6

How the Grammar Works

6.1 A Factorization of Grammatical Information

Three chapters ago, we began modifying the formalism of context-free grammar to better adapt it to the sorts of generalizations we find in natural languages. We broke grammatical categories down into features, and then we broke the values of features down into features, as well. In the process, we moved more and more syntactic information out of the grammar rules and into the lexicon. In effect, we changed our theory of grammar so that the rules give only very general patterns that cut across grammatical categories. Details about which expressions can go with which are specified in lexical entries in terms of valence features.

With the expanded ability of our new feature structure complexes to express cross-categorial generalizations, our four remaining grammar rules cover a wide range of cases. Two of them – the rules introducing complements and specifiers – were discussed extensively in Chapter 4. The third one – a generalization of our old rules introducing PP modifiers to VP and NOM – was illustrated in the previous chapter.[1] The fourth is the Coordination Rule. The formal statements of these rules were given at the end of the previous chapter, along with informal translations (given in italics below the rules).

In addition to our grammar rules, we must provide (as we did in the case of CFGs) some characterization of the 'initial symbol', corresponding to the type of phrases that can stand alone as sentences of the language. We postpone a careful characterization of this until Chapter 8, when we will have introduced a method for distinguishing finite (that is, tensed) clauses from others. For now, we can treat S (which we characterized in terms of features in Chapter 4) as the initial symbol.

We were able to make our grammar rules so general in part because we formulated four general principles about how information must be distributed in well-formed trees: the Head Feature Principle, the Valence Principle, the Semantic Compositionality Principle, and the Semantic Inheritance Principle. These were also reiterated at the end of Chapter 5.

The richer feature structures we are now using, together with our highly schematized rules, have required us to refine our notion of how a grammar is related to the fully

[1] It should be noted that the Head-Modifier Rule does not cover all kinds of modifiers. In particular, some modifiers – such as adjectives inside NPs – precede the heads that they modify. To accommodate such modifiers, we would need an additional grammar rule. This issue was addressed in Problem 1 of Chapter 5.

determinate phrase structure trees of the language. Intuitively, here is how it works:

First, each lexical entry licenses a family of word structures – each of which is a nonbranching tree. More precisely, a lexical entry $\langle \omega, \Phi \rangle$ licenses any word structure of the form:

$$\begin{array}{c} \text{F} \\ | \\ \omega \end{array}$$

if and only if F is a resolved feature structure that satisfies Φ. A resolved feature structure F satisfies Φ if and only if it assigns values to all features appropriate for feature structures of its type, and those values are consistent with all of the information specified in Φ.

Such lexical trees form the bottom layer of well-formed phrasal trees. They can be combined[2] into larger trees in the ways permitted by the grammar rules, obeying the constraints imposed by our four principles. This process can apply to its own output, making ever larger phrasal trees. So long as the local tree at the top of each tree structure that we construct is licensed by a grammar rule and conforms to these principles, it is well formed. Typically, each node in a well-formed tree will contain some information that was stipulated by a rule and other information that percolated up (metaphorically speaking) from lower nodes (and ultimately from the lexical entries) via the principles. In summary, the relation between our trees and the grammatical mechanisms that license them is as follows: a tree is well-formed if, and only if, it satisfies all of the conditions imposed by the lexical entries of the words it contains, by the grammar rules, and by the general grammatical principles.

We have formulated our theory so that the number of tree structures consistent with a given terminal string will shrink considerably as constraints from higher levels of structure are brought into the picture. This important effect of contextual constraints can be illustrated with the CASE value of proper nouns. Consider the lexical entry in (1):

(1)
$$\left\langle \text{Leslie}\,, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Leslie} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

[2]Our informal discussion is worded in terms of a process of building trees up from the bottom. This is a conceptually natural way of thinking about it, but it should not be taken too literally. The formal definition of well-formed tree structure that we give below is deliberately nonprocedural.

This lexical entry gives fully specified values for every feature except CASE and GEND. (It may look underspecified for PER and NUM as well, but recall that the type *3sing* is constrained to have specific values for each of those features.) Since the features CASE and GEND are left underspecified in the lexical entry, the lexical entry licenses six distinct word structures. We have shown two in (2):

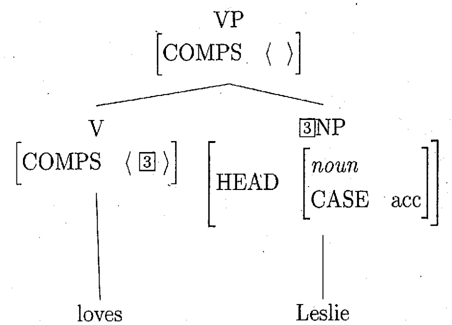
(2) a.
$$\begin{array}{c} \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{CASE} & \text{nom} \\ \text{AGR} & \begin{bmatrix} \textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{fem} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Leslie} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ | \\ \text{Leslie} \end{array}$$

b.
$$\begin{array}{c} \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{masc} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Leslie} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ | \\ \text{Leslie} \end{array}$$

Notice that we could have abbreviated the mother of these tree structures either as 'N' or as 'NP', since this is a node of type *word* whose HEAD value is of type *noun* with empty SPR and COMPS lists.

Although these two word structures both satisfy the constraints given in the lexical entry equally well, only the tree in (2b) can be embedded within a larger one like (3), licensed by the Head-Complement Rule:

(3)



That is because we have assumed here (following the results of Chapter 4, Problem 6) that the lexical entry for *loves* specifies that its complement is [CASE acc]. Because the Head-Complement Rule identifies the head daughter's COMPS list with the list of (the feature structures of the) complement daughters, the accusative case specification must be part of the object noun's HEAD value.[3]

The information specified by our rules and lexical entries is thus partial information. Each rule says, in effect, that subtrees of a certain kind are sanctioned, but the rule only specifies some of the constraints that the trees that it licenses must obey. Likewise, a lexical entry says that certain trees dominating the phonological form in that entry are sanctioned, but the entry only specifies some of the information relevant at higher levels of structure. The general principles of our theory constrain the ways in which feature values can be distributed in well-formed phrase structure trees. The job of determining well-formedness can be distributed among the various pieces of our grammatical system because the licensing mechanism requires simultaneous satisfaction of all of the relevant constraints.

In developing our grammar so far, we have arrived at a particular factorization of the information necessary for a precise account of grammatical structure. By far the richest source of information in this factorization is the lexicon. That is, our grammar embodies the claim that both the problem of determining which strings of words constitute well-formed sentences and the problem of specifying the linguistic meaning of sentences depend mostly on the nature of words. Of course, it must also be recognized that there are many regularities about which words go together (and how they go together). The theoretical constructs summarized here capture a number of such regularities; subsequent chapters will provide ways of capturing more.

[3]Nothing in the syntactic context constrains the GEND value, however. The appropriate value there will depend on the non-linguistic context, in particular, on the gender of the person the speaker intends to refer to.

6.2 Examples

6.2.1 A Detailed Example

The best way to understand how the various components of our grammatical theory interact is to work through detailed analyses of linguistic examples. In this subsection, we show in detail how the grammar of English, as we have developed it to this point, handles one simple sentence of English, namely:[4]

(4) They sent us a letter.

We begin our lexical analysis with the entry for the word *letter*:

(5)
$$\left\langle \text{letter}\ , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll}\textit{noun} & \\ \text{AGR} & \left[\begin{array}{ll}\textit{3sing} & \\ \text{GEND} & \text{neut}\end{array}\right]\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \left\langle \begin{array}{c}\text{D}\\ \left[\begin{array}{ll}\text{COUNT} & + \\ \text{INDEX} & k\end{array}\right]\end{array}\right\rangle \\ \text{COMPS} & \langle (\text{PP}_m) \rangle \\ \text{MOD} & \langle\ \rangle \end{array}\right]\end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll}\text{MODE} & \text{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \left[\begin{array}{ll}\text{RELN} & \textbf{letter} \\ \text{INST} & k \\ \text{ADDRESSEE} & m\end{array}\right]\right\rangle\end{array}\right]\end{array}\right] \right\rangle$$

We assume *letter* optionally selects a PP complement, as indicated.

How many word structures satisfy (5)? The answer to this question may be surprising. There are *infinitely many* word structures that satisfy (5). Moreover, this will be true whenever a lexical entry selects something on its COMPS or SPR list, because lexical entries specify such minimal information about the things they select for. For example, in the absence of further constraints, the member of the SPR list in a word structure licensed by (5) could have a RESTR list of any length. Similarly, if the COMPS list in the word structure contains a PP, that PP could have a RESTR value of any length. And this is as it should be, as there is no upper bound on the length of PP complements of this word:

[4]In this section, we present the details of trees over the course of several pages, depicting various subtrees and how they fit together to make larger trees. In doing this, we use tags to mark identity across distinct diagrams of trees that will eventually be put together into a single tree. We also reuse tags across different trees when the same lexical entry is used in different sentences. Strictly speaking, tags only mark identity within a given description. We are taking this liberty with the tag notation only in this section, because it is a convenient heuristic.

(6) a. the letter to Kim...
 b. the letter to Kim and Sandy...
 c. the letter to Kim, Lee and Sandy...
 d. the letter to the person who signed the document that started the mishap that...

That is, depending on the surrounding context (i.e. depending on which words the PP actually contains), the PP's RESTR list might have one, three, thirty-seven, or two hundred predications on it. The same is true of the specifier, as the examples in (7) indicate:

(7) a. the letter...
 b. almost every letter...
 c. Sandy's friend's mother's letter...
 d. the cricket club's former secretary's letter...

If we assume the analysis of quantificational determiners sketched at the end of Chapter 5, then the word structure for *letter* that is relevant to the sentence in (4), however, has a SPR value whose RESTR is singleton:

(8)
$$
\begin{bmatrix}
\textit{word} & \\
\text{SYN} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1}\begin{bmatrix}\textit{3sing} & \\ \text{PER} & \text{3rd}\\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{neut}\end{bmatrix} \\ \text{CASE} & \text{acc}\end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{det} & \\ \text{AGR} & \boxed{1} \\ \text{COUNT} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{none}\\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{exist} \\ \text{BV} & k\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix}
\end{bmatrix} \\
\text{SEM} & \begin{bmatrix}\text{MODE} & \text{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle\begin{bmatrix}\text{RELN} & \textbf{letter}\\ \text{INST} & k \\ \text{ADDRESSEE} & m\end{bmatrix}\right\rangle\end{bmatrix}
\end{bmatrix}
$$

letter

As for the COMPS value, the empty list option has been exercised in this tree, as the sentence whose structure we are building contains no PP complement. Notice that, with no PP, there is no constituent that will realize the ADDRESSEE role. Since we have not imposed any constraint requiring that semantic roles be realized syntactically, this does not present any technical problem. And having an ADDRESSEE role for the noun *letter*, even when no addressee is mentioned, seems quite intuitive. Finally, note that (8) obeys the Specifier-Head Agreement Constraint, which identifies the AGR value of the noun with that of the element on its SPR list.

