14

Long-Distance Dependencies

14.1 Introduction

One of the principal tasks of a theory of grammar is to provide mechanisms that allow economical formulations of the sorts of co-occurrence restrictions that exist in natural languages. In earlier chapters, we developed techniques for analyzing such aspects of syntax as differences in the valence of particular verbs, agreement between subject and verb, agreement between determiner and head noun, and restrictions on the distribution of dummy NPs. All of these co-occurrence restrictions are quite local, in the sense that they involve limitations on what can occur together as elements of a single clause. We extended this locality slightly with our analysis of raising, which in effect permits the co-occurrence restrictions of one verb to be transmitted to a higher verb.

The present chapter introduces a new type of construction in which the locality of co-occurrence restrictions appears to be violated in a more radical way. In these cases, two elements (say, an NP and a verb) appear far from one another in a sentence, despite the existence of a syntactic dependency (such as case marking or agreement) between them. Handling these 'long distance dependencies' (or LDDs, as we will call them) will require several changes to our theory:

- two new features,
- reformulation of the constraints on the types *word*, *lexeme* and *l-rule*, and on the initial symbol (in reference to the new features),
- a minor reformulation of some of our grammar rules,
- a new principle,
- a new grammar rule, and
- a new lexical rule.

14.2 Some Data

Our current grammar correctly rules out examples like the following:

(1) a.*They handed to the baby.
b.*They handed the toy.
c.*You have talked to.
d.*The children discover.

Because the lexical entry for *hand* specifies that its COMPS list has both an object NP and a PP, (1a–b) are ruled out through the interaction of the lexicon, the headed grammar rules, the Argument Realization Principle, and the Valence Principle. Similarly, (1c–d) are ruled out because both the preposition *to* and the verb *discover* require an object NP, which is absent from these examples.

So it's interesting to find that there are grammatical sentences that contain exactly the ungrammatical strings of words in (1). For example, there are questions containing *wh*-words ('*wh*-questions') such as following:

(2) a. What did they hand to the baby?
b. To whom did they hand the toy?
c. Who(m) should you have talked to?
d. What will the children discover?

There are also NPs modified by RELATIVE CLAUSES which contain the same ungrammatical strings:

(3) a. The toy which they handed to the baby...
b. The baby to whom they handed the toy ...
c. The people who(m) you have talked to...
d. The presents that the children discover...

Another sort of example is a kind of sentence that is used for a certain sort of emphasis that is usually called a 'topicalized' sentence. In such sentences, a topicalized element can be followed by one of those same ungrammatical word sequences in (1):[1]

(4) a. That toy, they handed to the baby.
b. To the baby, they handed a toy.
c. That kind of person, you have talked to (many times).
d. Presents that come from grandma, the children (always) discover.

And finally, there are certain adjectives like *easy* and *hard* whose infinitival complements may contain a verb or preposition lacking a normally obligatory object:

(5) a. That toy would be easy to hand to the baby.
b. You are easy to talk to.
c. The presents from grandma were hard for the children to discover.

In each of the examples in (2)–(5), there is a dependency between a phrase or 'filler' at the beginning of a clause and a 'gap' somewhere within the clause. In questions, relative

[1]When examples like (4) are first presented, some students claim that they find them unacceptable, but examination of actual usage indicates that topicalization is quite common, e.g. in examples like the following:

(i) Me, you bring an empty food dish; him, you bring a leash. (from a cartoon)
(ii) The film clips you're going to see tonight, no one's ever seen before. (Carol Burnett radio ad, November 26, 2001)

The name 'topicalization' is actually rather misleading. To be sure, the fronted element refers to an entity whose role in the discourse is distinguished in some way, but that entity need not correspond to the 'topic of discussion' in any straightforward way, as (i) indicates.

clauses, and topicalized sentences, the filler appears to be an extra phrase in that position; in examples like (5), the subject of the clause also serves as the filler.

In short, we see that elements whose presence is usually required in a clause are allowed to be absent if there is an appropriate filler in the right place. Likewise, if there is a filler, then there must be a gap somewhere within the sentence that follows the filler:

(6) a.*What did Kim hand the toys to the baby?
b.*The dolls that Kim handed the toys to the baby....
c.*The dolls, Kim handed the toys to the baby.
d.*The dolls are easy to hand the toys to the baby.

In such constructions, the filler can be separated from the gap by extra clauses, as indicated in (7)–(10). To help readers identify the location of the gaps, we have marked them with an underlined space.

(7) a. What did you say they handed __ to the baby?
b. Who(m) did he claim that they handed the toy to __ ?
c. Who(m) do you think you have talked to __ ?
d. What will he predict that the children discover __ ?

(8) a. The toy which we believe they handed __ to the baby...
b. The baby that I think they handed the toy to __ ...
c. The person who(m) everyone thinks you have talked to __ ...
d. The presents that it annoys me that the children discover __ ...

(9) a. That toy, I think they handed __ to the baby.
b. This baby, I know that they handed a toy to __ .
c. That kind of person, you know you have talked to __ .
d. Presents that come from grandma, I know that the children (always) discover __ .

(10) a. This toy isn't easy to try to hand __ to the baby.
b. The baby is easy to ask someone to hand a toy to __ .
c. That kind of person is hard to find anyone to talk to __ .
d. Presents from grandma are easy to help the children to discover __ .

In fact, there can be multiple extra clauses intervening:

(11) What did you think Pat claimed I said they handed __ to the baby?

14.3 Formulating the Problem

We want to be able to build clauses with elements missing within them. But somehow we have to keep track of the fact that something is missing. Furthermore, as the following contrasts show, we need to keep track of just what is missing:

(12) a. This, you can rely on.
b.*This, you can rely.
c.*On this, you can rely on.
d. On this, you can rely.
e.*On this, you can trust.

(13) a. Him, you can rely on.
b.*He, you can rely on.

(14) a. The twins, I can't tell the difference between.
b.*That couple, I can't tell the difference between.

Exercise 1: Long-Distance Selectional Dependencies
What exactly is wrong with the starred examples in (12)–(14)? Which element is selecting for the missing (or 'gapped') element, and which requirement of the selecting head does the filler not fulfill?

We can think of this as an information problem. We have to make sure that the phrases within the sentence keep track of what's missing from them as they are built. This has to be done just right, so that sentences missing a phrase of category *X* (no matter how deeply embedded that gap may be) combine with a filler of category *X*, and that fillers are allowed only when there is a gap for them to fill (cf. (6)).

14.4 Formulating a Solution

Our solution to this information problem will involve breaking it down into three parts: the bottom, the middle and the top. The bottom of an LDD is where the gap is 'introduced' – i.e. the smallest subtree where something is missing. Many theories handle the bottom by positing an empty element in the tree. We will avoid using empty elements in this way and instead handle the bottom by means of a feature (GAP) and a revision to the ARP that allows ARG-ST elements to show up on GAP instead of on the COMPS list. This is the topic of Section 14.4.1. The middle of an LDD is the 'transmission' of the information about what is missing from bottom to top (alternatively, the 'transmission' of what is available as a filler from top to bottom). We will handle this by means of a principle that relates the GAP values of phrases to the GAP values of their daughters. This is the topic of Section 14.4.2. The top of an LDD is where the filler is introduced, and the GAP requirement cancelled off. How exactly this happens depends on the particular kind of LDD. In Section 14.4.3, we will consider two kinds: 'topicalized' sentences, which we analyze in terms of a new phrase structure rule, and LDDs with *easy*-class adjectives, where the lexical entry for the adjective handles the top of the LDD.

