1 Chapter 4. Sections 1 – 3

Below an outline is given of the essential notions and definitions.

Lexical Entries

§ 1. The definition of a lexical entry is rather vague. Roughly, it can be identified with a feature structure as in GPSG or other theories. We have *features* and *values*. Each feature is associated with a range of values. A *feature matrix* is a set of pairs $\langle \text{FEAT}, \mathsf{val} \rangle$, where FEAT is a feature and val a value in the range of that feature. A *feature structure* is (α) a feature matrix or (β) a set of pairs $\langle \text{FEAT}, \mathsf{M} \rangle$ where FEAT is a feature and M a feature structure.

§ 2. A lexical entry λ is of the form

$$\begin{bmatrix} \text{P-FEAT} & : & \mathsf{M}_1 \\ \text{F-FEAT} & : & \mathsf{M}_2 \\ \text{S-FEAT} & : & \mathsf{M}_3 \end{bmatrix}$$

where P-FEAT, F-FEAT and L-FEAT are features. The value of P-FEAT is denoted by $\text{PF}(\lambda)$ and called the *phonetic matrix*, the value of F-FEAT is denoted by $\text{FF}(\lambda)$ and called the set of *formal features*, and the value of S-FEAT is denoted by $\text{LF}(\lambda)$ and called the *semantic form*. Notice that the phonetic matrix as well as the logical form are allowed to consist of features as well. However, the crucial assumption is that they play no role for C_{HL}.

Structures

§ 3. Objects are assumed to be formed from lexical entries using the simplest available constructors. Chomsky assumes set–constructors. The objects are formed by the computational system using the operation $\mathsf{substitute}$. We write $\bullet$ for it. The syntactic objects are of the form $K := \{\gamma, \{\alpha, \beta\}\}$; γ is called the *head* and α and β are the immediate constituents. We denote the head of K by $H(K)$.

$$K \bullet L := \begin{cases} \{H(K), \{K, L\}\} & \text{and } H(K \bullet L) := H(K) \\ \{H(L), \{K, L\}\} & \text{and } H(K \bullet L) := H(L) \end{cases}$$

Notice that substitution is not deterministic; it can produce either outputs. But depending on the choice, the head is defined. We may alternatively define two operations, $\bullet_1$ and $\bullet_2$, which make the first (second) term the head.

§ 4. Adjunction defines a different object. We write $K \circ L$ for the result of *adjoining* K to L or vice versa. The definition is

$$K \circ L := \begin{cases} \{\langle H(K), H(K)\rangle, \{K, L\}\} & \text{and } H(K \circ L) := K \\ \{\langle H(L), H(L)\rangle, \{K, L\}\} & \text{and } H(K \circ L) := L \end{cases}$$

Again we may distinguish between the operations $\circ_1$, $\circ_2$. Notice that the Kuratowski–Wiener definition of an ordered pair is $\langle a, b\rangle := \{a, \{a, b\}\}$ and so we have $\langle H(K), H(K)\rangle := \{H(K), \{H(K)\}\}$. It is important to realize that in case of adjunction the head is *not* identical to an immediate member of the set $K \circ L$.

§ 5. We call both $K \bullet L$ and $K \circ L$ the result of merge. Thus, merge has four options on a given pair K, L, depending on whether it chooses to substitute or to adjoin, and whether K is the head of the result or L. The head of the merger is said to *project* in the resulting structure. The notation basically achieves an encoding of a binary branching tree structure, in which the leaves carry the lexical entry, and each intermediate node is marked for the choice of operation that formed it. There are four choices. We can refer alternatively to the nodes of that tree by the term that formed it. We say that a term is a well–formed expression formed from lexical entries using one of the four operations $\bullet_1$, $\bullet_2$, $\circ_1$, $\circ_2$. A *tree* is a binary branching tree whose labels are terms, such that if x directly dominates y_1 and y_2, then the label of x is the result of merging the labels of y_1 and y_2. In a tree, the following notions can be defined. A *category* is a maximal subset C of that tree such that if $K \circ_1 L \in C$ then $K \in C$ and if $K \circ_2 L \in C$ then $L \in C$. The individual members of C are called *segments* of C. The category containing a lexical entry λ is called *minimal*. λ^{0max} is the highest segment of the category containing λ. A category is called *maximal* if its highest member is not a head.