The word structure for the word *a* is abbreviated in (9):[5]

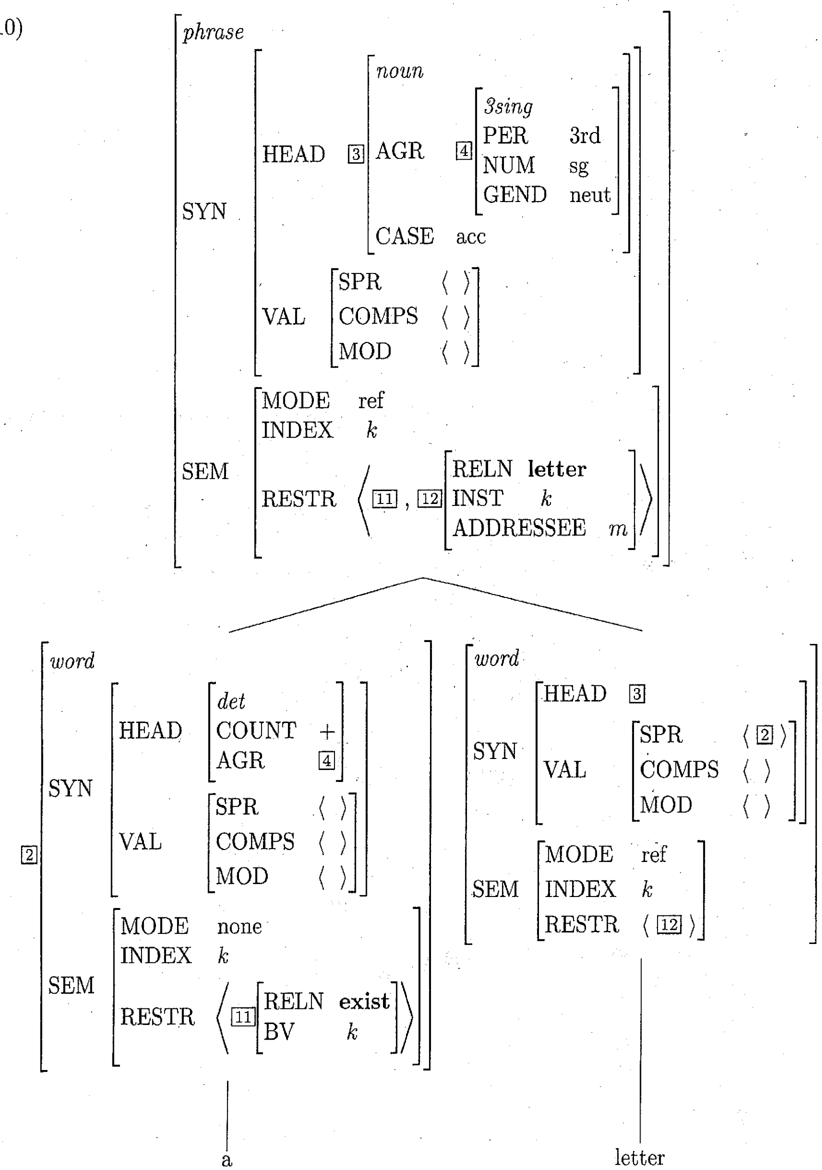
(9)
$$
\begin{bmatrix}
\textit{word} & \\
\text{SYN} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix}\textit{det} & \\ \text{AGR} & \begin{bmatrix}\textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{neut}\end{bmatrix} \\ \text{COUNT} & + \end{bmatrix} \\
\text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle\end{bmatrix}
\end{bmatrix} \\
\text{SEM} & \begin{bmatrix}\text{MODE} & \text{none} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle\begin{bmatrix}\text{RELN} & \textbf{exist} \\ \text{BV} & k\end{bmatrix}\right\rangle\end{bmatrix}
\end{bmatrix}
$$

a

[5]What is not shown in this tree is the complete feature specification for the **exist** predication. See Section 5.8 of Chapter 5 for discussion.

The following tree results from combining (8) and (9) via the Head-Specifier Rule:

(10)



In this tree, the left subtree is exactly the one shown in (9). The identification of the element on the head daughter's SPR list (2) and the feature structure of the left daughter is guaranteed by the Head-Specifier Rule, which licenses the combination of this determiner with this noun. When the Head-Specifier Rule enforces this identity, it forms a link in a chain of identities: the lexical entry for *letter* identifies the INDEX of the element

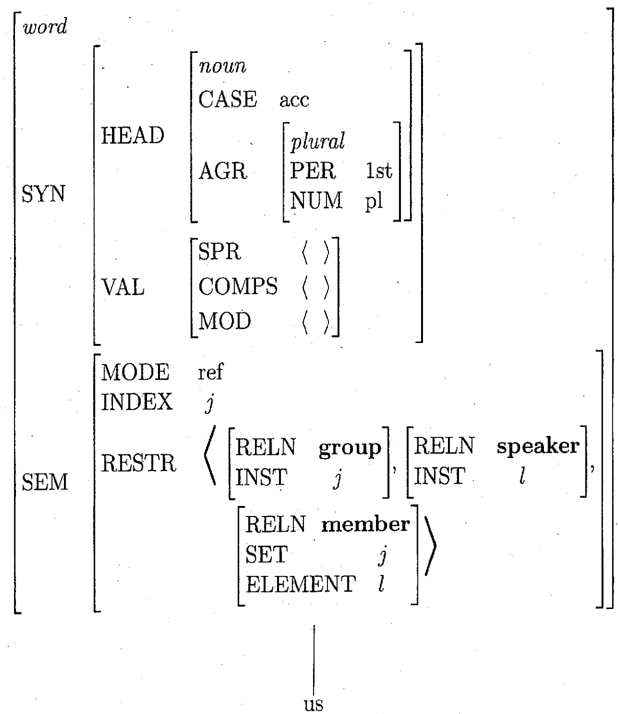
on its SPR list with its own INDEX and INST values. The lexical entry for *a* identifies its INDEX with its BV value. When these two words combine via the Head-Specifier Rule, the INDEX of the specifier of *letter* and the INDEX of *a* are identified. This chain of identities ensures that the BV of the **exist** predication and the INST of the **letter** predication are one and the same (k).

(10) obeys the HFP: the HEAD value of the head daughter is identified with that of the mother (3). And it obeys the Valence Principle: the COMPS value of the phrase is the same as that of the head daughter (the empty list). The mother's SPR value is the empty list, as required by the Head-Specifier Rule.

The Semantic Inheritance Principle says that the MODE and INDEX values of the head daughter must be shared by the mother, which is the case in (10). And the Semantic Compositionality Principle requires that the mother's RESTR value be the sum of the two daughters' RESTR lists. This concludes the analysis of the noun phrase *a letter*, as it appears in the sentence in (4).

The lexical entry for the pronoun *us* is quite straightforward, except for the RESTR list in the semantics. In the following, we have chosen to characterize the meaning of *us* roughly as reference to a group of which the speaker is a member. We have formalized this as a RESTR list with three elements, but there are many other possible ways of doing this. Our version gives rise to the following lexical tree:

(11)



All this information is lexically specified. Note that because the AGR value is of type *plural*, it contains no GEND specification.

Now consider the lexical entry for the word *sent*:[6]

(12)

$$\left\langle \text{sent}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{matrix} \text{NP}_i \\ \begin{bmatrix}\text{CASE} & \text{nom}\end{bmatrix} \end{matrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{matrix} \text{NP}_j \\ \begin{bmatrix}\text{CASE} & \text{acc}\end{bmatrix} \end{matrix} , \begin{matrix} \text{NP}_k \\ \begin{bmatrix}\text{CASE} & \text{acc}\end{bmatrix} \end{matrix} \right\rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_7 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{send} \\ \text{SIT} & s_7 \\ \text{SENDER} & i \\ \text{SENDEE} & j \\ \text{SENT} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Note that, as a past tense form, this lexical entry has an underspecified AGR value. All of the word structures licensed by (12), however, have fully resolved AGR values, and by the SHAC, must share those AGR values with their specifiers. Similarly, although the lexical entry in (12) places no restrictions on the AGR value of the complements, those AGR values are fully specified in the word structures. The word structure for *sent* that is relevant to the sentence in (4) is shown in (13):[7]

[6] We are ignoring the semantic contribution of the past tense in this discussion.

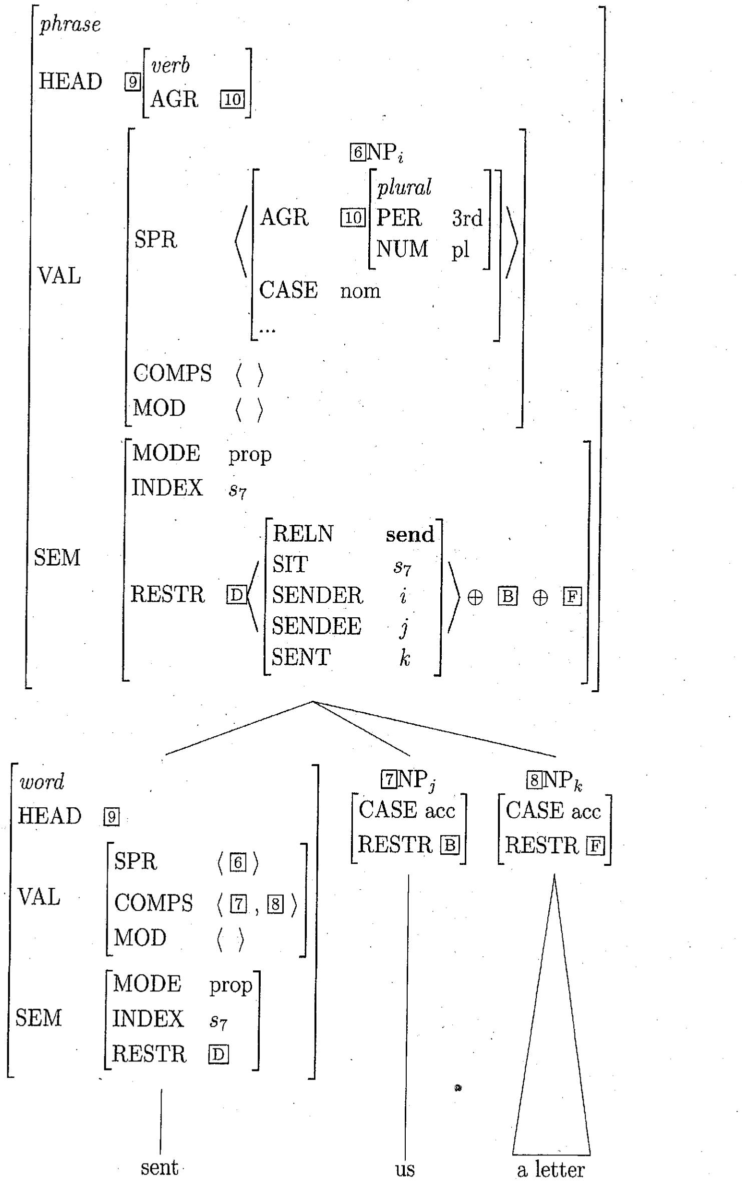
[7] Although the tree in (13) represents a fully resolved word structure, we have abbreviated somewhat. In particular, we have not shown the SEM values within the elements of the SPR and COMPS lists. Similar remarks apply to many of the trees in the remainder of this chapter.

