5

Semantics

5.1 Introduction

Our first example of syntactic argumentation in Chapter 1 was the distribution of reflexive and nonreflexive pronouns. In Chapter 7 we will return to this topic and show how it can be analyzed in the grammar we are developing. Before we can do so, however, we need to consider the nature of reference and coreference – topics that are fundamentally semantic in nature (i.e. that have to do in large part with meaning). And before we can do that, we need to discuss meaning more generally, sketching how to represent meaning in our grammar.

Reflexive pronouns provide perhaps the clearest case in which a semantic factor – coreference, in this case – plays an essential role in the grammatical distribution of particular words. But there are many other syntactic phenomena that are closely linked to meaning. Consider, for example, subject-verb agreement, which we have discussed extensively in the past two chapters. The NUM value of a noun is often predictable from its referent. Singular nouns generally refer to individual objects, and plural nouns normally refer to collections of objects. Mass nouns (which are mostly singular) usually refer to substances – that is, entities that are not naturally packaged into discrete objects. Of course, nature doesn't fully determine how the world should be divided up conceptually into objects, collections, and substances, so there may be differences between languages, or even between individuals, as to how things are referred to. Hence the German word *Hose* means essentially the same thing as English *pants* or *trousers*, but the German is singular while the English is plural. Likewise, the French use the plural noun *cheveux* to refer to the same stuff that we call *hair*. And individual English speakers differ as to whether they can use *lettuce* as a count noun. Although the correspondences are usually imperfect, syntactic properties (including such basic ones as the part-of-speech distinctions) are often closely linked to semantic characteristics. Trying to do syntax without acknowledging the associated semantic regularities would lead to missing many fundamental generalizations about linguistic structure.

The study of meaning is at least as old as the study of grammar, and there is little hope of doing justice to problems of semantics in a textbook whose primary concern is grammatical structure. However, if the grammars we develop are going to play any role in modeling real language use, then grammar minimally has to include some information about the meaning of individual words and a treatment of how these combine with each

other – that is, an account of how meanings of phrases and sentences are built up from the meanings of their parts. Let us begin by contemplating the nature of sentence meaning.

5.2 Semantics and Pragmatics

Meaning is inextricably bound up with actions – people use language intentionally to do many kinds of things. Some sentences are conventionally used to query; others to make simple assertions; still others are conventionally used to issue commands. Even a piece of a sentence, say an NP like *the student sitting behind Leslie*, can be used in isolation to perform the communicative act of referring to an individual.

The kind of meaning that a sentence can be used to convey depends crucially on its syntactic form. For example, a simple 'inverted' sentence like (1), with an auxiliary verb before the subject NP, is typically used to make a query:

(1) Is Sandy tall?

And the query posed by uttering (1) is closely related to the assertion made by an utterance of the noninverted sentence in (2):

(2) Sandy is tall.

In fact, uttering (2) is a perfectly good way of answering (1).

These observations about communication, or language use, have led researchers to the view that the conventional meanings of different kinds of sentences are different kinds of abstract objects. A declarative sentence like (2), for example, is usually associated with something called a PROPOSITION. A proposition is the kind of thing you can assert, deny, or believe. It is also something (the only kind of thing) that can be true or false. An interrogative sentence like (1) is associated with a semantic object called a QUESTION. Questions are the kind of thing that can be asked and answered. Similarly, we'll call the semantic object associated with an imperative sentence a DIRECTIVE. This is the kind of object that can be issued (by simply uttering an imperative sentence, for example), and fulfilled (by causing the conditions associated with the sentence to be met). Semantics is the study of abstract constructs like propositions, questions and directives, which are assumed to play a key role in a larger theory of communication.[1]

Semantic analysis provides just one part of the account of what people convey when they communicate using language, though. In this text, we make the standard assumption that communication has two components: linguistic meaning (as characterized by semantic analysis) and reasoning about communicative goals. When a linguistic expression is uttered, its linguistic meaning makes a significant contribution to, but does not fully determine, the communicative function of the utterance.

Consider, for example, an utterance of (3):

(3) Do you have a quarter?

As noted above, we take the linguistic meaning of this sentence to be a particular question. Once the identity of the hearer is determined in the relevant context of utterance, a

[1] When speaking informally, we will sometimes talk of a given sentence as conveying a given message (proposition, question, or directive). What we really mean is that our semantic analysis associates a particular message with a given sentence and that the communicative potential of that sentence (what it can be used to convey) is determined in large part by that message.

question of this form has a determinate answer: yes or no. However, an utterance of (3) might serve to communicate much more than such a simple factual inquiry. In particular, in addition to posing a financial query to a given hearer, an utterance of (3) is likely to convey a further message – that the speaker was making the following request of the hearer:

(4) Please give me a quarter!

The question asked by an utterance of (3) is generally referred to as its LITERAL or CONVENTIONAL meaning. A request like (4) is communicated by inference. Asking a certain question (the literal meaning of the interrogative sentence in (3)) in a certain kind of context can lead a hearer to reason that the deeper communicative goal of the speaker was to make a particular request, i.e. the one conveyed by (4). In a different context, i.e. a parent asking (3) of a child standing in a line of children waiting to pay a twenty-five cent admission fee for an amusement park ride, would not lead the hearer to infer (4), but rather to check to make sure that (s)he had the required admission fee. We will leave the account of such embellished communication (even the routine 'reading between the lines' that occurs more or less effortlessly in cases like this) to a more fully developed theory of language use, that is, to a theory of linguistic PRAGMATICS. The inference from query to request is pragmatic in nature.

By contrast, the fact that a sentence like (3) must express a question as its literal meaning is semantic in nature. SEMANTICS is the study of linguistic meaning, that is, the contribution to communication that derives directly from the conventions of the language. Pragmatics is a more general study, of how linguistic meaning interacts with situational factors and the plans and goals of conversational participants to achieve more subtle, often elaborate communicative effects.

The semantic analysis that a grammar provides serves as input for a theory of pragmatics or language use. Such a theory sets as its goal to explain what actually gets communicated via pragmatic inferences derived from the linguistic meaning of an utterance. For example, pragmatic theory might include a principle like (5):[2]

(5) Quantity Principle (simplified)

If X is weaker than Y, then asserting X implies the denial of Y.

This principle leads to pragmatic inference via 'proofs' of the following kind (justifications for steps of the proof are given in parentheses):

(6)
- A says to B: *Two things bother Pat.*
- A uttered something whose linguistic meaning is:
  'At least two things bother Pat'. (semantic analysis)[3]

[2]The principle in (5), due to Grice (1989), relies on the undefined term 'weaker'. In some cases (such as the example that follows), it is intuitively obvious what 'weaker' means. But a full-fledged pragmatic theory that included (5) would have to provide a precise definition of this term.

[3]Note that the meaning of the word *two* is no stronger than the 'at least two' meaning, otherwise the following would be contradictory:

(i) [Kim: Do you have two dollars?]
Sandy: Yes, I have two dollars. In fact, I have five dollars.

- 'At least two things bother Pat'. is weaker than 'At least three things bother Pat'. (This is true in the context; possibly true more generally)
- B assumes that A also meant to communicate: 'It's not the case that there are three things that bother Pat'. (Quantity Principle)

Note that exactly the same pragmatic inference would arise from an utterance by A of any semantically equivalent sentence, such as *There are two things that bother Pat* or *Pat is bothered by two things.* This is because pragmatic theory works from the linguistic meaning of an utterance (as characterized by our semantic analysis) and hence is indifferent to the form by which such meanings are expressed.[4]

There is much more that could be said about the fascinating topic of pragmatic inference. Here, our only goal has been to show that the semantic analysis that must be included in any adequate grammar plays an essential role, albeit an indirect one, in explaining the communicative function of language in context.[5]

5.3 Linguistic Meaning

5.3.1 Compositionality

In order to even begin to deal with semantic issues like

- Which proposition is conveyed by a given declarative sentence?
- Which question is conveyed by a given interrogative sentence?

we first have to clarify what smaller semantic units propositions and questions are constructed from. Moreover, we will need to formulate constraints that specify how the meaning of a given sentence is determined by the meanings of its parts and the way that they are combined.

When we ask a question, make an assertion, or even issue a command, we are also making reference to something that is often called a SITUATION or EVENT.[6] If you utter

[4]This is not quite true. Sometimes the manner in which something is said (the form of an utterance) can make some pragmatic contribution to an utterance. Grice's theory also included a 'Maxim of Manner', which was intended to account for such cases, e.g. (i):

(i) X produced a series of sounds that corresponded closely with the score of 'Home sweet home'.

Here, A conveys that there was something deficient in X's rendition of the song. A does this by intentionally avoiding the more concise sentence: *X sang 'Home sweet home'.*

[5]There is more to meaning than the literal meanings and pragmatic inferences that we have discussed in this section. In particular, there are contrasts in form that correspond to differences in when it is appropriate to use a sentence. One such contrast involves 'honorific' forms in Japanese and other languages. The difference between (i) and (ii), is that (i) is familiar and (ii) is formal, so that (i) would be used when talking to a friend or subordinate and (ii) would be used when talking to a stranger or someone higher in a social hiearchy:

(i) Hon-wo yonda.
Book-ACC read.PAST.FAMILIAR
'I read a book.'