14.4.1 The Feature GAP

We introduce the feature GAP (on *syn-cat*) to encode the fact that a phrase is missing a certain kind of element. There are examples of clauses where more than one phrase is missing,[2] a phenomenon we will return to in Problem 5 below:

(15) a. Problems this involved, my friends on the East Coast are hard to talk to __ about __ .
b. Violins this well crafted, these sonatas are easy to play __ on __ .

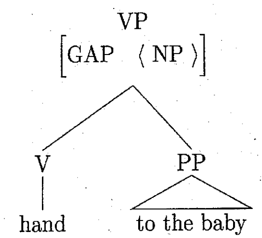
[2]Or, as linguists sometimes say (though it is somewhat of an oxymoron): 'where more than one gap appears'.

Note that the two gaps in each of these sentences have distinct fillers. In (15a), for example, the filler for the first gap is *my friends on the East Coast*, and the filler for the second one is *problems this involved*. Such examples are rare in English and sound a bit awkward, but there are other languages (for example several Slavic and Scandinavian languages) that allow multiple gaps more freely.

Given the existence of sentences with multiple gaps, we need a mechanism that can keep track of multiple missing elements. This suggests that the value of GAP is a list of feature structures, like the values of COMPS, SPR, MOD, and ARG-ST.

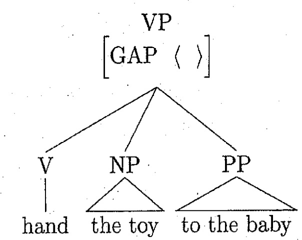
The intuitive significance of a phrase specified as, say, [GAP ⟨ NP ⟩] is that it is missing exactly one NP. The trick will be to make GAP have the right values in the right places. What we want is to allow a transitive verb or preposition to build a VP or PP without ever combining with an object NP. Furthermore, we want to ensure that it is only when an NP is absent that the relevant phrase is specified as [GAP ⟨ NP ⟩], as illustrated in (16):

(16)



When nothing is missing, we want the relevant phrase to be [GAP ⟨ ⟩], as in (17):

(17)



We will deal with this kind of 'missing element' as an instance of something that is present in argument structure but absent from the valence features. We could accomplish this by means of a lexical rule, but a more general solution is to modify the Argument Realization Principle. Our current version of the principle says that a word's SPR and COMPS lists add up to be its argument structure (ARG-ST) list. We now want to allow for the possibility that some element or elements of ARG-ST are on neither the SPR list nor the COMPS list, but on the GAP list instead.

To make this modification precise, we will introduce a kind of subtraction operation on lists, which we will mark with the symbol $\ominus$. Intuitively, if A and B are lists, then A $\ominus$ B is a list that results from removing the elements of B from A. A couple of caveats are in order here. First, we want A $\ominus$ B to be defined only when the elements of B all occur in A, and in the same order. So there are many pairs of lists for which this kind of list subtraction is undefined. This is unlike our form of list addition ($\oplus$), which is defined for any pair of lists. Second, when $A \ominus B$ is defined, it need not be unique. For example, if $A = \langle \text{NP, PP, NP} \rangle$ and $B = \langle \text{NP} \rangle$, then there are two possible values for $A \ominus B$, namely $\langle \text{NP, PP} \rangle$ and $\langle \text{PP, NP} \rangle$. We will interpret an equation like $A \ominus B = C$ to mean that there is some value for $A \ominus B$ that is identical to C.

With this new tool in hand, we can restate the Argument Realization Principle as follows:

(18) Argument Realization Principle:

$$\textit{word}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{A} \\ \text{COMPS} & \boxed{B} \ominus \boxed{C} \end{bmatrix} \\ \text{GAP} & \boxed{C} \end{bmatrix} \\ \text{ARG-ST} & \boxed{A} \oplus \boxed{B} \end{bmatrix}$$

The revised ARP guarantees that any argument that could appear on a word's COMPS list can appear on its GAP list instead. (We will deal with gaps that correspond to subjects, rather than complements, in Section 14.5) Further, (18) guarantees that whenever an argument is missing, any co-occurrence restrictions the word imposes on that argument will be registered on the element that appears on the GAP list.

Because the result of list subtraction ($\ominus$), as we have defined it, is not always unique, when we specify the ARG-ST in a verb's lexical entry without also specifying its SPR, COMPS, and GAP values, we are actually providing an underspecified lexical entry that will give rise to a family of words that differ with respect to how the ARP is satisfied. Consider, for example, the lexical entry for the lexeme *hand*, as specified in (19):

(19)

$$\left\langle \text{hand}, \begin{bmatrix} \textit{ptv-lxm} & \\ \text{ARG-ST} & \left\langle \text{X}_i, \text{Y}_k, \begin{bmatrix} \text{FORM} & \text{to} \\ \text{INDEX} & j \end{bmatrix} \right\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & \text{s} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{hand} \\ \text{SIT} & s \\ \text{HANDER} & i \\ \text{HANDED-TO} & j \\ \text{HANDED-ITEM} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

This can undergo the Non-3rd-Singular Verb Lexical Rule presented in Chapter 8, which gives rise to lexical sequences which satisfy the following description:

(20)

$$\left\langle \text{hand}, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-3sing} \end{bmatrix}, \overset{\text{NP}}{\begin{bmatrix} \text{CASE acc} \end{bmatrix}}, \begin{bmatrix} \text{FORM} & \text{to} \end{bmatrix} \right\rangle \end{bmatrix} \right\rangle$$

Since the second member of these lexical sequences is of type *word*, it is subject to the ARP. But now there are multiple ways to satisfy the ARP. In particular, the family of lexical sequences described in (20) includes lexical sequences meeting each of the following (more detailed) descriptions:

(21)

$$\left\langle \text{hand}, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2}\text{NP[acc]}, \boxed{3}\text{PP[to]} \rangle \end{bmatrix} \\ \text{GAP} & \langle\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\overset{\text{NP}}{\begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-3sing} \end{bmatrix}}, \boxed{2}, \boxed{3} \right\rangle \end{bmatrix} \right\rangle$$

(22)

$$\left\langle \text{hand}, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{3}\text{PP[to]} \rangle \end{bmatrix} \\ \text{GAP} & \langle \boxed{2}\text{NP[acc]} \rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\overset{\text{NP}}{\begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-3sing} \end{bmatrix}}, \boxed{2}, \boxed{3} \right\rangle \end{bmatrix} \right\rangle$$

(23)

$$\left\langle \text{hand}, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2}\text{NP[acc]} \rangle \end{bmatrix} \\ \text{GAP} & \langle \boxed{3}\text{PP[to]} \rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\overset{\text{NP}}{\begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-3sing} \end{bmatrix}}, \boxed{2}, \boxed{3} \right\rangle \end{bmatrix} \right\rangle$$

All of these are legitimate lexical sequences: (21) shows *hand*'s feature structure in sentences like (24a); (22) is the way *hand* appears in the tree our grammar assigns to sentences like (24b); and (23) shows *hand* as it appears in the tree we assign to sentences like (24c):[3]

(24) a. You handed the toy to the baby.
b. What did you hand to the baby?
c. To whom did you hand the toy?