(13)

$$\begin{array}{c} \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{10} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{matrix} \text{NP}_i \\ \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \boxed{10} \begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{pl} \end{bmatrix} \\ \ldots & \end{bmatrix} \end{matrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{matrix} \text{NP}_j \\ \begin{bmatrix} \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{plural} & \\ \text{PER} & \text{1st} \\ \text{NUM} & \text{pl} \end{bmatrix} \\ \ldots & \end{bmatrix} \end{matrix} , \begin{matrix} \text{NP}_k \\ \begin{bmatrix} \text{CASE} & \text{acc} \\ \text{AGR} & \begin{bmatrix} \textit{3sing} & \\ \text{PER} & \text{3rd} \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{neut} \end{bmatrix} \\ \ldots & \end{bmatrix} \end{matrix} \right\rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_7 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{send} \\ \text{SIT} & s_7 \\ \text{SENDER} & i \\ \text{SENDEE} & j \\ \text{SENT} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ | \\ \text{sent} \end{array}$$

The three trees we have now built up combine via the Head-Complement Rule to give the following tree structure:

(14)

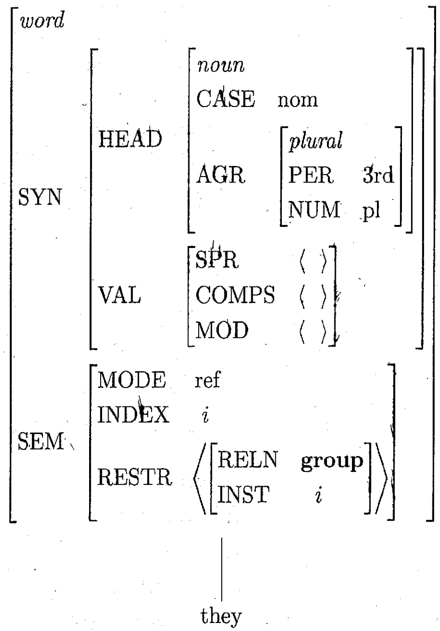


We have done a bit more abbreviating here. The node tagged 7 is identical to the top node of the word structure in (11). Likewise, the node tagged 8 is identical to the top node in (10).

The [CASE acc] constraints on both these NPs comes from the COMPS value of the lexical entry for *sent* (see (12)), and hence appears on this node, as required by the Head-Complement Rule. The RESTR values in the semantics for the two NP nodes are the ones shown in (11) and (10). We abbreviated these with the tags B and F, respectively. (14) obeys the conditions on COMPS values specified in the Head-Complement Rule, that is, the head daughter's complements are identified with the non-head daughters and the mother's COMPS value is empty. (14) obeys the Valence Principle, as the SPR value of the head daughter, not mentioned in the rule, is preserved as the mother's SPR value. Likewise, the HEAD value of mother and head daughter are correctly identified here, in accordance with the Head Feature Principle. Finally, the MODE and INDEX values of the mother are those of the head daughter, while the RESTR value of the mother is the sum of those of all the daughters, as specified by the semantic principles.

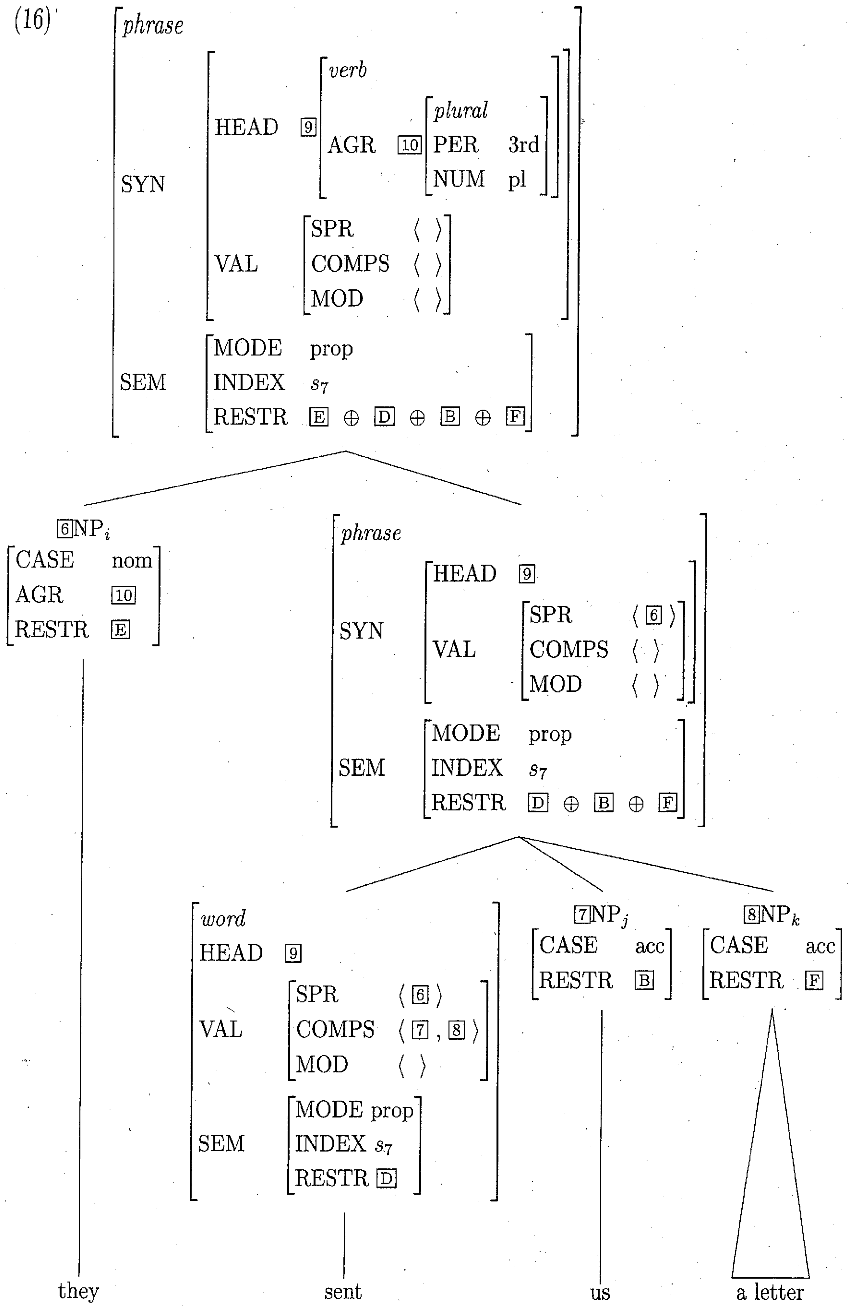
The last step is to combine the VP in (14) with the tree structure for its subject NP. The following is the word structure for the pronoun *they*, as licensed by an appropriate lexical entry:

(15)



The result is the tree in (16):

(16)



Again, we have abbreviated. The node labeled $\boxed{6}$ is just the top node in (15). The nodes labeled $\boxed{7}$ and $\boxed{8}$ are exactly as they were in (14), as is the VP node. We have abbreviated the RESTR values, simply putting in tags or sums of tags. The RESTR value of the top node, fully spelled out (except for the somewhat abbreviated contribution of the word *a*), is the list consisting of the following seven predications (in the indicated order):

(17) $$\begin{bmatrix}\text{RELN} & \textbf{group}\\ \text{INST} & i\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{send}\\ \text{SIT} & s_7\\ \text{SENDER} & i\\ \text{SENDEE} & j\\ \text{SENT} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{group}\\ \text{INST} & j\end{bmatrix},$$

$$\begin{bmatrix}\text{RELN} & \textbf{speaker}\\ \text{INST} & l\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{member}\\ \text{SET} & j\\ \text{ELEMENT} & l\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{exist}\\ \text{BV} & k\end{bmatrix},$$

$$\begin{bmatrix}\text{RELN} & \textbf{letter}\\ \text{INST} & k\\ \text{ADDRESSEE} & m\end{bmatrix}$$

The AGR value in the top node of (16) is identical to that in the subject NP, as required by the interaction of the HFP, the Head-Specifier Rule, and the SHAC. In general, this tree structure obeys the Head Feature Principle, the Valence Principle, and the two semantic principles.

This concludes our analysis of the sentence *They sent us a letter.* The various constraints in our grammar interact to ensure that this structure and infinitely many related to it are well-formed, while guaranteeing that infinitely many other structures similar to it are ill-formed.

Exercise 1: The Non-infinity of *Us*

The lexical entry for *letter* licenses infinitely many word structures, while the lexical entry for *us* licenses exactly one. What feature specifications in the lexical entries are behind this difference?

6.2.2 Another Example

The detailed analysis we just went through built the sentence from the bottom up. This is one way to use the grammatical machinery we have developed, but it is not the only way. We could equally well have started with at the top of the tree, showing how our rules, principles, and lexical entries interact to license all its parts.

To see this top-down approach in action, consider the following sentence:[8]

(18) We send two letters to Lee.

[8]This example sounds a bit odd in isolation, but it would be perfectly natural in the appropriate context, for example, in response to the question, *What do we do if Alex writes to us?*

Example (18) is structurally ambiguous in a way analogous to the familiar example, *I saw the astronomer with a telescope.* That is, the PP *to Lee* can be attached either to the VP or to the NP headed by *letters*. In our semantic representation, the two readings correspond to two different RESTR lists, shown in (19) and (20):

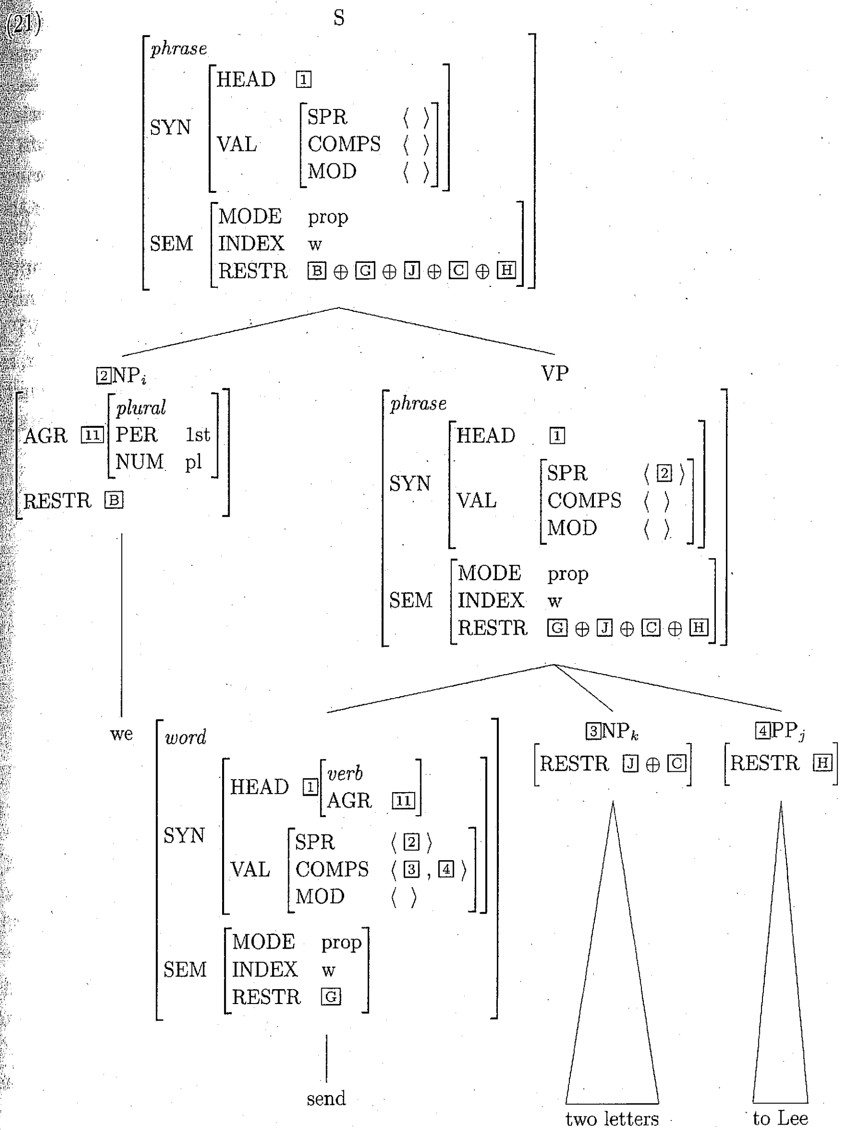
(19)
$$\begin{bmatrix}\text{RELN} & \textbf{group}\\ \text{INST} & i\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{speaker}\\ \text{INST} & l\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{member}\\ \text{SET} & i\\ \text{ELEMENT} & l\end{bmatrix},$$
$$\begin{bmatrix}\text{RELN} & \textbf{send}\\ \text{SIT} & s_7\\ \text{SENDER} & i\\ \text{SENDEE} & j\\ \text{SENT} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{two}\\ \text{BV} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{letter}\\ \text{INST} & k\\ \text{ADDRESSEE} & m\end{bmatrix},$$
$$\begin{bmatrix}\text{RELN} & \textbf{name}\\ \text{NAME} & \text{Lee}\\ \text{NAMED} & j\end{bmatrix}$$

