Some syntactic definitions
Jason Merchant, University of Chicago, 2019

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:
   $D_N = <N, <PM_1, \ldots, PM_n>>$, 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. $<PM_1, \ldots, PM_n>$ 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 phrasal movement (> or <) features
   3. $PM_n$ contains no selectional features
   4. $PM_n$ contains no head movement features (=)
   5. All elements in the Numeration have been Merged
   6. For each adjacent pair of phrase markers $<PM_k, PM_{k+1}>$ in $D_N$, there is an operation $\omega \in O$ such that $\omega$ 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 2-6 of this definition.
1 Operations

(1) Merge(α, β)
For any syntactic objects α, β, where α bears a nonempty selectional list \( \ell = \langle \bullet F_1, \ldots, \bullet F_n \rangle \) of selectional features, and β bears a categorial feature \( F' \) that matches \( \bullet F_1 \), call α the head and
a. let \( \alpha = \{ \gamma, \{ \alpha - \ell, \beta \} \} \) call γ the projection of α, and
b. if \( n > 1 \), let \( \ell = \langle \bullet F_2, \ldots, \bullet F_n \rangle \), else let \( \ell = \emptyset \), and
c. let \( \gamma = \begin{bmatrix} \text{CAT} & [\text{cat}(\alpha)] \\ \text{SEL} & [\ell] \end{bmatrix} \)

(2) 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 \( \alpha = \{ \gamma, \{ \alpha, \beta \} \} \) call γ the label (or projection) and
b. let \( \gamma = \alpha \)

(3) Move\textsubscript{head}(X,Y) (read: ‘Y moves to X’ or ‘X attracts Y’)
For any syntactic heads X, Y, where X has feature F= (‘suffixing on F’) or =F (‘prefixing on F’), Y has a matching feature F, and X c-commands Y, and there is no head Z that intervenes between X and Y, then
a. if X has F=, let X = \([\text{cat}(X) Y X] \), otherwise let X = \([\text{cat}(X) X Y] \), and
b. let Y = <Y>

(4) Move\textsubscript{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>

\[ In other words, all category features project, all unused selectional features project, and no inflectional features project. Inflectional features are therefore found only on heads, never on projections. \]
(5) 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\(^3\) unvalued\(^4\) inflectional feature F\(^′\):__, and X is accessible to Y,
a. let Val(F\(^′\) ) = Val(F)

2 Feature Structures

A lexical item LI has the following feature structure, with categorial, inflectional (or morphological), and selectional feature arrays:\(^5\)

\[
LI = \begin{bmatrix}
\text{CAT} & \ldots \\
\text{INFL} & \ldots \\
\text{SEL} & \ldots \\
\end{bmatrix}
\]

Some examples:

(6) √libro
\[
\begin{bmatrix}
\text{CAT} & [N, \text{gender:masc}, \text{number:sg}] \\
\text{INFL} & \text{Case:__} \\
\text{SEL} & []
\end{bmatrix}
\]

(7) √eat
\[
\begin{bmatrix}
\text{CAT} & [V] \\
\text{INFL} & \text{person:__, number:__ ]} \\
\text{SEL} & [<(D)>]
\end{bmatrix}
\]

(8) √dog
\[
\begin{bmatrix}
\text{CAT} & [N, \phi : 3sm] \\
\text{INFL} & \text{Case:__} \\
\text{SEL} & []
\end{bmatrix}
\]

(9) √see
\[
\begin{bmatrix}
\text{CAT} & [V] \\
\text{INFL} & [] \\
\text{SEL} & [D]
\end{bmatrix}
\]

\(^3\)A feature F matches a feature F\(^′\) iff F=F\(^′\).
\(^4\)A feature F is unvalued iff Val(F)=\emptyset.
\(^5\)If Georgi 2014 is right, then we don’t need to structure the ‘inflectional’ (including Agree and movement-triggering) and selectional features this way; we merely need to order them with respect to each other, on a possibly language-particular basis.
Three major syntactic phenomena have largely 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:

(13) Linearization (an algorithm or principle to determine the linear order of any two sister nodes)\(^6\)

(14) Locality of application (Relativized Minimality)

(15) The spellout of complex heads by the Morphology\(^7\)

---

\(^6\)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, \(\text{mutatis mutandis}\), 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 \(s\).