The prepositional lexeme in (25) will now give rise to the word structures sketched in (26) and (27) (omitting what is not directly relevant):

(25)
$$\left\langle \text{to}\ ,\ \begin{bmatrix} \textit{argmkp-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{prep} \\ \text{FORM} & \text{to} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\text{NP[acc]} \right\rangle \end{bmatrix} \right\rangle$$

(26)
$$\begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{prep} \\ \text{FORM} & \text{to} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle \boxed{2}\text{NP[acc]} \rangle \end{bmatrix} \\ \text{GAP} & \langle\ \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{2} \rangle \end{bmatrix}$$

to

[3]The ARP also allows for a family of lexical sequences in which both the NP and PP complements are in the GAP list, rather than the COMPS list. We will return to multiple-gap sentences in Problem 5 below.

(27)
$$\begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{prep} \\ \text{FORM} & \text{to} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle \boxed{2}\text{NP[acc]} \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{2} \rangle \end{bmatrix}$$

to

This last lexical tree is the one that allows for sentences like (28):

(28) Which baby did you hand the toy to?

14.4.2 The GAP Principle

The GAP feature tells us which of a word's arguments is missing. The Argument Realization Principle, as we have reformulated it, permits us to instantiate gaps freely (other than elements that must be on the SPR list). Now we need some way of passing the information in the GAP value up[4] from words like those just illustrated so that the phrases that they head will register the fact that something is missing, and from those phrases to larger phrases. To do so, we adopt the principle shown in (29):

(29) The GAP Principle (Preliminary Version)

A local subtree Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{bmatrix} \text{GAP} & \boxed{A_1} \oplus \ldots \oplus \boxed{A_n} \end{bmatrix}$$

$$[\ \text{GAP}\ \boxed{A_1}\ ]\quad \ldots \quad [\ \text{GAP}\ \boxed{A_n}\ ]$$

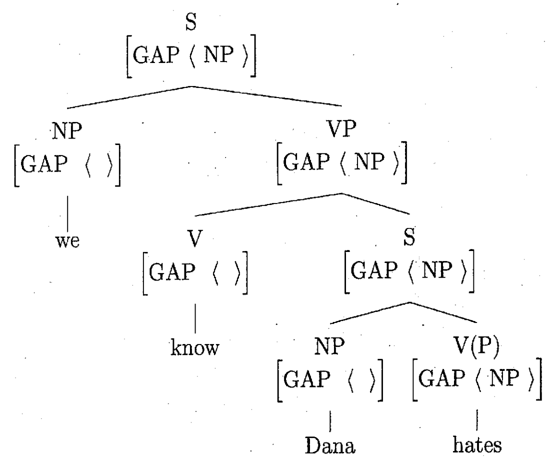
In other words, in a headed structure, the GAP values of all the daughters must add up to be the GAP value of the mother. That is, a phrase whose daughter is missing something is missing the exact same thing. There is one exception to this generalization, and that is the case where the larger phrase also contains the filler. We'll return to these cases directly.

The notion of lists 'adding up to' something is the same one we have employed before, namely the operation that we denote with the symbol '$\oplus$'. In most cases, most of the

[4]The metaphor of passing information between nodes should again not be taken literally. What the principle in (29) does is similar to what the Head Feature Principle and Valence Principle do, namely, enforce a particular relationship between certain feature values in mothers and daughters in phrase structure trees. That is, it is simply part of our definition of phrase-structure well-formedness.

GAP lists that are added up in this way are in fact empty, because most constituents don't contain gaps, so the addition is quite trivial. The effect of (29), then, given our lexical entries (and the word structures they sanction in virtue of our revision of the ARP), is illustrated in (30):

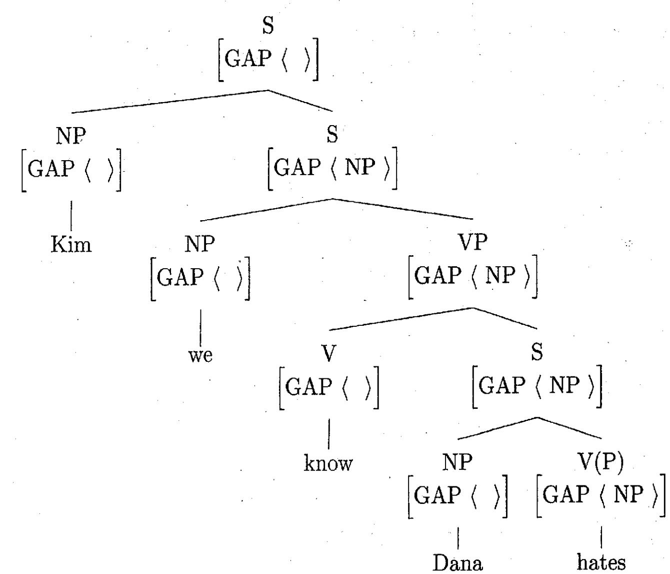
(30)



Note that each local tree in (30) satisfies the GAP Principle. That is, in each tree, the GAP values of the daughters add up to the mother's GAP value: $(\langle\ \rangle \oplus \langle \mathrm{NP} \rangle) = \langle \mathrm{NP} \rangle$

We now return to the exception (mentioned above) to the GAP Principle, as stated in the preliminary version: At the top of the LDD, where the gap is filled, we want the mother node to be [GAP ⟨ ⟩]. This is illustrated in (31):

(31)



We have not yet seen the phrase structure rule which licenses the topmost subtree of (31). It will be introduced in the next subsection. Here, we are concerned with the GAP values in that subtree. We want the mother to be [GAP ⟨ ⟩] as shown, because, intuitively, the NP *Kim* is 'filling' the gap. That is, the tree structure shown in (31) is no longer 'missing something', and this should be reflected in the GAP value of the root node in (31).

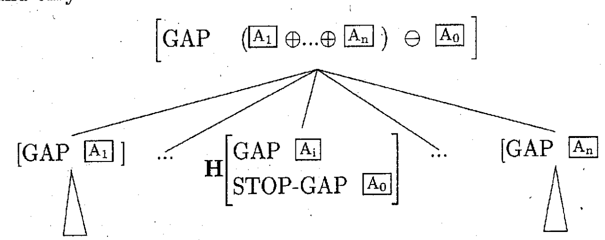
Adjectives like *hard* and *easy*, which we discussed earlier, also perform a gap-filling function, even though they also serve as the head daughter of a head-complement phrase. The VP in (32a) is 'gappy' – it is missing an NP and hence should be specified as [GAP ⟨ NP ⟩], while the AP in (32b) is not gappy and should be specified as [GAP ⟨ ⟩], like all other APs that we have encountered.