(20)
$$\begin{bmatrix}\text{RELN} & \textbf{group}\\ \text{INST} & i\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{speaker}\\ \text{INST} & l\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{member}\\ \text{SET} & i\\ \text{ELEMENT} & l\end{bmatrix},$$
$$\begin{bmatrix}\text{RELN} & \textbf{send}\\ \text{SIT} & s_7\\ \text{SENDER} & i\\ \text{SENDEE} & j\\ \text{SENT} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{two}\\ \text{BV} & k\end{bmatrix}, \begin{bmatrix}\text{RELN} & \textbf{letter}\\ \text{INST} & k\\ \text{ADDRESSEE} & m\end{bmatrix},$$
$$\begin{bmatrix}\text{RELN} & \textbf{name}\\ \text{NAME} & \text{Lee}\\ \text{NAMED} & m\end{bmatrix}$$

The only difference between the two semantic representations is which other role the NAMED value of the **name** predication (i.e. Lee) is identified with: the SENDEE value of the **send** predication or the ADDRESSEE value of the **letter** predication.

In this subsection, we will show how our grammar licenses two distinct trees for this sentence, and how it associates each with one of the semantic representations in (19) and (20). For expository convenience, we begin with the rather schematic tree in (21) (similar to (16)), waiting to show the detailed feature structures it contains until we look at its subtrees:

(21)



The top node in this tree is licensed by the Head-Specifier Rule. It differs from its second daughter, the VP, in only two ways: its SPR value is the empty list (as required by the Head-Specifier Rule), and its RESTR value includes the RESTR of the subject NP (as required by the Semantic Compositionality Principle). The HEAD features of the top node and of the VP are identical, as required by the Head Feature Principle. The COMPS list is empty both at the top and in the VP, in accordance with the Valence Principle. And both MODE and INDEX have the same value at the top as in the VP, in keeping with the Semantic Inheritance Principle. The first daughter (the subject NP) is identical to the sole element on the second daughter's SPR list, as required by the Head-Specifier Rule.

The subtree dominating *we* – that is the subject of the sentence – is labeled 'NP' here, but it could just as well have been labeled 'N'. It is simply a word structure, identical in its feature structure to the one in (11), except that the value of the CASE feature is 'nom', not 'acc'. This structure is the word structure licensed by the lexical entry for *we*.

The other daughter of the top node – the VP – is the mother of a tree licensed by the Head-Complement Rule. The VP's feature values are the same as those of its head (leftmost) daughter, except for COMPS and RESTR. The COMPS list of the VP is empty, as specified in the Head-Complement Rule. The RESTR value is the sum of its three daughters' RESTR values, by the Semantic Compositionality Principle. Again, the VP's HEAD, SPR, MODE, and INDEX values are the same as those of the head daughter, in accordance with the HFP, the Valence Principle, and the Semantic Inheritance Principle. The COMPS value of the head daughter is the list consisting of the other two daughters; this is specified by the Head-Complement Rule.

The subtree dominating the verb *send* is the following:

(22)
$$\left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll}\textit{verb} & \\ \text{AGR} & \boxed{11}\end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \left\langle \begin{array}{c}\text{NP}_i\\ \left[\begin{array}{ll} \text{CASE} & \text{nom}\\ \text{AGR} & \boxed{11}\left[\begin{array}{ll}\textit{plural} & \\ \text{PER} & \text{1st}\\ \text{NUM} & \text{pl}\end{array}\right]\\ \ldots & \end{array}\right]\end{array}\right\rangle \\ \text{COMPS} & \left\langle \begin{array}{c}\text{NP}_k\\ \left[\begin{array}{ll} \text{CASE} & \text{acc}\\ \text{AGR} & \left[\begin{array}{ll}\textit{plural} & \\ \text{PER} & \text{3rd}\\ \text{NUM} & \text{pl}\end{array}\right]\\ \ldots & \end{array}\right]\end{array}, \text{PP}_j \right\rangle \\ \text{MOD} & \langle\ \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop}\\ \text{INDEX} & s_7\\ \text{RESTR} & \boxed{G}\left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{send}\\ \text{SIT} & s_7\\ \text{SENDER} & i\\ \text{SENDEE} & j\\ \text{SENT} & k\end{array}\right]\right\rangle \end{array}\right] \end{array}\right]$$

send

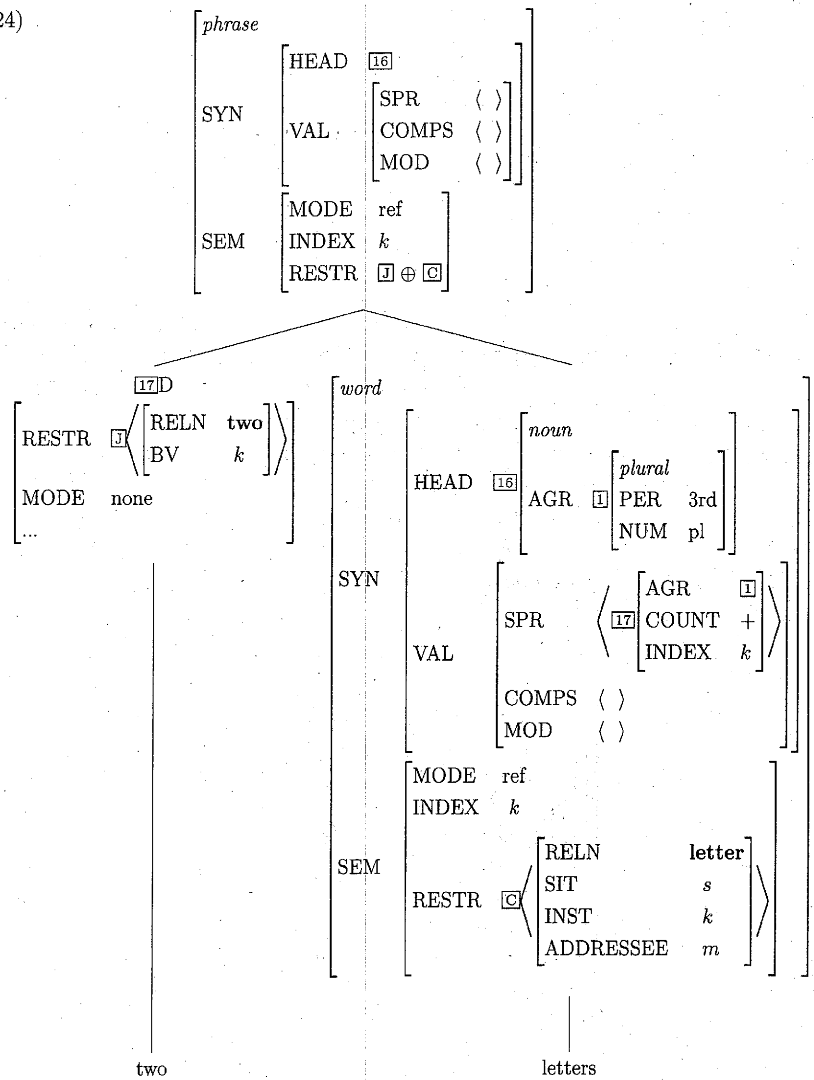
This is different from the verb subtree in our previous example (i.e. from (13)) in several ways. The most obvious is that the form is *send*, not *sent*. Although our SEM value does not reflect the clear meaning difference between the present and past tense forms, there are nonetheless several syntactic differences that are represented. Many of these differences follow from differences in the lexical entries that license the word structures. (22) is licensed by the lexical entry in (23):

(23)
$$\left\langle \text{send}\ , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb}\\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \left\langle \begin{array}{c}\text{NP}_i\\ \left[\begin{array}{ll}\text{CASE} & \text{nom}\\ \text{AGR} & \textit{non-3sing}\end{array}\right]\end{array}\right\rangle \\ \text{COMPS} & \left\langle \begin{array}{c}\text{NP}_k\\ \left[\text{CASE}\ \ \text{acc}\right]\end{array} (, \text{PP}_j) \right\rangle \\ \text{MOD} & \langle\ \rangle \end{array}\right] \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop}\\ \text{INDEX} & s_7\\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{send}\\ \text{SIT} & s_7\\ \text{SENDER} & i\\ \text{SENDEE} & j\\ \text{SENT} & k\end{array}\right]\right\rangle \end{array}\right] \end{array}\right] \right\rangle$$

(23)'s specifier is specified as [AGR *non-3sing*]; that is because the verb *send* (unlike *sent*) cannot be combined with a third-person singular subject (like *Terry*). Another difference is that the second element of the COMPS list in (22) is an optional PP, not an obligatory NP. Related to that is the fact that the first complement in (22) refers to the thing sent (indicated by the role 'SENT' in the predication on the verb's RESTR list), and the second complement corresponds to the sendee (also indicated in the RESTR). Problem 3 in Chapter 10 addresses the relation between pairs of lexical entries like (12) and (23).

The subtree for the object NP, *two letters*, is shown in (24):

(24)

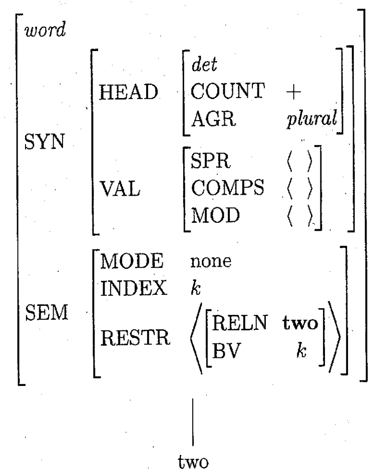


This tree is licensed by the Head-Specifier Rule, which says that the top node must have an empty SPR list and that the second (i.e. head) daughter must have a SPR list whose sole member is identical to the first daughter. The identity of the AGR values of the head noun *letters* and its determiner *two* (indicated by 1) is required by the SHAC. The HEAD value of the top node is identical to that of the second daughter, according to the Head Feature Principle. The COMPS values of these two nodes are identical, as guaranteed by the Valence Principle. The MODE and INDEX values of the second daughter and its mother are likewise shared, courtesy of the Semantic Inheritance

Principle. Finally, the Semantic Compositionality Principle requires that the RESTR value of the determiner combines with the RESTR value for the noun to give the RESTR value of the NP.

