Some definitions

Jason Merchant, University of Chicago, 2014

1. A grammar G consists of a pair of a set of lexical elements L and a set of operations O:
   $G = < L, O >$

2. A derivation on a numeration D_N is a pair:
   $D_N = < N, < PM_1, ..., PM_n >>$, where

   1. N, called the Numeration, is a nonempty set of lexical elements drawn from L and a possibly empty set S of phrase markers PM (each of which is itself the result of a separate convergent or semi-convergent derivation), and
   2. $< PM_1, ..., PM_n >$ is an ordered n-tuple of phrase markers PM.

3. A derivation D_N is said to be *convergent* (or to *converge*)[1] iff

   1. PM_n contains no unvalued (:__) features
   2. PM_n contains no unchecked strong (*) features
   3. PM_n contains no unchecked selectional features
   4. All elements in the Numeration have been Merged
   5. For each adjacent pair of phrase markers $< PM_k, PM_{k+1} >$ in D_N, there is an operation $\omega \in O$ such that ω applied to PM_k yields PM_{k+1}.

4. A phrase P (including a sentence) is *well-formed* iff there is at least one convergent derivation for P.

5. The Minimalist Program, in essence = $min|O|$ (Minimize the number of operations in O).

[1] A derivation D_N is *semi-convergent* iff it satisfies conditions 2-5 of this definition.

1 Feature Structures

A lexical item LI has the following feature structure, with categorial, inflectional (or morphological), and selectional feature arrays:

$$LI \begin{bmatrix} \textsc{cat}[...] \\ \textsc{infl}[...] \\ \textsc{sel}[...] \end{bmatrix}$$

Some examples:

(1) $$dog \begin{bmatrix} \textsc{cat} & [N, \phi : 3sm] \\ \textsc{infl} & [\text{Case:\_\_}] \\ \textsc{sel} & [\ ] \end{bmatrix}$$

(2) $$see \begin{bmatrix} \textsc{cat} & [V, -aux] \\ \textsc{infl} & [\ ] \\ \textsc{sel} & [D] \end{bmatrix}$$

(3) $$v_{trans} \begin{bmatrix} \textsc{cat} & [v, -aux] \\ \textsc{infl} & \begin{bmatrix} \phi : \_\_ \\ V^* \\ \text{Infl:\_\_} \\ \text{Case:}\textsc{acc} \end{bmatrix} \\ \textsc{sel} & \begin{bmatrix} < D, V > \end{bmatrix} \end{bmatrix}$$

(4) $$T_{pres} \begin{bmatrix} \textsc{cat} & [T, +\textit{fin, Tns:pres}] \\ \textsc{infl} & \begin{bmatrix} \phi : \_\_ \\ \text{Case:}\textsc{nom} \end{bmatrix} \\ \textsc{sel} & \begin{bmatrix} v \end{bmatrix} \end{bmatrix}$$

2 Operations

(5) Merge(α, β)
For any syntactic objects α, β,[2] where α bears an unchecked selectional feature F, and β bears a matching categorial feature F′,
call α the head and

a. let α = { γ, { α, β}}
call γ the label (or projection) and

b. let F be checked (written <F>), and

c. let $\gamma = \alpha \cap \overline{\Im}$, where $\overline{\Im}$ is the set of all unchecked non-inflectional features[3]

(6) Adjoin(α, β)
For any syntactic objects α, β, where neither α nor β has any unchecked selectional feature,
call α the host, and

a. let α = { γ,{ α, β}}
call γ the label (or projection) and

b. let $\gamma = \alpha$

(7) Agree(X,Y; F) (read: 'X triggers agreement on Y with respect to F' or 'Y agrees with X in F' or 'X controls agreement on target Y for F')
For any syntactic objects X and Y in a phrase marker, where X bears a feature F with value Val(F) and Y bears a matching[4] unvalued[5] inflectional feature F′:__, and either X c-commands Y or Y c-commands X,

a. let Val(F′) = Val(F)

(8) Move$_{head}$(X, Y) (read: 'Y moves to X')
If Y is a head with feature F, X a head with a matching feature F, and X c-commands Y, and F is a strong inflectional feature on either Y or X, then

a. let X = {X, {Y, X}} and

b. let F* = $F^{<*>}$, and

[2] α is a *syntactic object* iff α is a phrase marker which is the result of a convergent or semi-convergent derivation.

[3] In other words, all category features project, all unchecked selectional features project, and no inflectional features project. Inflectional features are therefore found only on heads, never on projections.

[4] A feature F matches a feature F′ iff F=F′.

[5] A feature F is unvalued iff Val(F)=∅.

c. let Y = <Y>

(9) Move_{phrase}(Y, X) (read: ‘Y moves to specXP’)
If X is a projection with a feature F, Y a maximal projection with a matching feature F, and X contains Y, and F is strong (marked F*) on X or Y or both, then

a. let X = {X, {Y, X}} and
b. let all occurrences of F* on {X, Y} = $F^{<*>}$, and
c. let Y = <Y>

3 Other

Three **major** syntactic phenomena have been factored out of the above definitions and must be added to the system to make it account for word order and other important syntactic facts:

(10) Linearization (an algorithm or principle to determine the linear order of any two sister nodes)[6]

(11) Locality of application (Relativized Minimality)

(12) The spellout of complex heads by the Morphology[7]

Also, Merge must be made sensitive to the order of the selectional features, if there is more than one; either we define different kinds of features, e.g., COMP and SPEC features, and declare that COMP features must be checked (trigger Merge) before SPEC features, or, equivalently, the definition of Merge must be altered to be sensitive to an ordered n-tuple (list) of selectional features.

[6]This could be done on some general basis, as Kayne 1994 does with his Linear Correspondence Axiom (LCA: x precedes y iff x c-commands y, for any two heads x and y, roughly), or on a more mundane, potentially head-by-head differing basis, by e.g. making the strong diacritic that drives movement come in two varieties: *< and *>, with *< resulting in the moved element preceding the probe, and *> following; the minimal changes to the definitions of the Move operations are left as an exercise for the reader. The same applies, *mutatis mutandis*, to Merge of complements and specifiers, and to adjoined elements.

[7]The input to the morphological component of the grammar is PM_n; the notion of generating a string can be defined on the output of the morphological component:

1. A string s is *generated* iff there is a well-formed phrase for which it holds that the concatenation of the Vocabulary Items that realize its ordered terminal nodes corresponds to s.