(ii) Hon-wo yomimashita.
Book-ACC READ.PAST.FORMAL
'I read a book.'

[6]Although the term 'event' is often used in a general sense in semantic discussions, this terminology can be misleading, especially in connection with circumstances like the following, where nothing very event-like is happening:

a declarative sentence like *Kim is running*, for example, you are claiming that there is some running situation in the world that involves something (usually a person) named Kim. The proposition that you assert is either true or false depending on a number of things, for example, whether this situation is a running event (maybe Kim is moving too slowly for it to really qualify as running), or whether the runner is someone named 'Kim' (maybe the person you have in mind is really named 'Nim'), whether the running situation is really happening now (maybe Kim has already run the race but your watch stopped several hours ago). If any of these 'maybes' turns out to be the case, then the proposition you have asserted is false – the situation you are describing as specified by the linguistic meaning of the sentence is not part of the real world.

An important part of the business of semantics is specifying truth conditions such as these, that is, specifying restrictions which must be satisfied by particular situations in order for assertions about them to be true. Consider what this means in the case of *Kim is running*. This sentence is associated with a proposition that has the following truth conditions:[7]

(7) a. there is a situation s
    b. s is a running situation
    c. the runner is some individual i
    d. i is named Kim
    e. s is temporally located around the time of utterance

If there is some situation s and some individual i such that all the conditions in (7) are satisfied, then the proposition expressed by *Kim is running* is true. If not, then that proposition is false.

Truth conditions are determined in large part by linguistic meaning, that is, the meaning associated with a sentence by the semantic component of the grammar. If our grammar consisted merely of a list of sentences, we could list the meanings of those sentences alongside their forms. However, as we saw in Chapter 2, lists do not provide plausible theories of the grammars of natural languages. Instead, we've developed a theory of grammar that allows us to systematically build up phrases and sentences from an inventory of words and phrase structure rules. Therefore we will need a semantic component to our grammar that systematically builds the meanings of sentences out of the meanings of words and the way they are put together (i.e. the phrase structure rules). In order to do this, we will need (i) some way of characterizing the linguistic meanings of words and (ii) a set of constraints that allows us to correctly specify the

(i) Bo knows baseball.
(ii) Dana is aggressive.
(iii) Sydney resembles Terry.
(iv) Chris is tall.
(v) 37 is a prime number.

It seems much more intuitive to discuss such sentences in terms of 'situations'; hence we have adopted this as our official terminology for the semantics of sentences.

[7]The exact meaning of the progressive (*be...-ing*) construction is a fascinating semantic topic with a considerable literature that we cannot do justice to here. We have adopted clause (7e) as a convenient first approximation of the truth conditional contribution of the present progressive in English.

linguistic meanings of phrase structures in terms of the meanings of their parts (their subconstituents).

In terms of the example *Kim is running*, we will need a way to ensure that the various pieces of this sentence – the noun *Kim*, the verb *is*, and the verb *running* – each make their appropriate contribution to the set of constraints summarized in (7), that the result is a proposition (not a question or a directive), and that the pieces of meaning get combined in the appropriate way (for example, that the same individual i has the properties of being named Kim and being the runner). In addition, our account must assign a meaning to *Sandy is running* that differs from that assigned to *Kim is running* only in the name of the individual i. Likewise, our account must analyze the sentence *Is Kim running?* as a question, and furthermore a question about whether or not there is a situation s and an individual i such that all the conditions in (7) are satisfied.

5.3.2 Semantic Features

The semantic objects of our grammar will be classified in terms of four SEMANTIC MODES – that is, the four basic kinds of meanings that are enumerated and illustrated in (8):

(8)

SEMANTIC MODE	KIND OF PHRASE	EXAMPLE
proposition	noninverted sentence	*Kim is happy.*
question	inverted sentence	*Is Kim happy?*
directive	imperative sentence	*Be happy!*
reference	NP	*Kim*

As we saw above, there are a number of differences among the various semantic modes. Despite these differences, the modes have something in common. Every kind of linguistic expression we have considered, irrespective of its semantic mode, refers to something that must satisfy an indicated list of restrictions for the expression to be correctly applicable. To express this generalization, we will model all expressions in terms of a single type of semantic object (a *sem-cat* or *semantic-category*) which bears three features: MODE, INDEX, and RESTR. The value of MODE provides the semantic mode of the object. The value of INDEX is an index corresponding to the situation or individual referred to. The value of RESTR (short for 'restriction') is a list of conditions that the situation or individual has to satisfy in order for the expression to be applicable to it. Semantic structures then will look like (9):

(9)
$$\begin{bmatrix} \textit{sem-cat} & \\ \text{MODE} & \{\text{prop, ques, dir, ref, none}\} \\ \text{INDEX} & \{i, j, k, \ldots, s_1, s_2, \ldots\} \\ \text{RESTR} & \langle \ldots \rangle \end{bmatrix}$$

There are a couple of things to note about the values of these features. The first is that, although we represent the value of RESTR as a list, the order of the elements on that list will not be semantically significant. The second is that the feature INDEX differs from other features we have encountered, in that it can take an unlimited number of different values. This is because there is no limit (in principle) to the number of different

individuals or situations which can be referred to in a single sentence. Consequently, we must have (in principle, at least) an infinite number of indices available to serve as values of the feature INDEX. These values of INDEX will conventionally be written with lower-case letters; instead of tagging two occurrences of the same INDEX value, we will simply write the same lower-case letter in both places.

Propositions are analyzed in terms of feature structures like the one in (10) (where 'prop' is short for 'proposition').

(10) $\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \langle \ldots \rangle \end{bmatrix}$

A proposition like (10) will be true just in case there is some actual situation s (and there exist appropriate other individuals corresponding to whatever indices are present in (10)) such that the constraints specified in the RESTR value of (10) are all satisfied. These restrictions, the nature of which will be explained in Section 5.3.3, must include all those that are relevant to the meaning of the sentence, for example, all the constraints just mentioned in conjunction with the truth or falsity of *Kim is running*. Our grammatical analysis needs to ensure that we end up with exactly the right constraints in the RESTR list of a sentence's semantics, so that we associate exactly the right meaning with any sentence sanctioned by our grammar.

A question like *Is Kim running?* is assigned a semantics just like the one assigned to *Kim is running*, except that the MODE value must be 'question' ('ques' for short), rather than 'prop':

(11) $\begin{bmatrix} \text{MODE} & \text{ques} \\ \text{INDEX} & s \\ \text{RESTR} & \langle \ldots \rangle \end{bmatrix}$

In this case, the value of RESTR is again interpreted as the set of conditions placed on the situation s, but if someone poses a question, they are merely inquiring as to whether s satisfies those conditions.

Directives ('dir' for short) are represented as in (12):

(12) $\begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \\ \text{RESTR} & \langle \ldots \rangle \end{bmatrix}$

What the RESTR list does in the case of a directive is to specify what conditions have to be satisfied in order for a directive to be fulfilled.

A reference ('ref' for short) is similar to the kinds of meanings just illustrated, except that it can be used to pick out all kinds of entities – not just situations. So the semantics we assign to a referring NP has the following form:[8]

[8]There are any number of intriguing referential puzzles that are the subject of ongoing inquiry by semanticists. For example, what does an NP like *a page* refer to in the sentence: *A page is missing from this book*? And what does *the unicorn that Chris is looking for* refer to in the sentence: *The unicorn that Chris is looking for doesn't exist*?

(13)
$$\begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \langle \ldots \rangle \end{bmatrix}$$

In this case, the RESTR list contains the conditions that the entity must meet in order for it to be legitimately referred to by the expression.

Note that we write INDICES in terms of the letters i, j, k, etc. when we are specifying the semantics of nominal expressions. The INDEX values written as s, s_1, s_2, etc. always refer to situations.

The differing values of MODE that we have just seen serve to differentiate between the kinds of meaning that are associated with various syntactic categories (like declarative, interrogative or imperative sentences or noun phrases). Many words and phrases that cannot be used by themselves to express a proposition, ask a question, refer to an individual, etc. (e.g. determiners and conjunctions) will be treated here in terms of the specification [MODE none].

5.3.3 Predications

We now turn to the question of what kind of entities make up the value of the RESTR list. Semantic restrictions associated with expressions come in many varieties, which concern what properties some individual has, who did what to whom in some situation, when, where, or why some situation occurred, and so forth. That is, semantically relevant restrictions specify which properties must hold of individuals and situations, and which relations must hold among them, in order for an expression to be applicable.