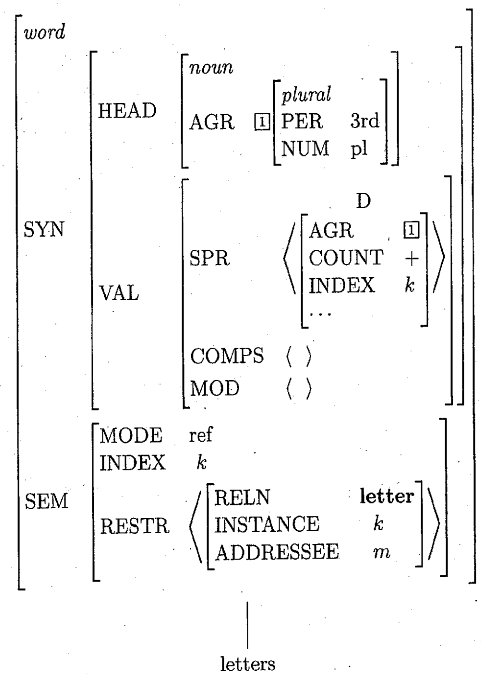
Licensing (24) via the Head-Specifier Rule requires the word structures for each of its words. The following is the word structure for *two*, which is similar to (9) above:

(25)



The relevant word structure for *letters* is sketched in (26):

(26)



This tree is quite similar to (8). The principle difference is that the type of the AGR value is *plural*, and it therefore lacks the GEND feature. If our treatment of semantics were more detailed, the RESTR value would also be different, since it would have to include some information about the meaning of the plurality of *letters*; but for present purposes, we will ignore that difference. This word structure is licensed by the entry for *letters*, shown in (27):

(27)

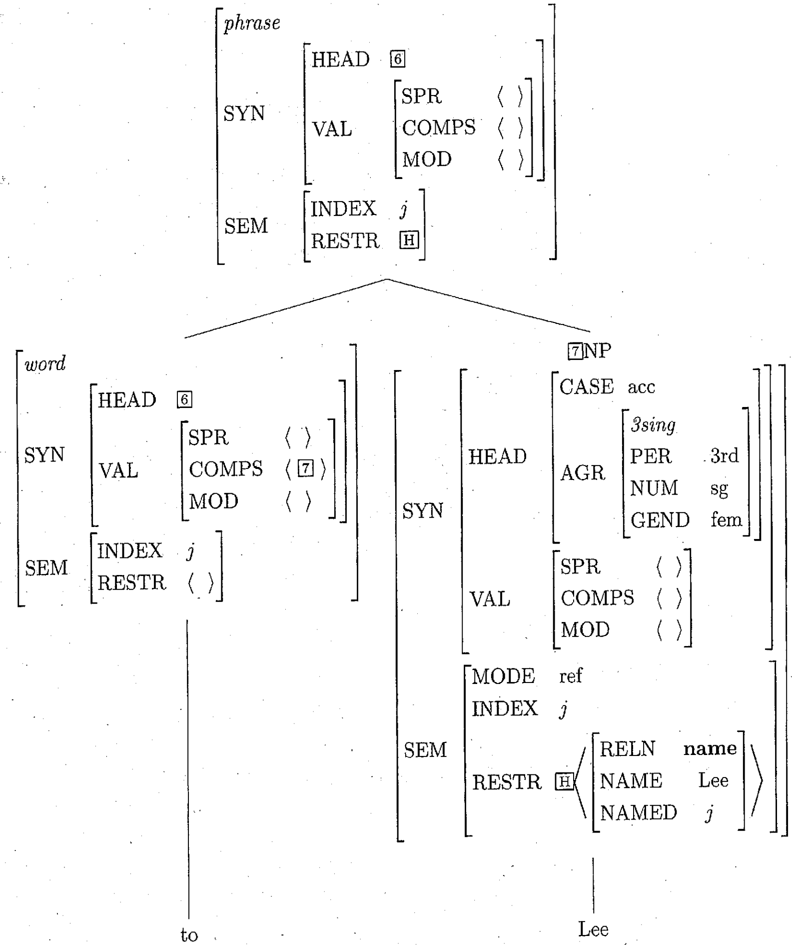
$$\left\langle \text{letters}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{plural} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \overset{\text{D}}{\begin{bmatrix} \text{COUNT} & + \\ \text{INDEX} & k \end{bmatrix}} \right\rangle \\ \text{COMPS} & \langle\ (\text{PP}_m)\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{letter} \\ \text{INST} & k \\ \text{ADDRESSEE} & m \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Notice that this lexical entry, like the one for *letter* in (5), provides for a possible PP complement. The word structure in (26) above uses the empty COMPS list option. We will return to the PP complement possibility below.

The subtree for the PP, *to Lee*, is highly schematized in (21). A more detailed version of the tree is given in (28):[9]

[9]As with the proper noun *Leslie* discussed in Section 6.1 above, the lexical entry for *Lee* is underspecified for GEND. All of the word structures that satisfy that lexical entry are fully specified, and therefore contain a value for GEND. Here we have arbitrarily chosen a word structure that is [GEND fem].

(28)



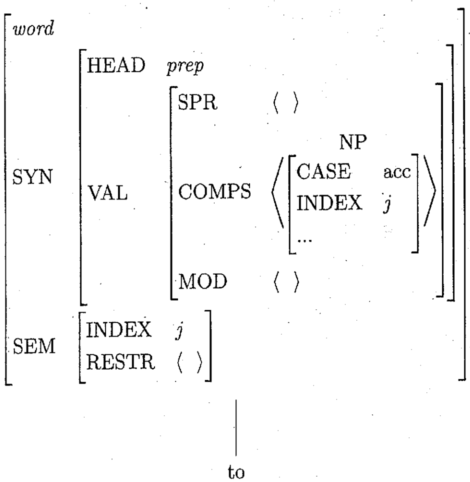
The most interesting thing about this subtree is how we have analyzed the semantics. The preposition *to* in this sentence is functioning to mark the role of its object NP with respect to the verb. That is, it does what many languages would do by means of case inflections on the noun. Since English has only a vestigial system of case marking, it relies on prepositions and word order to mark the roles of various NPs in the sentence. Note that the preposition can be omitted if the verb's arguments are presented in another order: *We sent Lee two letters.* Consequently, we have given the preposition no semantics of its own. Its RESTR value is the empty list, and its index is simply identified as the index of the object NP. We have said nothing about the MODE value, but in the next chapter, we will argue that it, too, should be identified with the MODE of the object NP.

The PP assumes the same INDEX value as the preposition (and hence as the NP) by the Semantic Inheritance Principle. Other identities in (28) should by now be familiar:

the one element of the preposition's COMPS list must be the object NP, by the Head-Complement Rule; the same rule specifies that the PP has an empty COMPS list; the Valence Principle is responsible for the fact that the PP and P have the same (empty) SPR list; the PP and the P share the same HEAD features in virtue of the Head Feature Principle; and the PP's RESTR value is the same as the NP's, in accordance with the Semantic Compositionality Principle (together with the fact that the preposition's RESTR is the empty list).

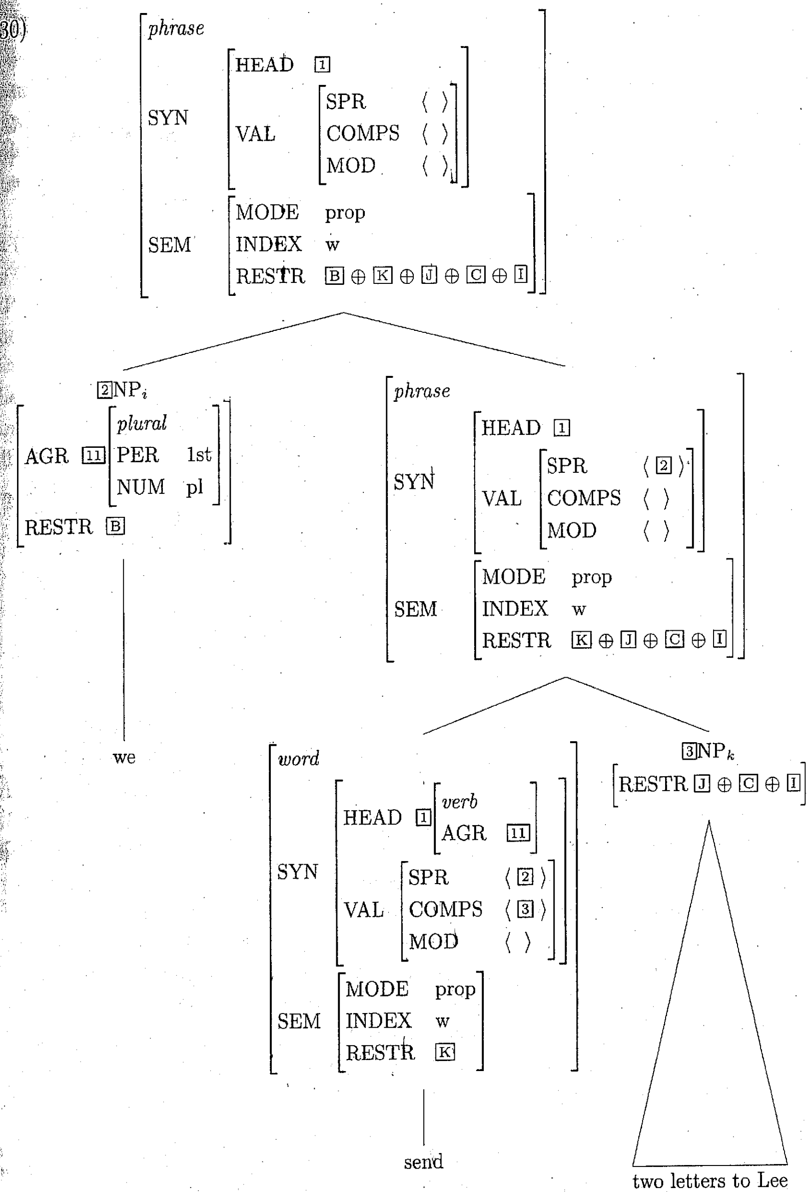
The NP in (28) is [CASE acc] because objects of prepositions in English are always accusative (although there is no morphological marking of it in this sentence). This requirement is encoded in the lexical entry for the preposition, as we will see when we look at the word structure for *to*, which is shown in (29):

(29)



This completes the analysis of one parse of *We send two letters to Lee.* A schematic tree for the other parse is given as (30):

