rangle = \langle a, b, c \rangle = / a b c / \)

(20) **Linear Correspondence Axiom**  
\( \langle x, y \rangle = / x y / \)

5. Back to the LCA problems

(21) **Problems of the (old) LCA**  
a. global (representational) rather than local (derivative)  
b. violates bare phrase structure requirement

Ad (21a)  
(22) Order is a function of merge, i.e. established at each step of the derivation

(23) specifier-head ordering:  
\( N = \{ x, y, z \} \quad D = \emptyset \)  
\[ x \quad \{ y, z \} \]
\[ \text{Merge } x \text{ yielding } N = \{ y, z \} \text{ and } D = \langle x, N \rangle \]
\[ x \]
\[ y \quad \{ z \} \]
\[ \text{Merge } y \text{ yielding } N = \{ z \} \text{ and } D = \langle x, \langle y, N \rangle \rangle \]
bare phrase structure requirements are met (no vacuous structure)

head-complement ordering:

Merge $z$ yielding $N = \emptyset$ and $D = \langle x, \langle y, \langle z, N \rangle \rangle \rangle$

6. Proof

Split-merge yields a derivational record $K$, which may be expressed as a set of sets of elements in syntactic positions (cf. (17)) at each step of the derivation

initial situation: $\{ a, b, c, d, e \}$
first merge: $\{ b, c, d, e \}$
next merge: $\{ c, d, e \}$
next merge: $\{ d, e \}$
next merge: $\{ e \}$

$K = \{ \{ a, b, c, d, e \}, \{ b, c, d, e \}, \{ c, d, e \}, \{ d, e \}, \{ e \} \}$

Kuratowski’s Definition (Kuratowski 1921): $\{ \{ a \}, \{ a, b \} \} = \langle a, b \rangle$

$K = \langle e, d, c, b, a \rangle$ (succession relation, interpretable at Spell-Out)

Linear order is a function of the order in which elements are merged (split off from $N$)

7. Outlook

What determines the order in which elements are merged?

Ideally: order is free, but interpretation is not.

Dependency: since each step yields $\langle x, N \rangle$, where $N$ is an unordered set, only $x$ is syntactically active, and nothing in $N$ can turn $x$ in a syntactic dependent.


References


Faculty of Arts, PO Box 716, NL-9700 AS, Groningen, The Netherlands
j.fortuny.andreu@rug.nl • c.j.w.zwart@rug.nl • www.let.rug.nl/zwart/diug/