Derivations

§ 6. A *multiset* is an unordered list; alternatively, a multiset over a set S is a function $f : S \to \omega$. For each $s \in S$, $f(s)$ denotes how many times s occurs in the list. Since the list is unordered, this information is sufficient. A multiset is distinguished from a set by subscript m. Thus, $\{a, b, b, c, a\}_m$ is a multiset, and distinct from $\{a, b, c\}_m$, but also distinct from $\{a, b, b, c, a\}$, which is the same as $\{a, b, c\}$. The union of multisets is denoted by $\cup_m$. It can be thought of as concatenation of the corresponding unordered lists. A multiset over the lexicon is called a *numeration* by Chomsky. The operation merge is defined on a multiset as follows.

$$\mathsf{merge}(\{K, L\}_m \cup_m S) := \{\mathsf{merge}(K, L)\}_m \cup_m S$$

This operation has various outputs, depending on the choice of K and L. Other transformations can be defined similarly, first on terms, and then on multisets of terms.

§ 7. Let Λ be a lexicon. A *derivation* over Λ is a finite sequence $\Delta = \mathsf{M}_1, \mathsf{M}_2, \mathsf{M}_3, \ldots, \mathsf{M}_n$ such that M_{i+1} is the result of applying a transformation on M_i, for all $1 \leq i < n$. In contrast to Chomsky's original definition we do not keep separate count of the numeration. Each lexical entry is assumed to be itself a term, and thus a legitimate part of any multiset in the derivation. The starting multiset is taken to be the *reference set*. There is a special operation, called *Spell–Out*. It operates on a *term* by removing the semantic features. Exactly how is not further specified. Clearly, for a lexical entry we can define

$$\mathsf{PF}(\lambda) := \{\langle \textsc{f-feat}, \textsc{ff}(\lambda)\rangle, \langle \textsc{p-feat}, \textsc{pf}(\lambda)\rangle\}$$

Then define for a term

$$\mathsf{PF}(K \star L) := \mathsf{PF}(K) \star \mathsf{PF}(L)$$

Finally, the operation can be defined on a multiset. Also, LF can be defined by

$$\mathsf{LF}(\lambda) := \{\langle \textsc{f-feat}, \textsc{ff}(\lambda)\rangle, \langle \textsc{s-feat}, \textsc{sf}(\lambda)\rangle\}$$

and lifting the definition to multisets of terms.

§ 8. A *syntactic derivation* is a triple of derivations $\langle \Delta_1, \Delta_2, \Delta_3\rangle$ such that (i.) Δ_1 begins with a multiset from Λ, and ends with M_n, (ii.) the beginning of Δ_2 is the result of applying LF to M_n, and (iii.) the beginning of Δ_3 is the result of applying PF to M_n. Let Δ_2 end in λ and Δ_3 end in π. Then the derivation is said to end in $\langle \lambda, \pi\rangle$. The derivation *meets the interface conditions* or *converges* if λ satisfies the conditions that the C–I interface specifies and π satisfies the conditions that the A–P interface specifies. If the derivation does not meet the interface conditions it is said to *crash*; a noncrashing derivation is called *convergent*. Given a multiset

N over Λ, $D(N)$ is the set of all derivations starting with N, $D_C(N)$ the set of convergent derivations in $D(N)$. There are a number of conditions that are formulated over the entire set $D_C(N)$, called economy conditions, and select a subset $D_A(N)$. Those define the correct pairs $\langle\lambda, \pi\rangle$ on the basis of N.

It is technically conceivable to define the Spell–Out in the following way. It is a step in a derivation where we have two ouptputs rather than one, each independently giving rise to a succeeding derivation. However, seen that way we would obtain numerous pairs $\langle\lambda, \pi\rangle$ even within a single derivation. It is then not clear how to select the output pairing defined from such a derivation. Rather, we assume that Spell–Out may operate only once, but in principle at any time.

§ 9. Given a multiset with n entries, there exist up to $\frac{1}{n}\binom{2n-2}{n-1}$ binary branching trees carrying these as terminal labels. This number is exponential, approaching $\frac{4^{n-1}}{\sqrt{\pi\cdot(n-1)}}$ in the limit. Thus it is roughly in the magnitude 4^{n-1}. Given that merge has four independent options to join two subtrees, we get 16^{n-1} output trees in the worst case. (The reason is that a binary branching tree with n leaves has $n-1$ nonleaves, and this gives an additional factor 4 per nonleaf, in total 4^{n-1}.) Thus, if economy principles are global principles, that is to say, can only be computed after completion of all derivations, then computation time is exponential in the size of the numeration.