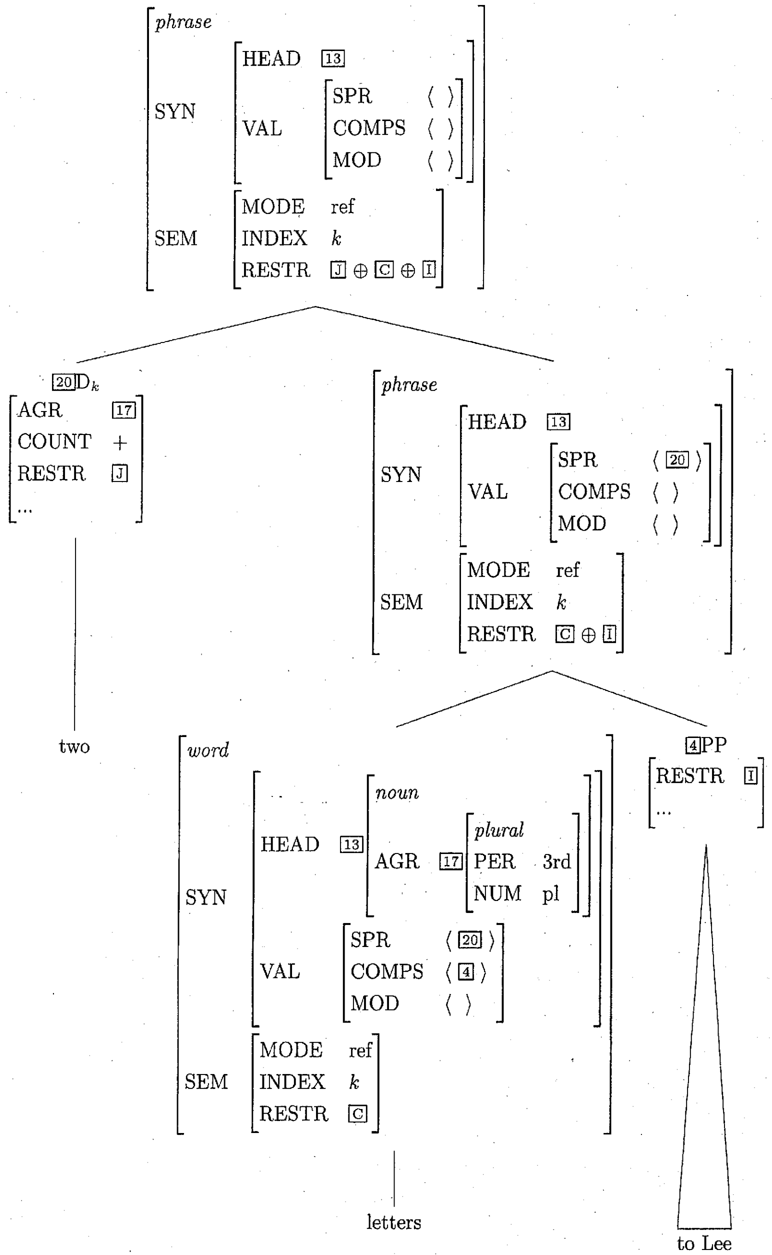
(30)



The subject NP *we* and the PP *to Lee* are exactly the same in this structure as in (21). The verb *send*, however, has two complements in (21) and only one in (30). That is because the lexical entry in (23) above, which licenses both verbal word structures, specifies that its second (PP) complement is optional. The noun *letters* in the two examples is licensed by the same lexical entry (27), which takes an optional PP complement.

In (21), there was no node spanning the string *two letters to Lee.* In (30), however, there is such a node. A more detailed subtree for that NP is the following:

(31)



The top node in this subtree is licensed by the Head-Specifier Rule which requires the identity of the determiner with the one element on the head's SPR list. The second daughter, dominating *letters to Lee* is licensed by the Head-Complement Rule, which also requires that the element on the COMPS list of the head noun is identical to the PP complement. The other identities are enforced by various principles in ways that should now be familiar.

Notice that the tag on the RESTR of *to Lee* in (30) and (31) is different from the tag in (21). That is because the role played by Lee is subtly different in the two sentences. In (30), the SENDEE role does not correspond to any syntactic constituent; in (21), the PP *to Lee* (and the noun *Lee*, with which it is coindexed) plays the SENDEE role. On the other hand, in (30), the PP plays the ADDRESSEE role with respect to the noun *letters* – a role that is syntactically unrealized in (21). While most letters are sent to their addressees, it is possible for the sendee and the addressee to be different, as in *I sometimes inadvertently send letters to my sister to my brother.* We have annotated this difference by giving *Lee* the two minimally different RESTR values in (32):[10]

(32) a. $\boxed{H}\left\langle\begin{bmatrix}\text{RELN} & \textbf{name}\\ \text{NAME} & \text{Lee}\\ \text{NAMED} & j\end{bmatrix}\right\rangle$

b. $\boxed{I}\left\langle\begin{bmatrix}\text{RELN} & \textbf{name}\\ \text{NAME} & \text{Lee}\\ \text{NAMED} & m\end{bmatrix}\right\rangle$

Since j is the index for the SENDEE role in all of our trees in this section, $\boxed{H}$ is used when *Lee* is the SENDEE argument of the verb *send.* We use m as the index for the ADDRESSEE role, so we use $\boxed{I}$ when *Lee* plays the ADDRESSEE role with respect to the noun *letters.* [11]

[10]For readers who are still skeptical of the existence of this second structure (and interpretation), we provide an alternative appropriate embedding context:

The Corrupt Postal Worker Ransom Context:

[Postal workers A, B and C have stolen some important letters. C, who is negotiating ransom money for release of the letters addressed to Lee, is going over the plan with A and B:]

C: So if the phone rings twice, what do you send us?

B: We send two letters to Lee.

[11]This difference could have been annotated in another way. We could have used the same RESTR value for *to Lee* in both cases and assigned alphabetically different values to the SENDEE and ADDRESSEE roles in the two sentences. These two alternatives are not substantively different. They only appear to be distinct because of the way we use tag identity across different sentences in this section.

6.3 Appendix: Well-Formed Structures

In this appendix, we lay out more precisely the constructs of the theory whose effects we have been illustrating in this chapter. This presentation (like the elaborations of it given in Chapter 9 and Appendix A) is intended for readers concerned with the formal foundations of our theory. For most purposes and for most readers, the relatively informal presentation in the body of text, taken together with the definitions in section 6.3.6 below, should be sufficient.

6.3.1 Preliminaries

According to our approach, a grammar G is defined by the following components:

- a finite set of features: $\mathcal{F} = \{\text{SYN, SEM, HEAD, AGR}, \ldots\}$,
- a finite set of primitive items:
  $\mathcal{A}_{atom} = \mathcal{A}_{pol} \cup \mathcal{A}_{gr.atom} \cup \mathcal{A}_{mode} \cup \mathcal{A}_{reln}$, where:
  1. $\mathcal{A}_{pol} = \{+, -\}$,
  2. (a set of ground atoms) $\mathcal{A}_{gr.atom} = \{\text{1st, 2nd, 3rd, sg, pl}, \ldots, \text{run, dog}, \ldots\}$,
  3. $\mathcal{A}_{mode} = \{\text{prop, ques, dir, ref, none}\}$, and
  4. $\mathcal{A}_{reln} = \{\textbf{walk}, \textbf{love}, \textbf{person}, \ldots\}$,
- a denumerably infinite set of primitive items: $\mathcal{A}_{index} = \mathcal{A}_{ind} \cup \mathcal{A}_{sit}$, where:
  1. $\mathcal{A}_{ind} = \{i, j, \ldots\}$ and
  2. $\mathcal{A}_{sit} = \{s_1, s_2, \ldots\}$,
- the distinguished element *elist* (*empty-list*), discussed below,
- a finite set of types: $\mathcal{T} = \{\textit{noun, agr-pos, plural, expression}, \ldots\}$,
- a type hierarchy with a tree structure associated with constraint inheritance (for instance, the type hierarchy represented by the tree and table in Section 5.10.1 and 5.10.2),
- a set $\mathcal{LT} \subset \mathcal{T}$ called the *leaf types* (a type τ is a *leaf type* if it is associated with a leaf in the type hierarchy tree, i.e. if τ is one of the most specific types),
- a set of list types (if τ is a type, then *list*(τ) is a type),
- a set of grammar rules (like the ones we have already encountered, see Section 5.10.4),
- a set of principles (like those in Section 5.10.5), and
- a lexicon (which is a finite set of lexical entries like those in Section 5.10.6).

Thus a grammar G comes with various primitives grouped into two sets: $\mathcal{A}_{atom}$ ($\mathcal{A}_{pol}$, $\mathcal{A}_{gr.atom}$, $\mathcal{A}_{mode}$, $\mathcal{A}_{reln}$) and $\mathcal{A}_{index}$ ($\mathcal{A}_{ind}$, and $\mathcal{A}_{sit}$). G assigns the type *atom* to all elements of $\mathcal{A}_{atom}$. The elements of $\mathcal{A}_{index}$ are used by the grammar for describing individual objects and situations; they are associated with the leaf type *index*. We assume that no items in these sets of primitives can be further analyzed via grammatical features.

Our grammar appeals to several ancillary notions which we now explicate: *feature structure description*, *feature structure*, *satisfaction of a description*, and *tree structure*.

6.3.2 Feature Structure Descriptions

For expressing the constraints associated with the grammar rules, principles, types, and lexical entries, we introduce the notion of a feature structure description. The feature structure descriptions are given as attribute-value matrices, augmented with the connective '|', set descriptors ($\{\ldots\}$), list descriptions ($\langle\ldots\rangle$, attribute-value matrices with FIRST/REST, or two list descriptions connected by $\oplus$), and a set *Tags* of tags (labels represented by boxed integers or letters).

6.3.3 Feature Structures

The set of feature structures $\mathcal{FS}$ is given by the following recursive definition:

(33) $\phi \in \mathcal{FS}$ (i.e. ϕ is a feature structure) iff

a. $\phi \in \mathcal{A}_{atom} \cup \mathcal{A}_{index}$, or

b. ϕ is a function from features to feature structures, $\phi : \mathcal{F} \longrightarrow \mathcal{FS}$ satisfying the following conditions

1. ϕ is of a leaf type τ;
2. $DOM(\phi) = \{F \mid \text{G declares } F \text{ appropriate for } \tau\} \cup$

$$\{F' \mid \exists\tau' \text{ such that } \tau' \text{ is a supertype of } \tau \text{ and}$$
$$\text{G declares } F' \text{ appropriate for } \tau'\},$$

i.e. ϕ is defined for any feature that is declared appropriate for τ or for any of τ's supertypes;

3. for each $F \in DOM(\phi)$, G defines the type of the value $\phi(F)$ (we call the value $\phi(F)$ of the function ϕ on F the value of the feature F); and
4. ϕ obeys all further constraints ('type constraints') that G associates with type τ (including those inherited from the supertypes τ' of τ), or

c. ϕ is of type *list*(τ), for some type τ, in which case either:

1. ϕ is the distinguished element *elist*, or else:
2. A. $DOM(\phi)$ is $\{\text{FIRST, REST}\}$,
   B. the type of $\phi(\text{FIRST})$ is τ, and
   C. the type of $\phi(\text{REST})$ is *list*(τ).

6.3.4 Satisfaction

We explain how feature structures satisfy descriptions indirectly – in terms of denotation, which we define as follows:

Denotation of Feature Structure Descriptions

The denotation of a feature structure description is specified in terms of a structure $\mathcal{M}$:

(34) $\mathcal{M} = \langle \mathcal{A}, \mathcal{F}, \mathcal{T}, \textbf{Type}, I \rangle$, where:

1. $\mathcal{A} = \mathcal{A}_{atom} \cup \mathcal{A}_{index} \cup \{\textit{elist}\}$,
2. $\mathcal{F}$ is a finite set of features,
3. $\mathcal{T}$ is a finite set of types,
4. **Type** is a function mapping feature structures to types –
   Type : $\mathcal{FS} \longrightarrow \mathcal{LT}$, where $\mathcal{LT}$ is the set of the leaf types, and

5. I is a function mapping feature names and atomic descriptors to features and atoms of the appropriate sort:

$$I \in I_{\widetilde{\mathcal{F}}} \cup I_{\widetilde{\mathcal{A}}_{atom}} \cup I_{\widetilde{\mathcal{A}}_{ind}} \cup I_{\widetilde{\mathcal{A}}_{sit}} \cup \{\langle \mathit{elist}, \mathit{elist}\rangle\},$$

where
$I_{\widetilde{\mathcal{F}}} \in \mathcal{F}^{\widetilde{\mathcal{F}}}$, $I_{\widetilde{\mathcal{A}}_{atom}} \in \mathcal{A}_{atom}^{\widetilde{\mathcal{A}}_{atom}}$, $I_{\widetilde{\mathcal{A}}_{ind}} \in \mathcal{A}_{ind}^{\widetilde{\mathcal{A}}_{ind}}$, $I_{\widetilde{\mathcal{A}}_{sit}} \in \mathcal{A}_{sit}^{\widetilde{\mathcal{A}}_{sit}}$,

and $\widetilde{X}$ denotes the set of expressions that have denotations in the set X.[12]

The function I is called an *interpretation function*. An *assignment function* is a function

$$g : \mathit{Tags} \longrightarrow \mathcal{FS}.$$

We say that a feature structure ϕ is of type $\tau \in \mathcal{T}$ iff there is a (unique) leaf type $\tau' \in \mathcal{LT}$ such that:

(35) 1. τ' is a subtype of τ, and
2. $\mathbf{Type}(\phi) = \tau'$.