To represent this sort of information, we must introduce into our semantics some way of specifying relations among entities quite generally. We do this by introducing a type of feature structure called *predication*. The features of a *predication* specify (i) what kind of relation is involved and (ii) who or what is participating in the relation. Examples of feature structures of type *predication* are given in (14):[9]

(14) a.
$$\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{love} \\ \text{SIT(UATION)} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{walk} \\ \text{SIT} & s \\ \text{WALKER} & i \end{bmatrix}$$

[9]The kind of event-based semantic analysis we employ was pioneered by the philosopher Donald Davidson in a number of papers. (See, for example, Davidson 1980.) Our simplified representations differ from other work in this tradition where all talk of existence is represented via explicit existential quantification, i.e. in terms of representations like (i):

(i) there is an event s and an individual i such that: s is a running event, the runner of s is i, i is named Kim, and s is temporally located around the time of utterance

We will treat all such existential quantification as implicit in our semantic descriptions.

c.
$$\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{RECIPIENT} & j \\ \text{GIFT} & k \end{bmatrix}$$

d.
$$\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{book} \\ \text{SIT} & s \\ \text{INST} & k \end{bmatrix}$$

e.
$$\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{happy} \\ \text{SIT} & s \\ \text{INST} & i \end{bmatrix}$$

f.
$$\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{under} \\ \text{SIT} & s \\ \text{LOWER} & i \\ \text{HIGHER} & j \end{bmatrix}$$

The predications in (14) are meant to correspond to conditions such as: 's is a situation wherein i loves j', 's is a situation wherein i walks', 's is a situation wherein i gives k to j', 's is a situation wherein k is an instance of bookhood (i.e. where k is a book)', 's is a situation wherein i is happy', and 's is a situation wherein i is under j', respectively. We will henceforth make frequent use of predications like these, without taking the time to present a proper theory of relations, predications, and the features that go with them. Note that the restriction associated with many nouns and adjectives (*book*, *happy*, etc.) includes a predication of only one (nonsituation) argument. In such cases – for example, (14d,e) – we use the feature INST(ANCE).

As indicated in (14), we are assuming that all predications are in principle 'situated', i.e. that they make reference to some particular situation (the index that is the value of the feature SIT inside each predication). This provides a semantic flexibility that allows us to analyze sentences like (15):

(15) The senator visited a classmate a week before being sworn in.

That is, one way to understand this (perhaps the most natural way) is in terms of the proposition that some person i who is now a senator was part of a visiting situation where the person who got visited – j – was once part of a certain academic situation that also included the senator. The three situations are all distinct: the situation where i instantiates senatorhood comes after the visiting situation and both these situations could come long after the situation where i and j were classmates. Yet the proposition expressed by (15) is making reference to all three situations at once, and the situational predications we have assumed give us a way to model this.[10] Though this use of multiple situations in the semantics of a single proposition is fascinating and may well be essential for semantic analysis to be successful,[11] secondary situations bring unwanted complexity

[10] Of course, sometimes we refer to someone as a senator even after they have left office. This could be analyzed as making reference to a past situation in which the individual referred to instantiated senatorhood.

[11] There is, of course, an issue as to how far to take the situation-based kind of analysis. General statements like *All cows eat grass* or *Two plus two is four* seem not to make reference to any particular situations.

and hence will be suppressed in subsequent discussion, unless they bear directly on a particular discussion. In general, we will only display the SIT feature on predications contributed by the head of a given phrase or when its value is identified with the value of some other feature.

Almost all words specify restrictions that involve predications of one kind or another, including verbs, adjectives, adverbs, prepositions, and nouns. In order for phrases containing such words to inherit these restrictions, there must be constraints that (minimally) guarantee that the RESTR values of a phrase's daughters are part of that phrase's RESTR value. Only in this way will we end up with a sentence whose meaning is a proposition (or question or directive) whose RESTR value includes all the necessary restrictions on the relevant event participants.

For example, we will want our grammar to ensure that a simple sentence like (16) is associated with a proposition like the one described in (17):

(16) Chris saved Pat.

(17)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{save} \\ \text{SIT} & s \\ \text{SAVER} & i \\ \text{SAVED} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Chris} \\ \text{NAMED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Pat} \\ \text{NAMED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

The restriction that s is a saving situation comes from the lexical entry for the verb *save*, the constraint that i – the saver – must be named Chris comes from the proper noun *Chris*, and the constraint that j – the saved (person) – must be named Pat comes from the lexical entry for the proper noun *Pat*. By associating (16) with the feature structure in (17), our semantic analysis says that the linguistic meaning of (16) is the proposition that will be true just in case there is an actual situation that involves the saving of someone named Pat by someone named Chris. But in order to produce the right set of restrictions in the sentence's semantic description, the restrictions of the parts of the sentence have to be amalgamated into a single list of restrictions. Note in addition that the main situation of the sentence is derived from that introduced by the verb. It is true in general that the semantics of a phrase will crucially involve the semantics of its head daughter. We will capture these semantic relationships between the parts of the sentence with two general principles, introduced in Section 5.5 below. First, however, we must consider how semantic structures fit into the tree structures our grammar licenses.

5.4 How Semantics Fits In

In earlier chapters, we considered only the syntactic properties of linguistic expressions. To accommodate the basic analysis of linguistic meaning just introduced, we need some way of introducing semantic structures into the feature structures we use to analyze words and phrases. We do this by adding two new features – SYN(TAX) and SEM(ANTICS) – and adding a level of embedding within our feature structures, as illustrated in (18):

(18)
$$\begin{bmatrix} \textit{expression} & \\ \text{SYN} & \begin{bmatrix} \textit{syn-cat} & \\ \text{HEAD} & [\ldots] \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \ldots \\ \text{COMPS} & \ldots \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \textit{sem-cat} & \\ \text{MODE} & \\ \text{INDEX} & \ldots \\ \text{RESTR} & \langle\ \ldots\ \rangle \end{bmatrix} \end{bmatrix}$$

There is now a syntactic side and a semantic side to all feature structures like (18), i.e. to all feature structures of type *expression*. Note that we have created another type – *syntactic-category* (*syn-cat*) – which is parallel to *sem-cat*, and which classifies the values of the feature SYN, just as *sem-cat* classifies the values of the feature SEM. Although we will add a few more features as we progress, this is in essence the feature geometry that we will adopt in the remainder of the book.

This changes the way lexical entries look, of course; their new feature geometry is illustrated in (19), though some details are not yet included:[12]

(19) a.
$$\left\langle \text{dog} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ [\text{HEAD}\ \textit{det}]\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

[12]It should be noted that our semantic analysis of proper nouns (one of many that have been proposed over the centuries) treats them as simple referring expressions whose referent must be appropriately named. In a more precise account, we might add the further condition that the speaker must intend to refer to the referent. Under this analysis, the proposition expressed by a sentence like *Kim walks* would be regarded as true just in case there is a walking event involving a certain individual that the speaker intends to refer to who is named 'Kim'.

b.
$$\left\langle \text{Kim} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

c.
$$\left\langle \text{love} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP}\ \rangle \\ \text{COMPS} & \langle\ \text{NP[acc]}\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

These entries also illustrate the function of the INDEX feature in fitting together the different pieces of the semantics. Notice that the INDEX value of *love* is identified with the SIT argument of the loving predication in its RESTR list. Similarly, the INDEX value of *dog* is the same as the INST value in the predication introduced by *dog*, and that the INDEX value of *Kim* is the same as the NAMED value in the predication introduced by *Kim*. By identifying these values, we enable the NPs to 'expose' those indices to other words that might select the NPs as arguments. Those words, in turn, can associate those indices with the appropriate role arguments within their predications (i.e. features like WALKER, LOVED, etc.). This is illustrated in (20) for the verb *love*:

(20)
$$\left\langle \text{love} , \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \left\langle \left[\begin{array}{l} \text{NP} \\ \text{INDEX } i \end{array} \right] \right\rangle \\ \text{COMPS} & \left\langle \left[\begin{array}{l} \text{NP} \\ \text{CASE acc} \\ \text{INDEX } j \end{array} \right] \right\rangle \end{array} \right] \end{array} \right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{array} \right] \right\rangle \end{array} \right] \end{array} \right] \right\rangle$$

In this way, as the verb combines with a particular NP object, the index of that NP is identified with the value of the feature LOVED in the verb's semantics. Likewise, since the verb's specifier requirement is identified with the VP's specifier requirement (by the Valence Principle), when the VP combines with a particular NP subject, the index of that NP will be identified with the value of the feature LOVER in the verb's semantics. All that is left is to ensure that the predications introduced by each word are collected together to give the RESTR list of the whole sentence, and to ensure that the INDEX and MODE values of phrases are appropriately constrained. These are the topics of the next section.

Note that the addition of semantic information to our grammar has changed the way we use abbreviations in two ways. First, the labels NP, S, V, etc. now abbreviate feature structures that include both semantic and syntactic information, i.e. *expression*s which bear the features SYN and SEM. Second, we will add a notation to our system of abbreviations to allow us to refer to the INDEX value of an abbreviated expression: NP_i will be used as a shorthand for an NP whose SEM value's INDEX is i. We occasionally use this same subscript notation with other categories, too, e.g. PP_i. (The abbreviations are summarized in the grammar summary in Section 5.10.)