(32) a. [to talk to __]
b. [easy to talk to __]

We will provide a unified account of gap filling by introducing a new list-valued feature called STOP-GAP. Like GAP, STOP-GAP is a feature of *syn-cats*. This feature signals what gap is to be filled in the local subtree where it appears. Most nodes will be [STOP-GAP ⟨ ⟩], but where a gap is associated with its filler, the feature has a non-empty list as its value. In particular, the lexical entries for gap stoppers like *easy* and *hard* will specify a non-empty value for this feature, as will the grammar rule we introduce for the topicalization construction. Making use of this new feature, we can reformulate the GAP Principle so that it passes up GAP values only if they are not filled. This is shown in (33):

(33) The GAP Principle (Final Version)

A local subtree Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:



What this revision says is that the GAP value of the mother node in a headed structure is determined by adding up the GAP values of all the daughters and then subtracting any gaps that are being filled, as indicated by the head daughter's STOP-GAP value.

14.4.3 The Head-Filler Rule and *Easy*-Adjectives

We have dealt with the bottom of LDDs, where non-empty values for GAP are introduced, and the middle of LDDs where those GAP values are propagated through the tree (until they meet their fillers). Now we turn to the top of LDDs: the filling of the gap. As noted above, we will consider two types of gap-filling here: topicalized sentences and *easy*-adjectives.

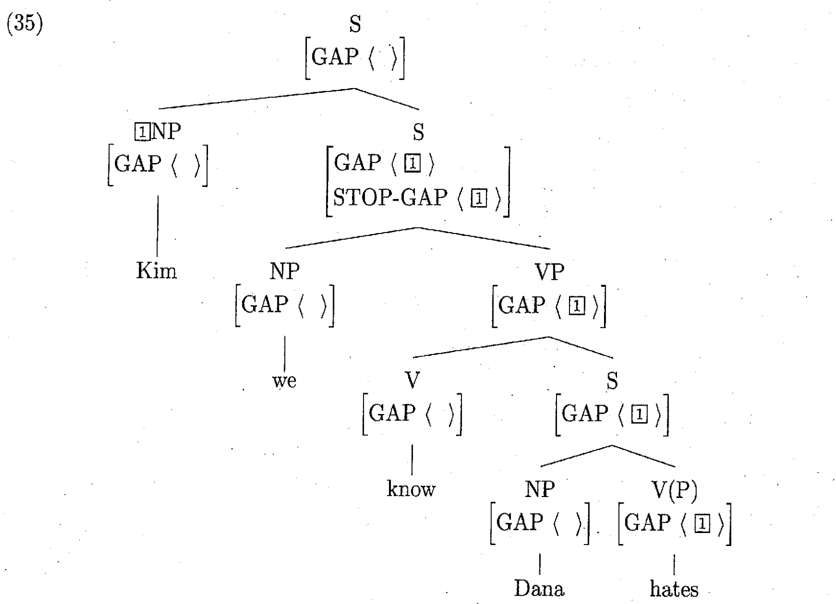
To deal with topicalized sentences, we now introduce a new grammar rule, formulated as follows:

(34) Head-Filler Rule

$$[phrase] \rightarrow \boxed{1}\begin{bmatrix}\text{GAP} & \langle\ \rangle\end{bmatrix} \quad \mathbf{H}\begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb}\\ \text{FORM} & \text{fin}\end{bmatrix}\\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle\\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\\ \text{STOP-GAP} & \langle \boxed{1} \rangle\\ \text{GAP} & \langle \boxed{1} \rangle\end{bmatrix}$$

This rule says that a phrase can consist of a head with a gap preceded by an expression that meets whatever requirements the head places on that gap.[5] The Head-Filler Rule licenses the topmost subtree in (35), and it enforces the identity between the NP *Kim* and the element on the GAP list of the gappy S *we know Dana hates* ($\boxed{1}$). Because that GAP element is identified with the GAP element of the V *hates* (and therefore also with an element of its ARG-ST list), any requirements that *hates* places on its complement (that it be a [CASE acc] NP, that its INDEX be identified with the HATED value in the **hate** predication) must be satisfied by the filler *Kim*.

(35)



The topmost node of (35) is [GAP ⟨ ⟩], indicating that the gap has been filled, thanks to the GAP Principle: The Head-Filler Rule in (35) specifies that the head daughter's GAP list and STOP-GAP list both contain the filler daughter, so that element is subtracted from the head daughter's GAP value in determining the GAP value of the mother:

$(\langle\ \rangle \oplus \langle \boxed{1} \rangle) \ominus \langle \boxed{1} \rangle = \langle\ \rangle$.

[5] And further that the filler must not be gappy.

It is important to see that our analysis entails that a filler NP can appear before a clause only when that clause is gappy, i.e. when that clause is missing an NP that would normally appear there. Moreover, the Head-Filler Rule does not require the filler to be an NP, but it does require that the filler's *synsem* be identified with the unique member of the head daughter's GAP list. From this it follows that topicalized sentences may contain PP fillers (and perhaps fillers of other categories) just as long as the gap within the clause matches the synsem of the filler. That is, if the filler is a PP, then the missing element must be a PP, not an NP. This is a consequence of the many identities triggered by the Head-Filler Rule and the GAP Principle, interacting with the Argument Realization Principle and particular lexically specified ARG-ST values.

We now turn to our other example of gap filling, adjectives like *easy* or *hard*. Most words don't fill gaps, so we will posit the following defeasible constraint on the type *lexeme*:

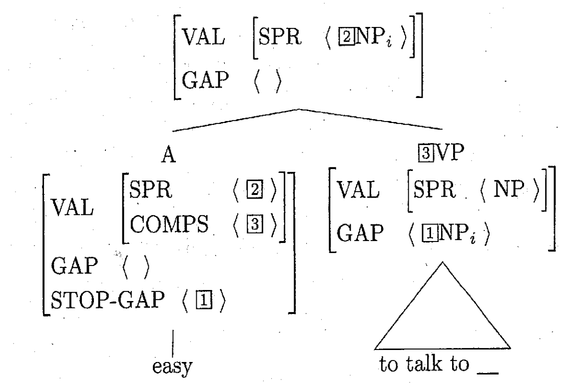
(36) $\textit{lexeme}: \begin{bmatrix}\text{STOP-GAP} & / \langle\ \rangle\end{bmatrix}$

Adjectives like *easy* or *hard* are the exceptions. We give them lexical entries which override this constraint, as shown for *easy* in (37):

(37)
$$\left\langle \text{easy}, \begin{bmatrix}\textit{adj-lxm}\\ \text{SYN} & \begin{bmatrix}\text{STOP-GAP} & \langle \boxed{1} \rangle\end{bmatrix}\\ \text{ARG-ST} & \left\langle \text{NP}_i\, , \overset{\text{VP}}{\begin{bmatrix}\text{INF} & +\\ \text{GAP} & \langle \boxed{1}\text{NP}_i\, , \ldots \rangle\end{bmatrix}} \right\rangle\end{bmatrix} \right\rangle$$

Because the member of the STOP-GAP list in (37) is identified with the first member of the VP argument's GAP list, adjectives of this type must perform gap stopping of the sort shown in (38):

(38)



Notice that the GAP list is empty at the top node of this subtree. That is, the AP *easy to talk to* is treated as having no gap, even though the infinitival VP *to talk to* inside it has an NP gap. This may seem puzzling, since *easy to talk to* seems to be missing the same NP as *to talk to*. But at the level of the AP, the referent of the missing NP is fully determined: it is the same as the subject of the AP. Hence, the GAP list at the AP level no longer needs to register the missing NP. Instead, the first argument (that is, the subject) of the AP is coindexed with the NP in the GAP list.[6] This guarantees that, in a sentence like (39), the Pat is understood as the person who is followed:

(39) Pat is easy to continue to follow __ .