Given $\mathcal{M}$, the interpretation $[\![d]\!]^{\mathcal{M},g}$ of a feature structure description d with respect to an assignment function g is defined recursively as follows:

(36) 1. if $v \in \widetilde{\mathcal{F}} \cup \widetilde{\mathcal{A}}_{atom} \cup \widetilde{\mathcal{A}}_{index}$, then $[\![v]\!]^{\mathcal{M},g} = \{I(v)\}$;
2. if τ is a type, i.e. $\tau \in \mathcal{T}$, then $[\![\tau]\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : \phi \text{ is of type } \tau\}$;
3. if $F \in \widetilde{\mathcal{F}}$, and d is a feature structure description, then $[\![[F\ d]]\!]^{\mathcal{M},g} =$ $\{\phi \in \mathcal{FS} : \text{there is some } \phi' \text{ such that } \phi' \in [\![d]\!]^{\mathcal{M},g} \text{ and } \langle I(F), \phi'\rangle \in \phi\}$;[13]
4. $$\text{if } d = \begin{bmatrix} d_1 \\ \ldots \\ d_n \end{bmatrix}$$
where $n \geq 1$, and $d_1, \ldots, d_n$ are feature structure descriptions, then
$$[\![d]\!]^{\mathcal{M},g} = \bigcap_{i=1}^{n} [\![d_i]\!]^{\mathcal{M},g};$$
5. if d is a set descriptor $\{d_1, \ldots, d_n\}$, then
$$[\![d]\!]^{\mathcal{M},g} = \bigcup_{i=1}^{n} [\![d_i]\!]^{\mathcal{M},g}$$
$([\![\{\ \}]\!]^{\mathcal{M},g} = \emptyset)$;
6. $[\![d_1 \mid d_2]\!]^{\mathcal{M},g} = [\![d_1]\!]^{\mathcal{M},g} \cup [\![d_2]\!]^{\mathcal{M},g}$;
7. if $d \in \mathit{Tags}$, then $[\![d]\!]^{\mathcal{M},g} = g(d)$;
8. if $d \in \mathit{Tags}$ and d' is a feature structure description, then $[\![d\ d']\!]^{\mathcal{M},g} = \{\phi \in \mathcal{FS} : g(d) = \phi \text{ and } \phi \in [\![d']\!]^{\mathcal{M},g}\}$;
(Note that tagging narrows the interpretation down to a singleton set.)

[12] Y^X is the standard notation for the set of all functions $f : X \to Y$.

[13] Note that the definition of a feature structure in (33), taken together with this clause, ensures that each element ϕ of the set $[\![[F\ d]]\!]^{\mathcal{M},g}$ is a proper feature structure.

9. List Addition:[14]
a. $[\![\mathit{elist} \oplus d]\!]^{\mathcal{M},g} = [\![d]\!]^{\mathcal{M},g}$,
b. $\text{if } d = \begin{bmatrix} \text{FIRST} & d_1 \\ \text{REST} & d_2 \end{bmatrix} \oplus d_3,$
then $[\![d]\!]^{\mathcal{M},g} =$
$\{\phi \in \mathcal{FS} : \phi(\text{FIRST}) \in [\![d_1]\!]^{\mathcal{M},g} \text{ and } \phi(\text{REST}) \in [\![d_2 \oplus d_3]\!]^{\mathcal{M},g}\}$.

Satisfaction of Feature Structure Descriptions

A feature structure $\phi \in \mathcal{FS}$ satisfies a feature structure description d iff there is some assignment function g such that $\phi \in [\![d]\!]^{\mathcal{M},g}$.

Examples:

(37) a. ϕ satisfies [NUM sg] iff $\langle \text{NUM}, \text{sg}\rangle \in \phi$.
b. ϕ satisfies [AGR [NUM sg]] iff there is a feature structure ϕ' (which is unique) such that $\langle \text{AGR}, \phi'\rangle \in \phi$ and $\langle \text{NUM}, \text{sg}\rangle \in \phi'$.
c. ϕ satisfies [AGR *3sing*] iff there is a feature structure ϕ' (which is unique) such that $\langle \text{AGR}, \phi'\rangle \in \phi$ and ϕ' is of type *3sing*.
d. ϕ satisfies [PER {1st, 2nd, 3rd}] iff $\langle \text{PER}, \text{1st}\rangle \in \phi$, $\langle \text{PER}, \text{2nd}\rangle \in \phi$, or $\langle \text{PER}, \text{3rd}\rangle \in \phi$.
e. ϕ satisfies [ARGS $\langle s_1, s_2, s_3\rangle$] iff: $\langle \text{ARGS}, \{\langle \text{FIRST}, s_1\rangle, \langle \text{REST}, \{\langle \text{FIRST}, s_2\rangle, \langle \text{REST}, \{\langle \text{FIRST}, s_3\rangle, \langle \text{REST}, \mathit{elist}\rangle\}\rangle\}\rangle\}\rangle \in \phi$.
f. ϕ satisfies:
$$\left[\text{SYN} \begin{bmatrix} \text{HEAD} & \left[\text{AGR}\ \boxed{1}\right] \\ \text{VAL} & \left[\text{SPR}\ \langle\ [\text{SYN}\ [\ \text{HEAD}\ [\text{AGR}\ \boxed{1}]]]\ \rangle\right] \end{bmatrix}\right]$$
iff
1. $\phi(\text{SYN})(\text{HEAD})(\text{AGR}) =$ $\phi(\text{SYN})(\text{VAL})(\text{SPR})(\text{FIRST})(\text{SYN})(\text{HEAD})(\text{AGR})$,[15] and
2. $\phi(\text{SYN})(\text{VAL})(\text{SPR})(\text{REST}) = \mathit{elist}$

6.3.5 Tree Structures

Finally, we assume a notion of tree structure described informally as follows:

(38) A tree structure is a directed graph that satisfies a number of conditions:[16]

[14] Where no confusion should arise, we use 'FIRST', 'SYN', etc. to refer either to the appropriate feature (an element of $\mathcal{F}$) or to its name (an element of $\widetilde{\mathcal{F}}$).

[15] Note that parentheses here are 'left associative': '$\phi(X)(Y)$' is equivalent to '$(\phi(X))(Y)$'. That is, both expressions denote the result of applying the function ϕ to (the feature) X and then applying the result to (the feature) Y.

[16] Here, we assume familiarity with notions such as *root*, *mother*, *terminal node*, *nonterminal node*, and *branches*. These and related notions can be defined more precisely in set-theoretic terms, as is done in various texts. See, for example, Hopcroft et al. 2001 and Partee et al. 1990.

1. it has a unique root node,
2. each non-root node has exactly one mother,
3. sister nodes are ordered with respect to each other,
4. it has no crossing branches,
5. each nonterminal node is labelled by a feature structure, and
6. each terminal node is labeled by a phonological form (an atom).

6.3.6 Structures Defined by the Grammar

We may now proceed to define well-formedness of tree structures in terms of the licensing of their component trees (recall from Chapters 2 and 3 that a local subtree consists of a mother and all its daughters):

(39) **Well-Formed Tree Structure:**

Φ is a Well-Formed Tree Structure according to G if and only if:

1. Φ is a tree structure,
2. the label of Φ's root node satisfies S,[17] and
3. each local subtree within Φ is either phrasally licensed or lexically licensed.

(40) **Lexical Licensing:**
A word structure of the form:

$$\begin{array}{c}\phi\\ |\\ \omega\end{array}$$

is licensed if and only if G contains a lexical entry $\langle d_1, d_2\rangle$, where ω satisfies d_1 and ϕ satisfies d_2.

(41) **Phrasal Licensing:**
A grammar rule $\rho = d_0 \rightarrow\ d_1 \ldots\ d_n$ licenses a local subtree:

$$\Phi = \begin{array}{c}\phi_0\\ \diagup\quad\diagdown\\ \phi_1\ \ \ldots\ \ \phi_n\end{array}$$

if and only if:

1. for each $i, 0 \leq i \leq n$, ϕ_i is of[18] the type *expression*,
2. there is some assignment function g under which the sequence $\langle\phi_0, \phi_1, ..., \phi_n\rangle$ satisfies the description sequence $\langle d_0, d_1, \ldots, d_n\rangle$,[19]
3. Φ satisfies the Semantic Compositionality Principle, and
4. if ρ is a headed rule, then Φ satisfies the Head Feature Principle, the Valence Principle and the Semantic Inheritance Principle, with respect to ρ.

[17]Recall once again that S abbreviates a certain feature structure constraint, as discussed in Chapter 4.
[18]That is, assigned to some leaf type that is a subtype of the type *expression*.
[19]Note that this clause must speak of a sequence of feature structures satisfying a sequence description. This is because of identities that must hold across members of the sequence, e.g. those required by particular grammar rules.

(42) Φ satisfies the Semantic Compositionality Principle with respect to a grammar rule ρ if and only if Φ satisfies:

$$\begin{array}{c}\left[\text{RESTR}\ \ \boxed{A_1} \oplus \ldots \oplus \boxed{A_n}\right]\\ \diagup\qquad\diagdown\\ \left[\text{RESTR}\ \boxed{A_1}\right]\ \ \ldots\ \ \left[\text{RESTR}\ \boxed{A_n}\right]\end{array}$$

(43) Φ satisfies the Head Feature Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{array}{c}\left[\text{HEAD}\ \boxed{1}\right]\\ \diagup\quad|\quad\diagdown\\ \ldots\quad \begin{array}{c}\phi_h\\ \left[\text{HEAD}\ \boxed{1}\right]\end{array}\quad \ldots\end{array}$$

where ϕ_h is the head daughter of Φ.

(44) Φ satisfies the Valence Principle with respect to a headed rule ρ if and only if, for any VAL feature F, Φ satisfies:

$$\begin{array}{c}\left[\text{F}\ \boxed{A}\right]\\ \diagup\quad|\quad\diagdown\\ \ldots\quad \begin{array}{c}\phi_h\\ \left[\text{F}\ \boxed{A}\right]\end{array}\quad \ldots\end{array}$$

where ϕ_h is the head daughter of Φ and ρ does not specify incompatible F values for ϕ_h and ϕ_0.