5.5 The Semantic Principles

We are now not only able to analyze the form of sentences of considerable complexity using our grammar, but in addition we can analyze the meanings of complex sentences by adding semantic constraints on the structures defined by our rules. The most general of these semantic constraints is given in (21):

(21) Semantic Compositionality Principle

In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters.

In other words, all restrictions from all the daughters in a phrase are collected into the RESTR value of the mother. The term 'sum' has a straightforward meaning here: the sum of the RESTR values of the daughters is the list whose members are those values, taken in order.[13] We will use the symbol '⊕' to designate the sum operator.[14]

In addition to the Semantic Compositionality Principle, we introduce the following constraint on the MODE and INDEX values of headed phrases:

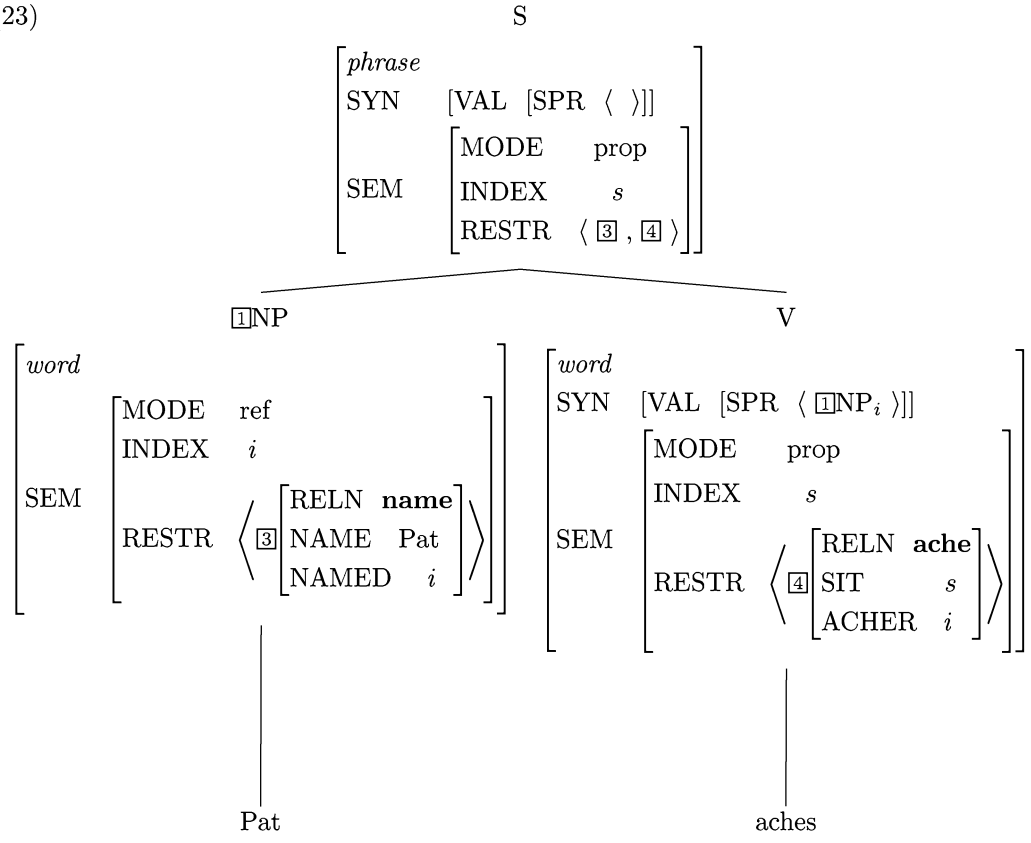
(22) Semantic Inheritance Principle

In any headed phrase, the mother's MODE and INDEX values are identical to those of the head daughter.

The Semantic Inheritance Principle guarantees that the semantic MODE and INDEX of a phrase are identified with those of the head daughter, giving the semantics, like the syntax, a 'head-driven' character.

The effect of these two semantic principles is illustrated in the simple example, (23):

(23)



[13] That is, the sum of lists ⟨ A ⟩, ⟨ B, C ⟩, and ⟨ D ⟩ is the list ⟨ A, B, C, D ⟩.

[14] Notice that, unlike the familiar arithmetic sum operator, ⊕ is not commutative: ⟨ A ⟩ ⊕ ⟨ B ⟩ = ⟨ A, B ⟩, but ⟨ B ⟩ ⊕ ⟨ A ⟩ = ⟨ B, A ⟩. And ⟨ A, B ⟩ ≠ ⟨ B, A ⟩, because the order of the elements matters. Although, as noted above, the order of elements in RESTR lists has no semantic significance, we will later use ⊕ to construct lists in which the ordering does matter (specifically, the ARG-ST lists introduced in Chapter 7 as part of our account of reflexive binding).

The effect of both semantic principles can be clearly observed in the S node at the top of this tree. The MODE is 'prop', inherited from its head daughter, the V node *aches*, by the Semantic Inheritance Principle. Similarly (as indicated by shading in (23)), the INDEX value *s* comes from the verb. The RESTR value of the S node, [RESTR ⟨ 3 , 4 ⟩], is the sum of the RESTR values of the NP and VP nodes, as specified by the Semantic Compositionality Principle.

In this way, our analysis provides a general account of how meanings are constructed. The Semantic Compositionality Principle and the Semantic Inheritance Principle together embody a simple yet powerful theory of the relation between the structures of our grammar and the meanings they convey.

5.6 Modification

The principles in Section 5.5 account for the semantics of head-complement and head-specifier phrases. We still need to consider the Coordination Rule (which, as a non-headed rule, isn't subject to the Semantic Inheritance Principle) and the Head-Modifier Rule, which hadn't yet reached its final form in the Chapter 4 grammar. This section addresses the Head-Modifier Rule. The Coordination Rule will be the subject of the next section.

The Head-Modifier Rule of the Chapter 4 grammar looked like this:

(24) Head-Modifier Rule (Chapter 4 version)

$$\left[\textit{phrase}\right] \rightarrow \mathbf{H}\left[\text{VAL} \left[\text{COMPS} \langle\ \rangle\right]\right] \text{ PP}$$

The only kind of modifier this rule accounts for is, of course, PPs. We'd like to extend it to adjectives and adverbs as well. Adverbs and adjectives, however, present a complication. Compared to PPs, they are relatively fussy about what they will modify. Adverbs modify verbs and not nouns (as illustrated in (25)) and adjectives modify nouns, but not verbs, (as illustrated in (26)).

(25) a. A rat died yesterday.
b.*A rat yesterday died.

(26) a. The person responsible confessed.
b.*The person confessed responsible.

In order to capture these facts, we introduce a feature called MOD which will allow modifiers to specify what kind of expressions they can modify. The value of MOD will be a (possibly empty) list of *expressions*. For elements that can be modifiers, this list contains just one *expression*. For elements that can't be modifiers, the list is empty. This allows us to make it a lexical property of adjectives that they are [MOD ⟨ NOM ⟩] (or [MOD ⟨ NP ⟩]) and a lexical property of adverbs that they were [MOD ⟨ VP ⟩] (or [MOD ⟨ S ⟩]).

MOD will be a VAL feature, like SPR and COMPS. The intuitive connection between these three features is that they all specify what the head can combine with, although the means of combination is somewhat different for MOD as opposed to SPR and COMPS. Like SPR and COMPS, MOD is passed up from the head daughter to the mother via the Valence Principle, as adjusted in (27):

(27) The Valence Principle
Unless the rule says otherwise, the mother's values for the VAL features (SPR, COMPS, MOD) are identical to those of the head daughter.

Unlike with SPR and COMPS, no rule will contradict the Valence Principle with respect to the value of MOD. This means that the MOD value of the mother will always be the same as the MOD value of the head daughter. This is desirable, as the kind of expression a phrasal modifier (such as *responsible for the mess* or *on the table*) can modify is determined by the head of the modifier (in this case, the adjective *responsible* or the preposition *on*).

Furthermore, MOD, like SPR and COMPS, must be shared between conjuncts in a coordinate structure. If it weren't, we would mistakenly license ungrammatical strings such as those in (28):

(28) a.*The cat slept soundly and furry.
b.*The soundly and furry cat slept.

Since the Coordination Rule identifies the VAL values of the conjuncts, making MOD a VAL feature immediately captures these facts.

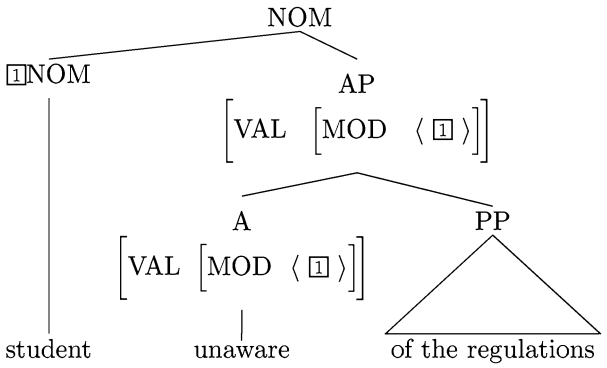
With modifiers now specifying what they can modify, the Head-Modifier Rule can be reformulated as in (29):[15]

(29) Head-Modifier Rule (Near-Final Version)

$$[\textit{phrase}] \rightarrow \mathbf{H}\boxed{1}\left[\text{VAL}\left[\text{COMPS} \quad \langle\ \rangle\right]\right] \left[\text{VAL}\begin{bmatrix}\text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \boxed{1}\ \rangle\end{bmatrix}\right]$$

The rule in (29) will license a phrase structure tree whose mother is, for example, a NOM just in case the head daughter is an expression of category NOM and the modifier daughter's MOD value is also of category NOM:

(30)



This NOM can combine with a determiner as its specifier to build an NP like (31):

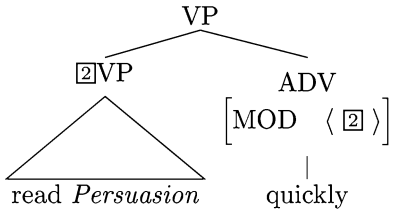
[15] In this rule, and in the A and AP nodes of (30), we have omitted the feature name 'SYN' to the left of 'VAL'. In the remainder of the book, we will often simplify our feature structure descriptions in this way, leaving out some of the outer layers of feature names when the information of interest is embedded within the feature structure description. We will only simplify in this way when no ambiguity about our intended meaning can arise.

This is the 'near-final version' of the Head-Modifier Rule. It will receive a further minor modification in Chapter 14.

(31) a student unaware of the regulations

The Head-Modifier Rule in (29) will also license the verb phrase in (32), under the assumption that adverbs are lexically specified as [MOD ⟨ VP ⟩]:

(32)



And a VP satisfying this description can combine with a subject like the one in (31) to build sentence (33):

(33) A student unaware of the regulations read *Persuasion* quickly.

Note that the value of MOD is an *expression*, which contains semantic as well as syntactic information. This will allow us to give an analysis of how the semantics of modifiers work. We will illustrate this analysis with the sentence in (34):

(34) Pat aches today.

Let us assume that an adverb like *today* has a lexical entry like the one in (35):[16] (We assume here that there is a subtype of *pos* for adverbs (*adv*).)

(35)

$$\left\langle \text{today} , \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{adv} \\ \text{VAL} & \left[\begin{array}{ll} \text{MOD} & \left\langle \left[\begin{array}{ll} \text{VP} \\ \text{INDEX} & s_1 \end{array} \right] \right\rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{array} \right] \end{array} \right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \text{none} \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{today} \\ \text{ARG} & s_1 \end{array} \right] \right\rangle \end{array} \right] \end{array} \right] \right\rangle$$

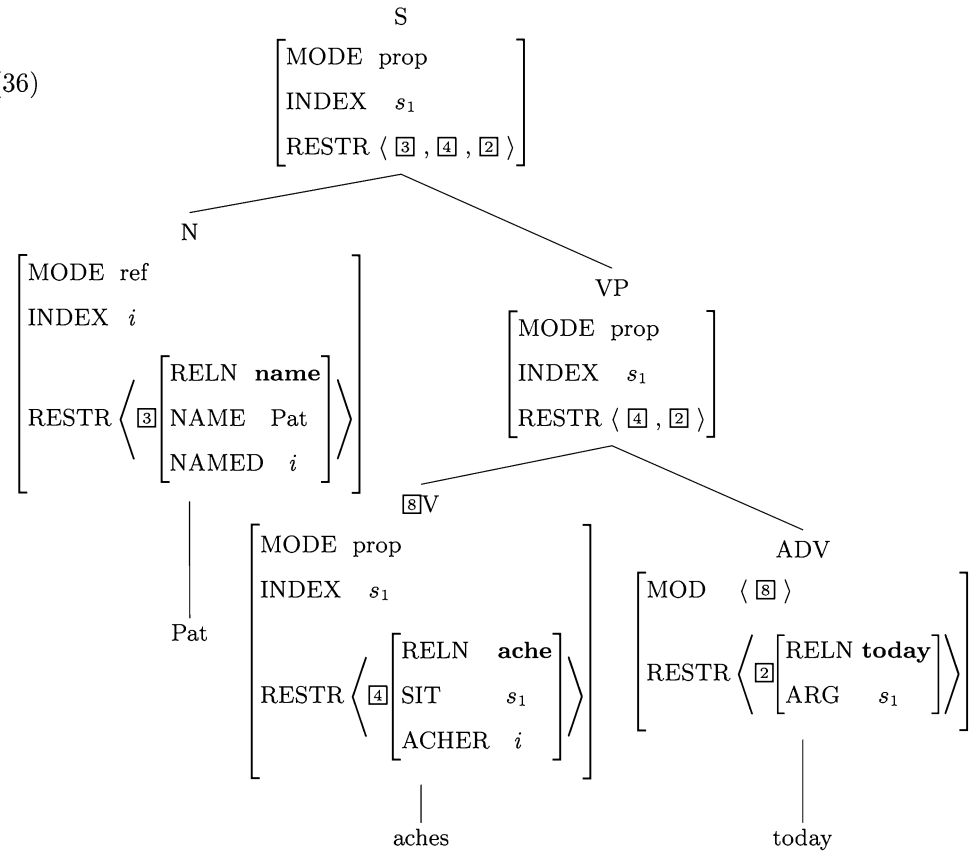
The key point here is that the MOD value identifies the index of the VP to be modified as 's_1', the same situation that is the argument of the relation 'today' in the semantic restriction. This means that once the adverb combines with a VP, the (situational) index of that VP is the argument of 'today'.

[16] We are suppressing the feature INDEX (along with SIT) here for clarity. For a more detailed analysis of adverbial modification, see Bender et al. 2002.

Exercise 1: The Missing INDEX
We have omitted INDEX from the SEM value in (35), although we said earlier that the value of SEM always consists of MODE, INDEX, and RESTR. Our omission was to simplify the presentation. Including INDEX under SEM would only have cluttered up the feature structure, without adding any useful information. In fact, we could assign any value we want to the missing INDEX, and the semantics of VPs like *aches today* would still be the same. Why?

Our two semantic principles, the Head-Modifier Rule, and the lexical entry in (35) as well as appropriate lexical entries for *aches* and *Pat* thus interact to define structure like (36) (only SEM values are indicated):

(36)



Exercise 2: VP or Not VP?
The lexical entry in (35) has a VP on the MOD list, but the corresponding node in the tree (36) is labeled V. Why isn't this an inconsistency? [*Hint: Remember that VP and V are abbreviations for feature structures, and check what they are abbreviations for.*]

5.7 Coordination Revisited

The analysis of the previous sections specifies how meanings are associated with the headed structures of our grammar, by placing appropriate constraints on those trees that result from our headed rules. It also covers the composition of the RESTR values in nonheaded rules. But nothing in the previous discussion specifies the MODE or INDEX values of coordinate phrases – the kind of phrase licensed by the Coordination Rule, a nonheaded rule.

In the previous chapter, we wrote this rule as follows:

(37) $\left[\text{VAL } \boxed{1}\right] \rightarrow \left[\text{VAL } \boxed{1}\right]^{+} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{conj} \end{bmatrix} \left[\text{VAL } \boxed{1}\right]$

This is equivalent to the following formulation, where the Kleene plus has been replaced by a schematic enumeration of the conjunct daughters:

(38) $\left[\text{VAL } \boxed{1}\right] \rightarrow \left[\text{VAL } \boxed{1}\right]_1 \ldots \left[\text{VAL } \boxed{1}\right]_{n-1} \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{conj} \end{bmatrix} \left[\text{VAL } \boxed{1}\right]_n$

We will employ this new notation because it lets us enumerate schematically the arguments that the semantic analysis of conjunctions requires.

Unlike the other predications we have used for semantic analysis, where each predication specifies a fixed (and small) number of roles, the predications that express the meanings of conjunctions like *and* and *or* allow any number of arguments. Thus each conjunct of coordinate structures like the following is a semantic argument of the conjunction:

(39) a. Chris [[walks]$_1$, [eats broccoli]$_2$, and [plays squash]$_3$].
b. [[Chris walks]$_1$, [Pat eats broccoli]$_2$, and [Sandy plays squash]$_3$].

Because the number of arguments is not fixed, the predications for conjunctions allow not just indices as arguments, but lists of indices. Consequently, the sentences in (39) may be represented in terms of a semantic structure like the following:

(40)
$$\begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s_0 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s_0 \\ \text{ARGS} & \langle s_1, s_2, s_3 \rangle \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{walk} \\ \text{SIT} & s_1 \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{RELN} & \textbf{eat} \\ \text{SIT} & s_2 \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{play} \\ \text{SIT} & s_3 \end{bmatrix} \right\rangle \end{bmatrix}$$

In (40), the situations s_1, s_2, and s_3 are the simplex situations of walking, eating and playing, respectively. The situation s_0, on the other hand, is the complex situation that involves all three of the simplex situations. Note that it is this situation (s_0) that is the INDEX of the whole coordinated phrase. That way, if a modifier attaches to the coordinated phrase, it will take the index of the complex situation as its semantic argument.

In order to be sure our grammar assigns semantic representations like (40) to sentences like (39), we need to update our lexical entries for conjunctions and revise the Coordination Rule. Let us assume then that the lexical entry for a conjunction looks roughly as shown in (41):

(41)
$$\left\langle \text{and} \ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{MODE} & \text{none} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

As for the Coordination Rule, we need to revise it so that it relates the indices of the conjuncts to the predication introduced by the conjunction. In addition, we need to say something about the index of the mother. This leads us to the following reformulation of our Coordination Rule (where 'IND' is short for 'INDEX'):

(42) Coordination Rule

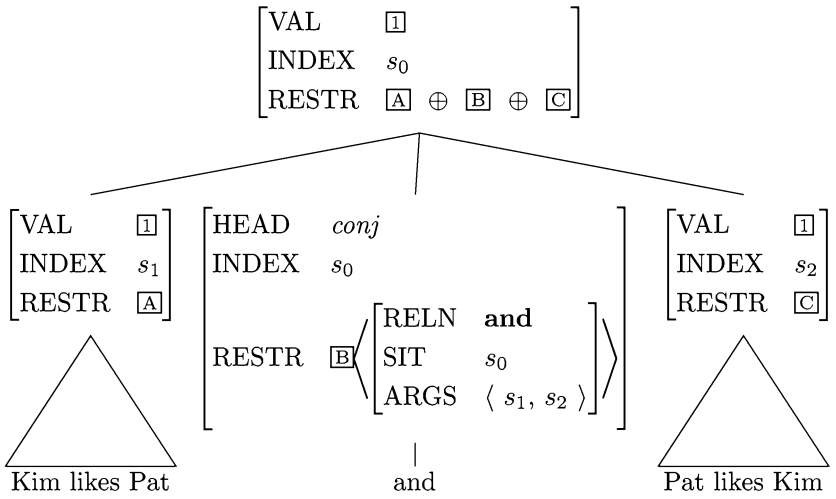
$$\begin{bmatrix} \text{SYN} & [\text{VAL} \ \boxed{0}] \\ \text{SEM} & [\text{IND} \ s_0] \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYN} & [\text{VAL} \ \boxed{0}] \\ \text{SEM} & [\text{IND} \ s_1] \end{bmatrix} \ \ldots \ \begin{bmatrix} \text{SYN} & [\text{VAL} \ \boxed{0}] \\ \text{SEM} & [\text{IND} \ s_{n-1}] \end{bmatrix}$$
$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s_0 \\ \text{RESTR} & \langle [\text{ARGS} \ \langle s_1, \ldots, s_n \rangle] \rangle \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{SYN} & [\text{VAL} \ \boxed{0}] \\ \text{SEM} & [\text{IND} \ s_n] \end{bmatrix}$$

This rule accomplishes a number of goals, including:

- requiring that all conjuncts of a coordinate structure have identical values for SPR, COMPS, and MOD.
- collecting the RESTR values of all daughters into the RESTR list of the mother (guaranteed because the structures built in accordance with this rule must satisfy the Semantic Compositionality Principle),
- identifying the indices of the conjuncts with the semantic arguments of the conjunction, and
- identifying the index of the conjunction with that of the coordinate structure.

These effects are illustrated in the following tree, which shows a (coordinate) phrase structure satisfying the Coordination Rule:

(43)



Our revised Coordination Rule goes a long way toward accounting for sentences containing coordinate structures and associating them with appropriate meanings. We will return to coordination in Chapters 8 and 14 to add further refinements.

5.8 Quantifiers

The final semantic topic we will address in this chapter is quantifiers and quantifier scope ambiguities. Consider the example in (44):

(44) A dog saved every family.

Sentences like this are usually treated as ambiguous, the two distinct READINGS being paraphrased roughly as (45a,b):

(45) a. There was some particular dog who saved every family.
b. Every family was saved by some dog or other (not necessarily the same dog).

Ambiguities of this kind might be familiar from the study of predicate logic, where the two readings in question are often represented in the fashion shown in (46a.b):

(46) a. (**Exist** i: **dog**(i))[(**All** j: **family**(j))[**save**(i,j)]]
b. (**All** j: **family**(j))[(**Exist** i: **dog**(i))[**save**(i,j)]]

The first three parts of these representations are a quantificational relation (e.g. **Exist**, **All**), a variable (e.g. i, j), and a formula called the quantifier's RESTRICTION (e.g. **dog**(i), **family**(j)). The expression in square brackets that follows a quantifier is its SCOPE. In (46a), the scope of the quantifier (**All** j: **family**(j)) is the expression repeated in (47):

(47) [**save**(i,j)]

In the same example, the scope of the quantifier (**Exist** i: **dog**(i)) is the expression repeated in (48):

(48) [(**All** j: **family**(j))[**save**(i,j)]]

The two distinct semantic analyses associated with a sentence like (44) thus differ only in terms of scope: in (46a), the existential quantifier has 'wide' scope; in (46b), the universal quantifier has wide scope.

The semantics we adopt in this book is compatible with recent work on quantification known as the theory of generalized quantifiers. This theory models the interpretation of quantifiers set-theoretically in a way that makes it possible to represent nonstandard quantifiers like '**most**', as well as the standard universal and existential quantifiers of predicate logic. Although our representations look different from those in (46), we can express the notions of quantifier, variable, restriction and scope using feature structures. We achieve this by treating quantifiers in terms of predications like (49):

(49)
$$\begin{bmatrix} \textit{predication} & \\ \text{RELN} & \textbf{exist} \\ \text{BV} & i \\ \text{QRESTR} & \textit{predication} \\ \text{QSCOPE} & \textit{predication} \end{bmatrix}$$

In (49), the quantifier predication has three new features: BOUND-VARIABLE (BV), QUANTIFIER-RESTRICTION (QRESTR) and QUANTIFIER-SCOPE (QSCOPE). The values of the latter two features can be identified with other predications in the RESTR list.

We can then identify the two quantifiers' QSCOPE values in different ways to express the two different scopal readings of (44). If the existential quantifier has wide scope, as in (46a), we can identify the QSCOPE values as shown in (50):

(50)
$$\begin{bmatrix} \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \\ \text{QRESTR} & \boxed{1} \\ \text{QSCOPE} & \boxed{2} \end{bmatrix}, \boxed{1}\begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix}, \boxed{2}\begin{bmatrix} \text{RELN} & \textbf{all} \\ \text{BV} & j \\ \text{QRESTR} & \boxed{3} \\ \text{QSCOPE} & \boxed{4} \end{bmatrix}, \right. \\ \left. \boxed{3}\begin{bmatrix} \text{RELN} & \textbf{family} \\ \text{INST} & j \end{bmatrix}, \boxed{4}\begin{bmatrix} \text{RELN} & \textbf{save} \\ \text{SAVER} & i \\ \text{SAVED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

And to represent the reading where the universal quantifier outscopes the existential, as in (46b), we can simply identify the QSCOPE values differently, as shown in (51):

(51)
$$\begin{bmatrix} \text{RESTR} & \left\langle \boxed{2}\begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \\ \text{QRESTR} & \boxed{1} \\ \text{QSCOPE} & \boxed{4} \end{bmatrix}, \boxed{1}\begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{all} \\ \text{BV} & j \\ \text{QRESTR} & \boxed{3} \\ \text{QSCOPE} & \boxed{2} \end{bmatrix}, \right. \\ \left. \boxed{3}\begin{bmatrix} \text{RELN} & \textbf{family} \\ \text{INST} & j \end{bmatrix}, \boxed{4}\begin{bmatrix} \text{RELN} & \textbf{save} \\ \text{SAVER} & i \\ \text{SAVED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

Notice that only the QSCOPE specifications have changed; the order of quantifiers on the RESTR list remains constant. That is because there is no semantic significance attached to the order of elements on the RESTR list. But (50) and (51) differ crucially in that the existential quantifier in (50) is not within the scope of any other quantifier, while in (51) it is the universal quantifier that has wide scope.

The differing constraints on QSCOPE values thus carry considerable semantic significance. Our grammar imposes constraints on the RESTR list of a multiply quantified sentence like (44) that can be satisfied in more than one way. Feature structures satisfying either (50) or (51) are allowed by the grammar. Moreover, if we make the further assumption that each index (variable) introduced by a quantificational NP (e.g. *every family*, *a dog*) must be BOUND, i.e. must occur within a feature structure that serves as the QSCOPE value of some quantificational predication with that index as its BV value, then these two are in fact the only possible RESTR lists that will satisfy the constraints of our grammar for a sentence like (44).

Though the feature structures satisfying our sentence descriptions must resolve the scope of quantifiers, note that the descriptions themselves need not. Our semantic representations thus enjoy an advantage that is not shared by standard predicate logic: if we don't specify any constraints on the QSCOPE values, we can essentially leave the quantifier scope unspecified. This kind of underspecification may have considerable appeal from a processing point of view: not only is it difficult for computational natural language applications to resolve the precise scope of quantifiers in even simple sentences, there is also psycholinguistic evidence that people don't always resolve scope.[17] Thus from the perspective of embedding our grammar within a model of human sentence processing or within a computational language processing system, it is significant that we can express generalized quantification in a way that allows unresolved, or even partially resolved, quantifier scope, depending on how many constraints are imposed on the values of QSCOPE.

Despite the interest and importance of these issues, we will leave quantification out of the picture in the semantic analyses we develop in the rest of the book. It will become apparent that we have our hands full with other aspects of meaning that interact in crucial ways with the syntactic phenomena that are our primary focus here.[18] We will therefore use simplified semantic representations for quantifiers as placeholders for the more complete analysis sketched. An example of how this would look for the determiner *a* is given in (52):

[17] See for example Kurtzman and MacDonald 1993.

[18] See the further reading section at the end of this chapter for references to recent work that integrates a view of quantification like the one just sketched with grammars of the sort we will motivate in subsequent chapters.

(52)
$$\left\langle \text{a} , \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} \\ \text{AGR} & \textit{3sing} \\ \text{COUNT} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Even with this simplified representation, there remains an interesting issue of compositional semantics: the value of the feature BV should end up being the same as the INDEX of the noun for which *a* is the specifier. However, this identity cannot be expressed as a constraint within the lexical entry for the determiner, since the determiner does not select for the noun (note that its COMPS and SPR lists are both empty). Instead, the determiner identifies its OWN index with the value of BV (i), and the lexical entry for a noun identifies its INDEX value with that of its SPR:

(53)
$$\left\langle \text{dog} , \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{INDEX} & i \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

This means that the noun's INDEX value and the determiner's BV value end up being the same. Because *dog* identifies its own index (and the INST value of the **dog** predication) with the index of its specifier, and *a* identifies its index with the BV value of the **exist** predication, the lexical entries together with the grammar rules produce semantic representations like the one shown in (54) for the noun phrase *a dog*, with the value of BV correctly resolved:

(54)
$$\begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix}$$

Because the Semantic Inheritance Principle passes the head's INDEX value up to the phrasal level, this analysis generalizes naturally to syntactically complex specifiers, such as possessive NPs (see Problem 5 of Chapter 6).

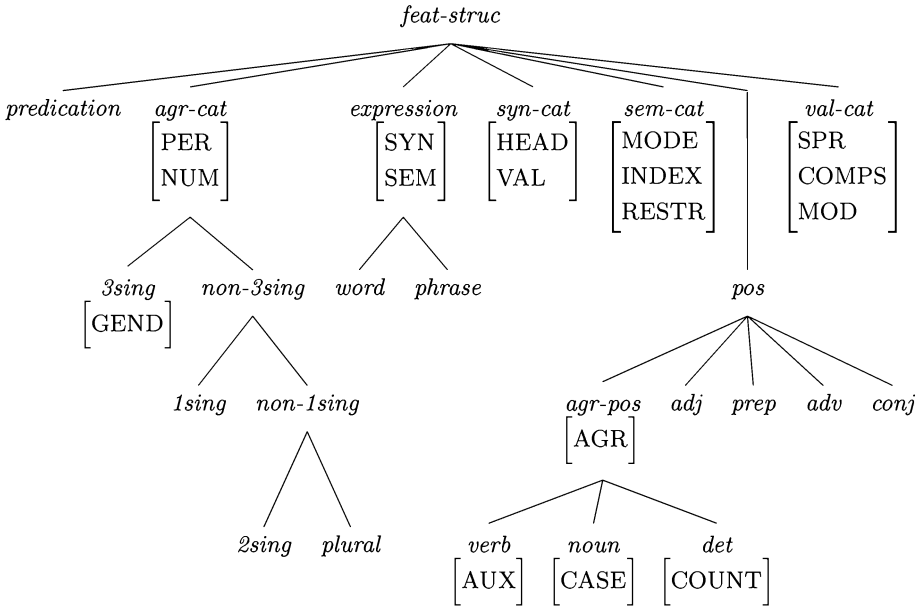
5.9 Summary

In this chapter, we introduced fundamental issues in the study of linguistic meaning and extended our grammar to include semantic descriptions. We then provided a systematic account of the relation between syntactic structure and semantic interpretation based on two constraints: the Semantic Compositionality Principle and the Semantic Inheritance Principle. These principles together provide a general account of how the semantics of a phrase is related to the semantics of its daughters. This chapter also extended the treatments of modification and coordinate structures to include an account of their linguistic meaning.

5.10 The Chapter 5 Grammar

5.10.1 The Type Hierarchy

The current version of our type hierarchy is summarized in (55):



5.10.2 Feature Declarations and Type Constraints

TYPE	FEATURES/CONSTRAINTS	IST
feat-struc		
expression	$\begin{bmatrix}\text{SYN} & \textit{syn-cat}\\ \text{SEM} & \textit{sem-cat}\end{bmatrix}$	*feat-struc*
syn-cat	$\begin{bmatrix}\text{HEAD} & \textit{pos}\\ \text{VAL} & \textit{val-cat}\end{bmatrix}$	*feat-struc*
sem-cat	$\begin{bmatrix}\text{MODE} & \{\text{prop, ques, dir, ref, none}\}\\ \text{INDEX} & \{i, j, k, \ldots, s_1, s_2, \ldots\}^{19}\\ \text{RESTR} & \textit{list(predication)}\end{bmatrix}$	*feat-struc*
predication	$\begin{bmatrix}\text{RELN} & \{\textbf{love}, \textbf{walk}, \ldots\}\\ \ldots\end{bmatrix}$	*feat-struc*
word, phrase		*expression*
val-cat	$\begin{bmatrix}\text{SPR} & \textit{list(expression)}\\ \text{COMPS} & \textit{list(expression)}\\ \text{MOD} & \textit{list(expression)}\end{bmatrix}$	*feat-struc*
pos		*feat-struc*
agr-pos	$[\text{AGR} \quad \textit{agr-cat}]$	*pos*
verb	$[\text{AUX} \quad \{+,-\}]$	*agr-pos*
noun	$[\text{CASE} \quad \{\text{nom, acc}\}]$	*agr-pos*
det	$[\text{COUNT} \quad \{+,-\}]$	*agr-pos*
adj, prep, adv, conj		*pos*
agr-cat	$\begin{bmatrix}\text{PER} & \{\text{1st, 2nd, 3rd}\}\\ \text{NUM} & \{\text{sg, pl}\}\end{bmatrix}$	*feat-struc*
3sing	$\begin{bmatrix}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sg}\\ \text{GEND} & \{\text{fem, masc, neut}\}\end{bmatrix}$	*agr-cat*
non-3sing		*agr-cat*
1sing	$\begin{bmatrix}\text{PER} & \text{1st}\\ \text{NUM} & \text{sg}\end{bmatrix}$	*non-3sing*
non-1sing		*non-3sing*
2sing	$\begin{bmatrix}\text{PER} & \text{2nd}\\ \text{NUM} & \text{sg}\end{bmatrix}$	*non-1sing*
plural	$[\text{NUM} \quad \text{pl}]$	*non-1sing*

[19] The possible values of the feature INDEX will be grouped together as the type *index* in the formal appendix to Chapter 6.

5.10.3 Abbreviations

(55)

$$\text{S} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{NP}_i = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \end{bmatrix} \end{bmatrix}$$

$$\text{VP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \text{X}\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{NOM} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \text{X}\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{V} = \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \end{bmatrix} \end{bmatrix} \qquad \text{N} = \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \end{bmatrix}$$

$$\text{PP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{AP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{P} = \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{prep} \end{bmatrix} \end{bmatrix} \qquad \text{A} = \begin{bmatrix} \textit{word} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adj} \end{bmatrix} \end{bmatrix}$$

$$\text{DP}^{20} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{det} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

5.10.4 The Grammar Rules

In this summary, we give fully explicit versions of the grammar rules. In later chapters and the summary in Appendix A, we will abbreviate by supressing levels of embedding, e.g. by mentioning features such as SPR and COMPS without mentioning SYN or VAL.

(56) Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H} \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \boxed{1}\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head preceded by its specifier.

[20] We replace our old abbreviation D with a new abbreviation DP in anticipation of Problem 4 of Chapter 6, which introduces the possibility of determiner phrases. The abbreviation DP, like NP and VP, is underspecified and may represent either a *word* or a *phrase*.

(57) Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{SYN} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{SYN} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{COMPS} & \langle \boxed{1}, \ldots, \boxed{n} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \boxed{1} \ldots \boxed{n}$$

A phrase can consist of a lexical head followed by all its complements.

(58) Head-Modifier Rule

$$[\textit{phrase}] \rightarrow \mathbf{H}\boxed{1} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle \boxed{1} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

A phrase can consist of a (lexical or phrasal) head followed by a compatible modifier.

(59) Coordination Rule

$$\begin{bmatrix} \text{SYN} & [\text{VAL} & \boxed{0}] \\ \text{SEM} & [\text{IND} & s_0] \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYN} & [\text{VAL} & \boxed{0}] \\ \text{SEM} & [\text{IND} & s_1] \end{bmatrix} \ldots \begin{bmatrix} \text{SYN} & [\text{VAL} & \boxed{0}] \\ \text{SEM} & [\text{IND} & s_{n-1}] \end{bmatrix}$$

$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s_0 \\ \text{RESTR} & \langle [\text{ARGS} & \langle s_1, \ldots, s_n \rangle] \rangle \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{SYN} & [\text{VAL} & \boxed{0}] \\ \text{SEM} & [\text{IND} & s_n] \end{bmatrix}$$

Any number of elements with matching valence specifications can form a coordinate phrase with identical valence specifications.

5.10.5 The Principles

(60) Head Feature Principle (HFP)

In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical.

(61) Valence Principle

Unless the rule says otherwise, the mother's values for the VAL features (SPR, COMPS, and MOD) are identical to those of the head daughter.

(62) Specifier-Head Agreement Constraint (SHAC)

Verbs and common nouns must be specified as:

$$\begin{bmatrix} \text{SYN} \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(63) Semantic Inheritance Principle

In any headed phrase, the mother's MODE and INDEX values are identical to those of the head daughter.

(64) Semantic Compositionality Principle

In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters.

5.10.6 Sample Lexical Entries

(65)

$$\left\langle \text{dog} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{AGR} \quad \textit{3sing} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{DP}_i\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(66)

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

(67)

$$\left\langle \text{love} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \text{NP[acc]}_j\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{love} \\ \text{SIT} & s \\ \text{LOVER} & i \\ \text{LOVED} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(68)

$$\left\langle \text{today} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{adv} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{MOD} & \left\langle \begin{bmatrix} \text{VP} \\ \text{INDEX} \quad s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{today} \\ \text{ARG} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(69)

$$\left\langle \text{and} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{MODE} & \text{none} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

(70)

$$\left\langle \text{a} , \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{AGR} & \textit{3sing} \\ \text{COUNT} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{MOD} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{none} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{exist} \\ \text{BV} & i \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

5.11 Further Reading

Much work on linguistic pragmatics builds directly on the pioneering work of the philosopher H. Paul Grice (see Grice 1989). A seminal work in modern research on natural language semantics is Frege's (1892) essay, 'Über Sinn und Bedeutung' (usually translated as 'On Sense and Reference'), which has been translated and reprinted in many anthologies (e.g. Geach and Black 1980). More recently, the papers of Richard Montague (Thomason, ed. 1974) had a revolutionary influence, but they are extremely technical. An elementary presentation of his theory is given by Dowty et al. (1981). General introductory texts in semantics include Chierchia and McConnell-Ginet 1990, Gamut 1991, and de Swart 1998.

All of these textbooks cover generalized quantifiers. For a more recent, more technical overview of generalized quantifiers, see Keenan and Westerståhl 1997. Shorter overviews of semantics include Bach 1989, Barwise and Etchemendy 1989 and Partee 1995. A short and very elementary introduction to generalized quantifiers is given in Larson 1995. The treatment of quantification sketched in Section 5.3 is developed more fully in Copestake et al. 1995, Copestake et al. 1999, and Copestake et al. 2001.

5.12 Problems

Problem 1: Two Kinds of Modifiers in English

In English, modifiers of nouns can appear either before or after the noun, although any given modifier is usually restricted to one position or the other.

(i) The red dog on the roof
(ii)*The on the roof dog
(iii)*The dog red

Our current Head-Modifier Rule only licenses post-head modifiers (like *on the roof* in (i)).

A. Write a second Head-Modifier Rule that licenses pre-head modifiers (e.g., *red* in (i)).
B. Modify the Head-Modifier 1 and Head-Modifier 2 Rules so that they are sensitive to which kind of modifier is present and don't generate (ii) or (iii). [*Hint: Use a feature* [*POST-HEAD* {+,−}] *to distinguish red and on the roof.*]
C. Is POST-HEAD a HEAD feature? Why or why not?
D. Give lexical entries for *red* and *on* that show the value of POST-HEAD. (You may omit the SEM features in these entries.)
E. Is (i) ambiguous according to your grammar (i.e. the Chapter 5 grammar modified to include the two Head-Modifier Rules, instead of just one)? Explain your answer.

This problem assumed that we don't want to make the two Head-Modifier Rules sensitive to the part of speech of the modifier. One reason for this is that modifiers of the same part of speech can occur before and after the head, even though individual modifiers might be restricted to one position or the other.

F. Provide three examples of English NPs with adjectives or APs after the noun.
G. Provide three examples of adverbs that can come before the verbs they modify.
H. Provide three examples of adverbs that can come after the verbs they modify.

Problem 2: Modes of Coordination
Consider the following data:

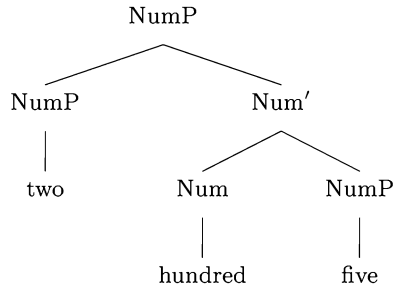
(i) Kim left and Sandy left.
(ii) ?*Kim left and did Sandy leave.
(iii) ?*Did Sandy leave and Kim left.
(iv) Did Sandy leave and did Kim leave?
(v) Go away and leave me alone!
(vi) ?*Kim left and leave me alone!
(vii) ?*Leave me alone and Kim left.
(viii) ?*Leave me alone and did Kim leave?
(ix) ?*Did Kim leave and leave me alone!

A. Formulate a generalization about the MODE value of conjuncts (and their mother) that could account for these data.
B. Modify the Coordination Rule in (42) so that it enforces the generalization you formulated in (A).

Problem 3: Semantics of Number Names
In Problem 5 of Chapter 3, we considered the syntax of English number names, and in particular how to find the head of a number name expression. Based on the results of that problem, the lexical entry for *hundred* in a number name like *two hundred five* should include the constraints in (i): (Here we are assuming a new subtype of *pos*, *number*, which is appropriate for number name words.)

(i)
$$\left\langle \text{hundred} , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{\begin{bmatrix} \text{HEAD} & \textit{number} \end{bmatrix}} \rangle \\ \text{COMPS} & \langle \boxed{\begin{bmatrix} \text{HEAD} & \textit{number} \end{bmatrix}} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle$$

This lexical entry interacts with our ordinary Head-Complement and Head-Specifier Rules to give us the phrase structure shown in (ii):

(ii)


Smith (1999) provides a compositional semantics of number names. The semantics of the top node in this small tree should be (iii):

(iii)
$$\begin{bmatrix} \text{INDEX} & i \\ \text{MODE} & \text{ref} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{constant} \\ \text{INST} & l \\ \text{VALUE} & 2 \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{times} \\ \text{RESULT} & k \\ \text{FACTOR1} & l \\ \text{FACTOR2} & m \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{constant} \\ \text{INST} & m \\ \text{VALUE} & 100 \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{RELN} & \textbf{plus} \\ \text{RESULT} & i \\ \text{TERM1} & j \\ \text{TERM2} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{constant} \\ \text{INST} & j \\ \text{VALUE} & 5 \end{bmatrix} \right\rangle \end{bmatrix}$$

This may seem long-winded, but it is really just a way of expressing "(two times one hundred) plus five" (i.e. 205) in our feature structure notation.

A. Assume that the two constant predications with the values 2 and 5 are contributed by the lexical entries for *two* and *five*. What predications must be on the RESTR list of the lexical entry for *hundred* in order to build (iii) as the SEM value of *two hundred five*?

B. The lexical entry for *hundred* will identify the indices of its specifier and complement with the value of some feature of a predication on its RESTR list. Which feature of which predication is the index of the specifier identified with? What about the index of the complement?

C. The lexical entry for *hundred* will identify its own INDEX with the value of some feature of some predication on its RESTR list. Which feature of which predication must this be, in order for the grammar to build (iii) as the SEM value of *two hundred five*?

D. Based on your answers in parts (A)–(C), give a lexical entry for *hundred* that includes the constraints in (i) and a fully specified SEM value. [*Note: Your lexical entry need only account for hundred as it is used in two hundred five. Don't worry about other valence possibilities, such as two hundred, two hundred and five, or a hundred.*]

E. The syntax and semantics of number names do not line up neatly: In the syntax, *hundred* forms a constituent with *five*, and *two* combines with *hundred five* to give a larger constituent. In the semantics, the constant predications with the values 2 and 100 are related via the times predication. The result of that is related to the constant predication with the value 5, via the plus predication Why is this mismatch not a problem for the grammar?