14.4.4 GAP and STOP-GAP in the Rest of the Grammar

We have added two features to our grammar (GAP and STOP-GAP) which are involved in passing information around the tree. As such, we must pause and ask whether the rest of our grammar (in particular, lexical rules, the rest of our grammar rules and the initial symbol) are currently doing the right thing with respect to these new features. The answer is (unsurprisingly) that we will need to make a few modifications.

First, with respect to the feature GAP: Nothing we have said so far ensures that all gaps ultimately get filled. We make sure that SPR and COMPS requirements are ultimately fulfilled by requiring that both be empty on the initial symbol. We can do the same for GAP. That is, our initial symbol is now the following:

(40)
$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

Without this specification, we would license examples like (1), repeated here for convenience, as stand-alone utterances:

(41) a.*They handed to the baby.
b.*They handed the toy.
c.*You have talked to.
d.*The children discover.

The other consideration with respect to the feature GAP is whether its value is sufficiently constrained. The GAP Principle applies to headed phrases, but not non-headed phrases. Thus, in our discussion so far, we have not constrained the GAP value of coordinate phrases or imperatives. We will return to coordination in Section 14.6 below. As for imperatives, in order to ensure that we don't allow gappy VPs as the daughter (as in (42)), we can identify the mother's and daughter's GAP values, as shown in (43). Since imperative phrases must also satisfy the initial symbol, they must be [GAP ⟨ ⟩] on the mother.

(42)*Hand the toy!

[6] More precisely, with the NP in initial position in the GAP list.

(43) Imperative Rule (Revised Version)

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \boxed{A} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{dir} \\ \text{INDEX} & s \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{NP[PER 2nd]} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \boxed{A} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \end{bmatrix} \end{bmatrix}$$

Thanks to the GAP Principle and the two modifications given above, GAP values are now sufficiently constrained throughout our grammar. We haven't said much about STOP-GAP values, however, except to say that they are non-empty in two places: on the head daughter of a head-filler phrase, and in the lexical entries for adjectives like (*easy*). In addition, the defeasible constraint given in (36) above and repeated here ensures that all other lexical entries are [STOP-GAP ⟨ ⟩]:

(44) $\textit{lexeme}: \begin{bmatrix} \text{STOP-GAP} \ / \ \langle\ \rangle \end{bmatrix}$

Since we want the STOP-GAP values given on lexemes to be reflected in the word-structures they license, we need to make sure that all lexical rules preserve that information. We do that by adding the following non-defeasible constraint to the type *l-rule*:

$$\textit{l-rule}: \begin{bmatrix} \text{INPUT} & \left\langle \text{X}\ ,\ [\text{STOP-GAP}\ \boxed{A}] \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ ,\ [\text{STOP-GAP}\ \boxed{A}] \right\rangle \end{bmatrix}$$

When STOP-GAP is non-empty, the GAP Principle subtracts the relevant element from the GAP list being passed 'up' the tree. It follows that we want to ensure that STOP-GAP is empty when there is no gap-filling going on. Gaps are never filled in head-specifier or head-modifier phrases, so we constrain the head daughters of the Head-Specifier and Head-Modifier Rules to be [STOP-GAP ⟨ ⟩]:

(45) Head-Specifier Rule (Revised Version)

$$\begin{bmatrix} \textit{phrase} & \\ \text{SPR} & \langle\ \rangle \end{bmatrix} \rightarrow \boxed{1}\ \ \mathbf{H}\begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{STOP-GAP} & \langle\ \rangle \end{bmatrix}$$

(46) Head-Modifier Rule (Revised Version)

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

Gap-filling sometimes occurs in head-complement phrases (in particular, when the head is an adjective like *easy*), so we do not want to constrain the head daughter of the Head-Complement Rule to be [STOP-GAP ⟨ ⟩]. However, since the head daughter of this rule is always a *word*, the STOP-GAP value will be appropriately constrained by the lexical entries.

This completes our discussion of complement gaps.[7]

14.5 Subject Gaps

We have covered only the basic cases of long-distance dependencies. There are many additional complexities. For example, we have not discussed cases in which the gaps are not complements, but rather subjects or modifiers. In addition, we have not discussed the distribution of *wh*-words (such as *who, what, which,* etc.) in questions and relative clauses, nor the obligatory inverted order of subject and auxiliary verb in many *wh*-questions. There is a rich literature investigating these and many other questions associated with LDDs, but such matters are beyond the scope of this text. In this section we sketch the basics of an account of what is subject extraction – that is LDDs in which the gaps are in subject position.

Our present account does not yet deal with examples like (47):

(47) a. Which candidates do you think like oysters on the half-shell?
  b. That candidate, I think likes oysters on the half-shell.

This is because of an interaction between the ARP and the constraints (including the SHAC, inherited from *infl-lxm*) that all verb lexemes have SPR lists of length one. Together, these constraints require that the first member of a verb's ARG-ST list must appear on its SPR list. It may not belong to the rest of the list – i.e. to the list of elements that can appear on either COMPS or GAP, according to the ARP.

Rather than attempt to revise the ARP to handle these cases, we will treat them in terms of a post-inflectional lexical rule which provides [SPR ⟨ ⟩] lexical sequences for verbs, and puts the right information into the GAP list:

(48) Subject Extraction Lexical Rule

$$\begin{bmatrix} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \text{X}, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & [\text{SPR} \ \ \langle \text{Z} \rangle] \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & [\text{SPR} \ \ \langle \ \rangle] \\ \text{GAP} & \langle \boxed{1} \rangle \end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{A}} \langle \boxed{1}, \ldots \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

This rule maps any finite verb form into a word with an empty SPR list and a GAP list containing an element identified with the first argument – the subject of the verb. The

[7] There are further constraints governing complement gaps that we will not treat here. For example, an ADV_{pol} like *not* or accented so, which were analyzed as complements in Chapter 13, cannot serve as a topicalization filler:

(i) *Not, Kim will go to the store.
(ii) *So, Kim will go to the store.

This contrasts with the behavior of adverbial modifiers (left untreated in this text), which may be topicalized:

(iii) Tomorrow, (I think) Kim will go to the store __ .

lexical sequences that are the outputs of this rule are illustrated by the description in (49):

(49)

$$\left\langle \text{likes}, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \ \rangle \\ \text{COMPS} & \langle \boxed{2} \rangle \end{bmatrix} \\ \text{GAP} & \left\langle \boxed{1} \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \right\rangle \\ \text{STOP-GAP} & \langle \ \rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1}, \boxed{2}\text{NP[acc]} \rangle \end{bmatrix} \right\rangle$$

Note that the ARP (inherited from the type *word*) is satisfied in (49): the SPR list is empty, and the rest of the ARG-ST list (i.e. the whole ARG-ST list) is appropriately related to the list values of COMPS and GAP. That is, the COMPS value (⟨NP[acc]⟩) is just the ARG-ST value (50a) minus the GAP value (50b):

(50) a. $\left\langle \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{3sing} \end{bmatrix}, \begin{matrix} \text{NP} \\ [\text{CASE acc}] \end{matrix} \right\rangle$

  b. $\left\langle \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{3sing} \end{bmatrix} \right\rangle$

14.6 The Coordinate Structure Constraint

One of the most discussed topics related to LDDs concerns restrictions on possible filler/gap associations. Although the position of filler and gap may be arbitrarily far apart, there are certain configurations that do not permit LDDs. Such configurations are known as 'islands' (a term due to Ross (1967)), and a major goal of syntactic research over the past three decades has been to understand where and why islands occur. In this section, we will look at one type of island and show how our grammar correctly predicts its existence and its properties.

The following examples illustrate what Ross called the 'Coordinate Structure Constraint':

(51) a.*Here is the student that [the principal suspended [__ and Sandy]].
  b.*Here is the student that [the principal suspended [Sandy and __]].

(52) a.*Here is the student that [[the principal suspended __] and [the student council passed new rules]].
  b.*Here is the student that [[the student council passed new rules] and [the principal suspended __]].

(53) a.*Apple bagels, I can assure you that [[Leslie likes __] and [Sandy hates lox]].
  b.*Apple bagels, I can assure you that [[Leslie likes lox] and [Sandy hates __]].

Translating Ross's transformation-based formulation of the constraint into the language of fillers and gaps that we have been using, it can be stated as follows:

(54) Coordinate Structure Constraint (CSC)
In a coordinate structure,
(a) no conjunct can be a gap,
(b) nor can a gap be contained in a conjunct if its filler is outside of that conjunct.

(54a) is often referred to as the CONJUNCT CONSTRAINT, while (54b) is sometimes called the ELEMENT CONSTRAINT.

Ross also noticed a systematic class of exceptions to the Element Constraint, illustrated by (55):

(55) a. This is the dancer that we bought [[a portrait of __] and [two photos of __]].
b. Here is the student that [[the school suspended __] and [we defended __]].
c. Apple bagels, I can assure you that [[Leslie likes __] and [Sandy hates __]].

To handle examples like these, he appended an additional clause to the constraint, which we can formulate as follows:

(56) 'Across-the-Board' Exception (addendum to CSC):
...unless each conjunct properly contains a gap paired with the same filler.

As presented, the Coordinate Structure Constraint seems quite arbitrary, and the Across-the-Board Exception is just an added complication. And most analyses of these phenomena – specifically those that handle LDDs transformationally – have never come to grips with the full range of facts, let alone derived them from general principles.

Note first of all that the Conjunct Constraint is already explained by our grammar. Examples like (51) are ungrammatical for the simple reason that the elements on GAP lists must also be on ARG-ST lists, and coordinate conjunctions like *and* have empty ARG-ST lists. Unlike many other analyses (in particular, transformational approaches) our grammar does not employ empty elements (usually referred to as 'traces') to occupy the position of the gap in the syntactic structure. Since there are no empty NPs in our analysis, there is no empty element that could serve as a conjunct in a coordinate structure. That is, the Conjunct Constraint follows directly from the decision to treat the bottoms of LDDs in terms of an unrealized argument, rather than the presence of an empty element.

Now reconsider the grammar rule for coordination last updated in Chapter 8:

(57) Coordination Rule (Chapter 8 Version)

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_0 \end{bmatrix} \rightarrow$$

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

As stated, this rule doesn't say anything about the GAP values of the conjuncts or of the mother. (Note that the GAP Principle doesn't apply to subtrees licensed by this rule, as it is not a headed rule.) In our discussions of coordination so far, we have seen that some features must be identified across conjuncts (and with the mother) in coordination and that others should not. The Element Constraint examples cited above in (52) and (53) show that GAP is one of the features that must be identified. We thus modify our Coordination Rule slightly to add this constraint:

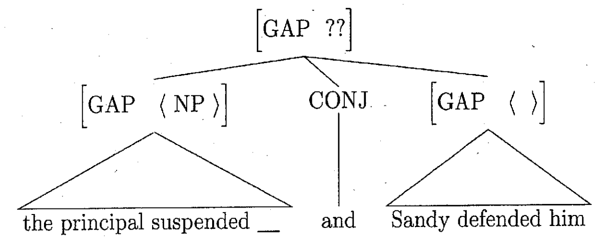
(58) Coordination Rule (Final Version)

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_0 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_1 \end{bmatrix} \ldots \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_{n-1} \end{bmatrix} \begin{bmatrix} \text{HEAD} & \textit{conj} \\ \text{IND} & s_0 \\ \text{RESTR} & \left\langle \left[\text{ARGS} \ \langle s_1 \ldots s_n \rangle \right] \right\rangle \end{bmatrix} \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_n \end{bmatrix}$$

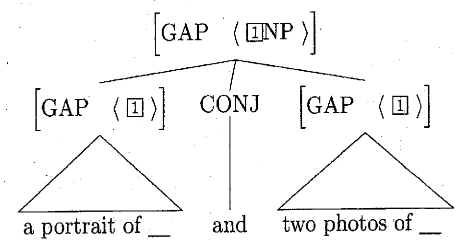
This revision guarantees that two conjuncts in a coordinate structure cannot differ in their GAP value. If one has an empty GAP list and the other has a nonempty GAP list (as in (51)–(53)), then the structure is not licensed. The GAP values that must be identical cannot be, as shown in (59):

(59)



On the other hand, it is possible for conjuncts to have nonempty GAP lists if they are all nonempty and all share the same value. This is what is illustrated in (55), whose structure is as shown in (60):

(60)



In short, both the Element Constraint and the Across-the-Board exceptions to it are treated properly in this revision of our analysis of coordination.

We close this discussion with one final observation about LDDs and coordinate structures. There is an exception to (56), illustrated by (61):

(61)*Which rock legend would it be ridiculous to compare [[__] and [__]]?

Our statements of the generalizations in (54) and (56), like Ross's original formulations of them, would in fact permit (61), whose deviance should have a syntactic (rather than a semantic) explanation, it would appear, because the meaning of this putative sentence could certainly be expressed as (62):

(62) Which rock legend would it be ridiculous to compare __ with himself?

But our analysis correctly rules out any sentences in which a gap constitutes a full conjunct. As noted above, this is because nonempty GAP values in the lexicon are licensed by the Argument Realization Principle, which allows ARG-ST elements not to be expressed as complements, rather than allowing them to appear as a phonetically empty element, or 'trace'. The difference is subtle, but the predictions are quite striking: our traceless analysis of gaps provides an immediate account of the deviance of (61) as well as an explanation of the examples in (51)–(53), which motivated Ross's Conjunct Constraint. The Coordinate Structure Constraint and its exceptions are thus properly accounted for in the analysis of coordination we have developed. Many alternative approaches – particularly those involving movement transformations to account for LDDs – have been unable to account for them at all.

14.7 Summary

Deducing the Conjunct Constraint from the interaction of our analyses of coordination and LDDs is an elegant result, providing significant support for our general approach to syntax. We also showed that we could extend our account of coordination in order to account for the Element Constraint as well.[8]

We will not examine other island constraints in this text. As with the Coordinate Structure Constraint, linguists have not been content to catalog the environments in which filler-gap pairings are impossible. Rather, a great deal of effort has gone into the search for explanations of syntactic islands, either in terms of the interaction of independently motivated elements of the theory (as in the example given above), or in terms of such factors as the architecture of the human language-processing mechanisms. This is a fertile area of research, in which definitive answers have not yet been found.

14.8 Changes to the Grammar

In this chapter, we developed an analysis of long-distance dependencies involving 'fillers' and unrealized elements, or 'gaps'. Our analysis involved two new features, GAP and STOP-GAP, both appropriate for feature structures of type *syn-cat*:

$$\textit{syn-cat}: \begin{bmatrix} \text{HEAD} & \textit{pos} \\ \text{VAL} & \textit{val-cat} \\ \text{GAP} & \textit{list(expression)} \\ \text{STOP-GAP} & \textit{list(expression)} \end{bmatrix}$$

[8] Essentially this account was first developed by Gazdar (1981), within the framework of Generalized Phrase Structure Grammar.

We treated the introduction of gaps at the bottom of LDDs in terms of the following modification of the Argument Realization Principle:

Argument Realization Principle:

$$\textit{word}: \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{A} \\ \text{COMPS} & \boxed{B} \ominus \boxed{C} \end{bmatrix} \\ \text{GAP} & \boxed{C} \end{bmatrix} \\ \text{ARG-ST} & \boxed{A} \oplus \boxed{B} \end{bmatrix}$$

To introduce subject gaps, we created the following lexical rule:

Subject Extraction Lexical Rule

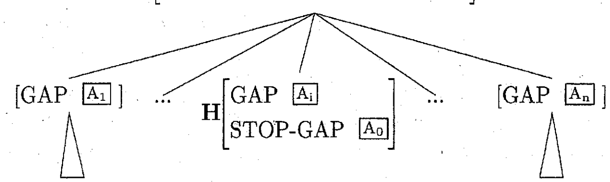
$$\begin{bmatrix} \textit{pi-rule} & \\ \text{INPUT} & \left\langle \text{X}\,, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{FORM} & \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{Z} \rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \boxed{A} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\,, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \end{bmatrix} \\ \text{GAP} & \langle \boxed{1} \rangle \end{bmatrix} \\ \text{ARG-ST} & \boxed{A}\langle \boxed{1}, \ldots \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

We also introduced a new principle, which has the effect of passing GAP specifications from daughter to mother within headed phrase structures, while subtracting out any GAP elements that are bound within the phrase:

The GAP Principle

A local subtree Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\begin{bmatrix} \text{GAP} & (\boxed{A_1} \oplus \ldots \oplus \boxed{A_n}) \ominus \boxed{A_0} \end{bmatrix}$$



The value of STOP-GAP is assigned in the lexicon by the following defeasible constraint that is overridden by the lexical entries for adjectives like *easy* and *hard*:

$$\textit{lexeme}: \begin{bmatrix}\text{STOP-GAP} & / \langle\ \rangle\end{bmatrix}$$

$$\left\langle \text{easy}\ , \begin{bmatrix} \textit{adj-lxm} & \\ \text{SYN} & \begin{bmatrix}\text{STOP-GAP} & \langle \boxed{1} \rangle\end{bmatrix} \\ \text{ARG-ST} & \left\langle \text{NP}_i\ , \overset{\text{VP}}{\begin{bmatrix} \text{INF} & + \\ \text{GAP} & \langle \boxed{1}\text{NP}_i\ , \ldots \rangle \end{bmatrix}}\right\rangle \end{bmatrix}\right\rangle$$

And we added the following nondefeasible constraint on the type *l-rule*:

$$\textit{l-rule}: \begin{bmatrix}\text{INPUT} & \left\langle \text{X}\ , \begin{bmatrix}\text{STOP-GAP} & \boxed{A}\end{bmatrix}\right\rangle \\ \text{OUTPUT} & \left\langle \text{Y}\ , \begin{bmatrix}\text{STOP-GAP} & \boxed{A}\end{bmatrix}\right\rangle\end{bmatrix}$$

To handle the top of LDDs, we introduced lexical entries for *easy*-type adjectives and we also introduced one new grammar rule – the Head-Filler Rule, which licenses sentences with a 'topicalized' initial element:

Head-Filler Rule

$$[\textit{phrase}] \rightarrow \boxed{1}\begin{bmatrix}\text{GAP} & \langle\ \rangle\end{bmatrix} \quad \mathbf{H}\begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{STOP-GAP} & \langle \boxed{1} \rangle \\ \text{GAP} & \langle \boxed{1} \rangle\end{bmatrix}$$

To properly constrain the values of GAP and STOP-GAP throughout our trees, we made minor revisions to the 'initial symbol' clause of the definition of well-formed tree structure and to the Head-Specifier, Head-Modifier and Imperative Rules:

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

...

the label of Φ's root node satisfies the constraint:

$$\begin{bmatrix}\textit{phrase} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \text{fin}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{GAP} & \langle\ \rangle\end{bmatrix}\end{bmatrix}$$

Head-Specifier Rule

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

Head-Modifier Rule

$$[\textit{phrase}] \rightarrow \mathbf{H}\boxed{1}\begin{bmatrix}\text{COMPS} & \langle\ \rangle \\ \text{STOP-GAP} & \langle\ \rangle\end{bmatrix} \quad \begin{bmatrix}\text{COMPS} & \langle\ \rangle \\ \text{MOD} & \langle \boxed{1} \rangle\end{bmatrix}$$

Imperative Rule

$$\begin{bmatrix}\textit{phrase} & \\ \text{SYN} & \begin{bmatrix}\text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\ \rangle\end{bmatrix} \\ \text{GAP} & \boxed{A}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{dir} \\ \text{INDEX} & s\end{bmatrix}\end{bmatrix} \rightarrow \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{INF} & - \\ \text{FORM} & \text{base}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle \text{NP}[\text{PER}\ \ \text{2nd}] \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{GAP} & \boxed{A}\end{bmatrix} \\ \text{SEM} & [\text{INDEX}\ \ s\,]\end{bmatrix}$$

Finally, to deal with the Element Constraint (part of Ross's Coordinate Structure Constraint), we modified the Coordination Rule as follows:

Coordination Rule

$$\begin{bmatrix}\text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_0\end{bmatrix} \rightarrow$$

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

14.9 Further Reading

Ross 1967 is probably the most influential work to date on the topic of long-distance dependencies. Chomsky (1973, 1977, 1986a) developed one of the most influential approaches to analyzing these constructions, using transformations. The treatment presented here is based loosely on that developed in Pollard and Sag 1994, which is compared with transformational approaches in Levine and Sag 2003. This analysis is unusual in not positing an empty category (a trace) in the position of the gap. Arguments for such a traceless analysis are discussed by Sag and Fodor (1994). Other nontransformational treatments are presented in Gazdar 1981, Kaplan and Zaenen 1989, Steedman 2000, and Bouma et al. 2001.

14.10 Problems

Problem 1: A Tree with a Gap
Draw a tree for (9b). Use abbreviations for the node labels, and show the value of GAP on all nodes. Show the value of STOP-GAP on any node where it is non-empty.

⚠ **Problem 2: Blocking Filled Gaps**
Examples (i) and (ii) are well-formed, but example (iii) is ungrammatical:

(i) Pat thinks that I rely on some sort of trick.
(ii) This mnemonic, Pat thinks that I rely on.
(iii) *This mnemonic, Pat thinks that I rely on some sort of trick.

Explain in detail why the mechanisms that license (i) and (ii) do not also permit (iii).

Problem 3: Subject Gaps
This problem is to make sure you understand how our analysis accounts for examples like (47).

A. Sketch the family of lexical sequences for *likes* that is the input to the Subject Extraction Lexical Rule.
B. Sketch the family of lexical sequences for *likes* that is the corresponding output to the Subject Extraction Lexical Rule.
C. Sketch the tree for the sentence in (47b). Use abbreviations for node labels, but show the value of GAP on all nodes and the value of STOP-GAP on any node where it is non-empty. You may abbreviate the structure over the NP *oysters on the half-shell* with a triangle.
D. Does our analysis correctly predict the contrast between (47b) and (i)?
(i) *Those candidates, I think likes oysters on the half-shell.
Explain why or why not.

Problem 4: Irish Complementizers
Consider the following example that shows the typical word order pattern of Modern Irish (data from McCloskey 1979):

(i) Shíl mé goN mbeadh sé ann.
thought I COMP would-be he there
'I thought that he would be there.'

Irish is a VSO language. One way of analyzing such languages is to posit a Head-Specifier-Complement Rule that introduces both kinds of dependents as sisters of the lexical head. In addition, we'll need a Head-Complement Rule that realizes only complements, and requires the head daughter to be [SPR ⟨ ⟩].

A. Formulate these two rules and show the structure for sentence (i). You may use abbreviations such as NP, S, etc., but be sure to show the value of COMPS and SPR on each node.

Now consider some further Irish data:

(ii) Dúirt mé gurL shíl mé goN mbeadh sé ann.
said I goN.PAST thought I COMP would-be he there
'I said that I thought that he would be there.'

(iii) an fear aL shíl mé aL bheadh ann
the man COMP thought I COMP would-be __ there
'the man that I thought would be there'

(iv) an fear aL dúirt mé aL shíl mé aL
the man COMP said I COMP thought I COMP
bheadh ann
would-be __ there
'the man that I said I thought would be there'

(v) an fear aL shíl goN mbeadh sé ann
[the man]$_j$ COMP thought __ COMP would-be he$_j$ there
'[the man]$_j$ that thought he$_j$ would be there'

(vi) an fear aL dúirt sé aL shíl goN
the man COMP said he COMP thought __ COMP
mbeadh sé ann
would-be he there
'the man that he said thought he would be there'

The complementizers *goN* and *aL* are in complementary distribution. That is, wherever *goN* is possible in these examples, *aL* is not, and vice versa.[9] Assume that both these elements are heads of CPs similar to those headed by *that* complementizers in English. If we then make the further assumption that LDDs in Irish work much as they do in English, we have all the tools we need to analyze the contrasts in (i)–(vi).[10]

B. Provide lexical entries for these two complementizers. [*Note: You may assume for purposes of this problem that the type comp-lxm which we proposed for English is applicable to Irish as well.*]
C. Show how your analysis successfully explains the distributional differences between the two complementizers. Be sure to cite the data given in the problem.

[9] For the purposes of this problem, you should ignore the difference between *gurL* and *goN*.
[10] Examples (iii)–(vi) involve relative clauses, which we have not discussed in much detail. Assume that the complementizers are the same whether they appear in relative clauses or in CP complements to verbs.

Problem 5: Nested Dependencies
We have made GAP a list-valued feature, which leaves open the possibility of multiple GAPs. This problem considers sentences which instantiate this possibility, such as (i) and (ii):

(i) Problems this involved, my friends on the East Coast are hard to talk to __ about __ .

(ii) Violins this well crafted, these sonatas are easy to play __ on __ .

A. Indicate which NP is interpreted as the filler for each of the gaps in (i) and (ii).

B. Draw a tree for sentence (i), indicating the value of GAP and STOP-GAP on every node. You do not need to include other features, though you should have a node label (e.g. VP, PP, etc.) on each node, and use tags and coindexing as appropriate. You may abbreviate the structure of the NPs *problems this involved* and *my friends on the East Coast* with triangles.

The PP complements of *talk* can actually appear in either order:

(iii) Dana tried to talk about it to everyone in the building.

(iv) Dana tried to talk to Leslie about this problem.

For the sake of this problem, we will assume that this is dealt with by allowing two distinct ARG-ST lists for *talk*: ⟨ NP , PP[about] , PP[to] ⟩ and ⟨ NP , PP[to] , PP[about] ⟩.

However, when we switch the order of the PPs in the multiple gap example, we get a sentence with a bizarre meaning (in which someone is talking to problems about people):

(v) Problems this involved, my friends on the East Coast are hard to talk about __ to __ .

C. Is this predicted by our analysis of LDDs? Why or why not?
[*Hint: Remember that the value of GAP is a list, and the order of the GAP list on phrasal nodes is determined by the GAP Principle.*]

Problem 6: Binding and LDDs
Assuming that reciprocals are [MODE ana], does our analysis of LDDs interact with the Binding Theory to predict that (i) should be grammatical? Why or why not?

(i) [Those people]$_i$ I tend to believe will tell [each other]$_i$ everything.

15

Variation in the English Auxiliary System

15.1 Introduction

English auxiliaries constitute a particularly interesting syntactic system, involving a small set of words that exhibit many intricate interactions and some fascinating idiosyncrasies. This system is peculiar to English, though many other languages have elements with intriguingly parallel properties.

English auxiliaries have changed considerably over the last thousand years or so, and their evolution has been well documented, making them a natural domain for studying syntactic change.[1] Change begins with variation, and the auxiliary system is also the locus of some fascinating differences among varieties of English. Variation is interesting in its own right, but studying it also helps us to ascertain which properties of our grammar we should formulate as or deduce from general principles, and which ones we should treat as essentially accidental.

In this chapter, we provide two examples of variation in English auxiliaries. The first example concerns the behavior of *have*, whose syntax is that of an auxiliary verb in some instances but not in others. The second example deals with a much studied phenomenon in the dialect known as African American Vernacular English (AAVE, for short). In both cases, we will explore how the variation might be handled within our theory. Our examples and discussion are intended only as samples of how data on variation might be relevant to work on syntactic theory. Syntactic variation is a topic worthy of a textbook in its own right, and we make no pretense of doing it justice here.

15.2 Auxiliary Behavior in the Main Verb *Have*

Our first example of variation in the English auxiliary system is one that occurs in a number of varieties of English, and can even vary for individual speakers, depending on the formality of the context: In some circumstances, certain uses of *have* as a main verb exhibit auxiliary-like behavior. The examples in (1) were judged acceptable by the speakers of British English we consulted: (Speakers of other varieties may well have heard such examples, even if they wouldn't produce them.)[2]

[1] Discussion of the history of the English auxiliary system, though an exciting topic, would take us too far afield.

[2] Our small survey on these data also included some speakers from Australia, New Zealand and India. These sentences were most uniformly accepted by the British speakers, however, and it is their judgments