(45) Φ satisfies the Semantic Inheritance Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{array}{c}\begin{bmatrix}\text{MODE} & \boxed{4}\\ \text{INDEX} & \boxed{5}\end{bmatrix}\\ \diagup\quad|\quad\diagdown\\ \ldots\quad \begin{array}{c}\phi_h\\ \begin{bmatrix}\text{MODE} & \boxed{4}\\ \text{INDEX} & \boxed{5}\end{bmatrix}\end{array}\quad \ldots\end{array}$$

where ϕ_h is the head daughter of Φ.

6.4 Problems

⚠ Problem 1: A Sentence

For the purposes of this problem, assume that the preposition *on* in the example below is like *to* in (18) in that makes no contribution to the semantics other than to pass up the INDEX and MODE values of its object NP. That is, assume it has the following lexical entry:

$$\left\langle \text{on}\,, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\;\rangle \\ \text{COMPS} & \left\langle \begin{array}{c}\text{NP}\\ \begin{bmatrix}\text{CASE} & \text{acc}\\ \text{MODE} & \boxed{1} \\ \text{INDEX} & \boxed{2}\end{bmatrix}\end{array}\right\rangle \\ \text{MOD} & \langle\;\rangle \end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \boxed{1}\\ \text{INDEX} & \boxed{2} \\ \text{RESTR} & \langle\;\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

A. Draw a fully resolved tree structure for the sentence in (i). Use tags to indicate identities required by the grammar. When two feature structures are tagged as identical, you need only show the information in one place.

   (i) I rely on Kim.

B. In the VP and PP nodes of your tree, indicate which aspects of the grammar constrain each piece of information (i.e. each feature value). [*Hint: Possible answers include grammar rules and the combined effect of general principles and lexical entries.*]

Problem 2: Spanish NPs II

In this problem we return to Spanish NPs (see Problem 2 in Chapter 4), this time adding adjectives. Unlike English adjectives, Spanish adjectives agree with the nouns they modify, as shown in (i)–(iv):

(i) a. La jirafa pequeña corrió.
The.FEM.SG giraffe small.FEM.SG ran.3SG
'The small giraffe ran.'
b.*La jirafa pequeñas/pequeño/pequeños corrió.

(ii) a. Las jirafas pequeñas corrieron.
The.FEM.PL giraffes small.FEM.PL ran.3PL
'The small giraffes ran.'
b.*Las jirafas pequeña/pequeño/pequeños corrieron.

(iii) a. El pingüino pequeño corrió.
The.MASC.SG penguin small.MASC.SG ran.3SG
'The small penguin ran.'
b.*El pingüino pequeña/pequeñas/pequeños corrió.

(iv) a. Los pingüinos pequeños corrieron.
The.MASC.PL penguins small.MASC.PL ran.3PL
'The small penguins ran.'
b.*Los pingüinos pequeña/pequeñas/pequeño corrieron.

A. Using the MOD feature to specify which nouns the adjective can modify, give a lexical entry for *pequeños*. Be sure to specify both SYN and SEM features.

[*Hint: The semantics of adjectives is very similar to that of adverbs, so the entry for today in Chapter 5 (page 147) may be a helpful guide in doing this.*]

B. Assuming the rules we have developed for English are appropriate for Spanish as well, draw a tree for the NP *los pingüinos pequeños* in (iv). Show values for all features, using tags to show identities required by the grammar.

C. Explain how the INDEX value of *pingüinos* is identified with the argument of the *predication* introduced by *pequeños*. (Your explanation should indicate the role of lexical entries, rules, and principles in enforcing this identity.)

⚠ Problem 3: English Possessives I

English uses *'s* to express possession, as in the following examples:

(i) Leslie's coffee spilled.
(ii) Jesse met the president of the university's cousin.
(iii)*Jesse met the president's of the university cousin.
(iv) Don't touch that plant growing by the trail's leaves.
(v)*Don't touch that plant's growing by the trail leaves.
(vi) The person you were talking to's pants are torn.
(vii)*The person's you were talking to pants are torn.

(While examples (iv) and (vi) are a bit awkward, people do use such sentences, and there is certainly nowhere else that the *'s* could be placed to improve them).

A. What is the generalization about where the *'s* of possession appears in English?

One traditional treatment of the possessive marker (*'s*) is to claim it is a case marker. In our terms this means that it indicates a particular value for the feature CASE (say, 'poss' for 'possessive') on the word it attaches to. If we tried to formalize this traditional treatment of *'s*, we might posit a rule along the following lines, based on the fact that possessive NPs appear in the same position as determiners:

$$\text{D} \rightarrow \begin{array}{c}\text{NP}\\ [\text{CASE}\;\;\text{poss}]\end{array}$$

Taken together with our assumption that CASE is a HEAD feature, such an analysis of *'s* makes predictions about the grammaticality of (i)–(vii).

B. Which of these sentences does it predict should be grammatical, and why?

Problem 4: English Possessives II

An alternative analysis of the possessive is to say that *'s* is a determiner that builds a determiner phrase (abbreviated DP), via the Head-Specifier Rule. On this analysis, *'s* selects for no complements, but it obligatorily takes an NP specifier. The word *'s* thus has a lexical category that is like an intransitive verb in valence.

This analysis is somewhat unintuitive, for two reasons: first, it requires that we have an independent lexical entry for *'s*, which seems more like a piece of a word, phonologically; and second, it makes the nonword *'s* the head of its phrase! However, this analysis does a surprisingly good job of predicting the facts of English possessives, so we shall adopt it, at least for purposes of this text.

A. Ignoring semantics for the moment, give the lexical entry for *'s* assuming its analysis as a determiner, and draw a tree for the NP *Kim's brother.* (The tree should show the value of HEAD, SPR and COMPS on every node. Use tags to show identities required by the grammar. You may omit other features.)

B. Explain how your lexical entry gets the facts right in the following examples:

   (i) The Queen of England's crown disappeared.

   (ii)*The Queen's of England crown disappeared.

C. How does this analysis handle recursion in possessives, for example, *Robin's brother's wife*, or *Robin's brother's wife's parents*? Provide at least one tree fragment to illustrate your explanation. (You may use abbreviations for node labels in the tree.)

Problem 5: English Possessives III

The semantics we want to end up with for *Pat's book* is the one shown in (i) (*poss* is the name of the general possession relation that we will assume provides the right semantics for all possessive constructions):[20]

$$\text{(i)}\ \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAMED} & j \\ \text{NAME} & \text{Pat} \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{poss} \\ \text{POSSESSOR} & j \\ \text{POSSESSED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{the} \\ \text{BV} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{book} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix}$$

[20]We have chosen to use 'the' as the quantifier introduced by possessives, but this is in fact a matter of debate. On the one hand, possessive NPs are more definite than standard indefinites such as *a book.* On the other hand, they don't come with the presupposition of uniqueness that tends to come with *the.* Compare (i) and (ii):

(i) That's the book.

(ii) That's my book.

Part (A) of this problem will ask you to give a SEM value for the determiner *'s* that will allow the grammar to build the SEM value in (i) for the phrase *Pat's book.* Recall that, on our analysis, nouns like *book* select for specifiers like *Pat's*, and the specifiers do not reciprocally select for the nouns. In order to get the correct semantics, *'s* will have to identify its BV value with its INDEX value. In this, it is just like the determiner *a* (see (9) on page 171). This constraint interacts with the constraint on all common nouns shown in (ii) to ensure that the value of BV is correctly resolved:

$$\text{(ii)}\ \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle [\text{SEM [INDEX } \boxed{1}]] \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & [\text{INDEX } \boxed{1}] \end{bmatrix}$$

A. Given the discussion above, what is the SEM value of the determiner *'s*?

B. Draw a tree for the phrase *Pat's book*, showing all SEM features on all nodes and SPR on any nodes where it is non-empty. Use tags (or matching indices, as appropriate) to indicate identities required by the grammar.

C. Describe how your analysis guarantees the right SEM value for the phrase. (Your description should make reference to lexical entries, rules and principles, as appropriate.)

Problem 6: English Possessive Pronouns

Possessive pronouns like *my*, *your*, etc. function as determiners in NPs like *my books* and *your mother.* You might think we should treat possessive pronouns as determiners that have the same AGR value as the corresponding nonpossessive pronoun. That is, you might think that *my* should be specified as:

$$\text{(i)}\ \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{AGR} & \begin{bmatrix} \textit{1sing} & \\ \text{PER} & \text{1st} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

A. Explain why this analysis (in particular, the AGR value shown in (i)) will fail to provide an adequate account of *my books* and *your cousin.*

B. The semantics we want to end up with for *my book* is this:

$$\text{(ii)}\ \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{speaker} \\ \text{INST} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{poss} \\ \text{POSSESSOR} & j \\ \text{POSSESSED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{the} \\ \text{BV} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{book} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix}$$

Formulate the SEM value of the determiner *my.*

C. Draw an explicit tree for the phrase *my book.*

[Hint: Refer to Problem 5.]

Problem 7: French Possessive Pronouns

Problem 6 asked you to provide an argument as to why *my* isn't [PER 1st, NUM sg], but didn't concern what the AGR value should be instead.

A. Provide an argument, with suitable data, that the AGR value of English possessive pronouns (e.g. *my* or *our*) should be left unspecified for number.

Now consider the following data from French. French nouns, like Spanish nouns, are all assigned either masculine or feminine gender. In these examples, *pie* is feminine and *moineau* is masculine.

(i) ma pie
my magpie
(ii)*mon/mes pie
(iii) mon moineau
my sparrow
(iv)*ma/mes moineau
(v) mes pies
my magpies
(vi)*ma/mon pies
(vii) mes moineaux
my sparrows
(viii)*ma/mon moineaux

B. Give the AGR values for *ma*, *mon*, and *mes*.

7

Binding Theory

7.1 Introduction

This chapter revisits a topic introduced very informally in Chapter 1, namely, the distribution of reflexive and nonreflexive pronouns. In that discussion, we noticed that the well-formedness of sentences containing reflexives usually depends crucially on whether there is another expression in the sentence that has the same referent as the reflexive; we called such an expression the 'antecedent' of the reflexive. Nonreflexive pronouns, on the other hand, often lack an antecedent in the same sentence. The issue for a nonreflexive pronoun is typically whether a particular NP could have the same referent (or, as linguists often put it, be coreferential with it) – that is, whether that NP could serve as the antecedent for that pronoun.

In discussing these phenomena, we will use the notation of subscripted indices to mark which expressions are intended to have the same referent and which are intended to have distinct referents. Two expressions with the same index are to be taken as coreferential, whereas two expressions with different indices are to be understood as having distinct referents.

Thus the markings in (1) indicate that *himself* must refer to the same person as *John*, and that the referent of *her* must be someone other than Susan:

(1) a. John$_i$ frightens himself$_i$.
b.*Susan$_i$ frightens her$_i$.
c. Susan$_i$ frightens her$_j$.

As mentioned in Chapter 5, the subscript notation is shorthand for the value of the feature INDEX.

In examples like (1a), the reflexive *himself* is often said to be 'bound' by its antecedent. This terminology derives from an analogy between natural language pronouns and variables in mathematical logic. The principles governing the possible pairings of pronouns and antecedents are often called BINDING PRINCIPLES, and this area of study is commonly referred to as BINDING THEORY.[1] The term ANAPHORIC is also used for

[1]Much of the literature on Binding Theory actually restricts the term 'binding' to elements in certain syntactic configurations. Specifically, an element *A* is often said to bind an element *B* if and only if: (i) they have the same index; and (ii) *A* c-commands *B*. The technical term 'c-command' has been defined in several (nonequivalent) ways in the literature; the most commonly used definition is the following: