Chapter 12

Categorical (‘Syntactic’) Structures

In this chapter the focus is on the categorical properties of phrases and sentences. We will see that, even though sentences generally have a greater complexity, the principles that govern the categorical structure of complex words and of sentences are quite similar.

Introduction

There are restrictions on how one can group words together and on the ordering of words into phrases and sentences. As before, we have three systems, each imposing its own set of constraints: the categorical system, the phonotactic system, and the semantic system. We briefly looked at the sentence-level phonotactic system in Chapter 10. We will even more briefly look at the sentence-level semantic system in Chapter 13. In this chapter, we will focus our attention on the categorical side of sentences. (Recall that most linguists call this syntax, even though syntax in a broad sense refers to the merging of words with all their properties, not just the categorical ones. In this chapter I will follow this tradition and use ‘syntactic’ and ‘sentence-categorical’ to refer to the same thing.)

In most languages, words cannot be combined randomly (just like morphemes cannot be combined randomly in making complex words). Whereas we have not clearly established that there are word-level categorical (morphological) repair rules, we will learn that the sentence-level categorical system has repair rules (called transformations).

You will see that the syntactic system is quite challenging. Modern Linguistics has been greatly influenced by Noam Chomsky, and he has mainly worked on syntax. As a result, the last four decades have produced a rich literature on the syntactic organization of many different languages. Over the years Chomsky and his followers have often changed course (i.e., formalisms and terminology), and I will not be able to do full justice to this energetic subfield of linguistics, but I will give it more space than was used for the other grammatical modules. Fasten your seat belts!
Syntactic Wellformedness

Consider the following string of words:

The boy throws the ball.

When we say that this sentence is grammatical, we mean that:

- Word categories have been grouped and linearized in the right way (i.e., the sequence is categorically wellformed).
- We can make sense of it (i.e., it is semantically wellformed).
- We can pronounce it (i.e., it is phonotactically wellformed).

It thus follows that sentences can be ungrammatical for three reasons. Consider the following sentences:

a. *The boy the ball throws.
   *Boy the throws ball the.

b. *Colorless green ideas sleep furiously.
   *He is a man and he is not a man

c. *I saw a elephant
   *Mary's smarter than John's.

A case can be made for saying that the sentences in (a) is ungrammatical for categorical reasons, in (b) for semantic reasons, and in (c) for phonotactic reasons.

With reference to the first semantic example in (b), it could be claimed that this sentence is not semantically illformed, but just semantically odd if used in a world in which ideas cannot sleep or have a color. Still, it seems that speaking of something being colorless and green constitutes a semantic contradiction and must thus be bad in whatever world we imagine. The first semantic example in (b) was constructed by Chomsky. He meant to demonstrate that a sentence can be syntactically wellformed while being illformed in semantic terms. This shows that syntactic and semantic wellformedness (and we might add phonotactic wellformedness) are independent of each other.

With respect to (c), I already mentioned the fact that in English the definite article must have the phonemic shape ‘an’ when the following noun starts with a vowel. In the second example, it would appear that the phonotactic contraction of is to ‘s is not allowed at the end of a sentence.

With respect to the sentences in (a) the proper reason for its illformedness seems neither semantic nor phonotactic, and this leaves only categorical (‘syntactic’) illformedness as an option. There must be a system of categorical constraints that tells a speaker of English that the sentences in (a) are no good.

You might ask how we know that there is a categorical system for constructing sentences? Perhaps we’ve simply learned all the English sentences that we use in the course of language acquisition. Of course, you could have raised the same point when we discussed the
morphological system. But no, this can’t be correct because there does not seem to be a limit on
the number of sentences (or complex words). Due to recursivity in the system of combination
rules (we’ll see that sentence structure is also recursive), the number of sentences and words is
infinite and you can’t learn an infinite number of things by heart no matter how smart you are.
I’ll demonstrate below how the property of recursivity accounts for the fact that there is indeed
no limit on the number of sentences.

The Syntactic Primitives and Combination Rules

Like the other grammatical systems, the syntactic system consists of three parts:

```
finite set of primitives (category labels)
wellformed categorical combinations (syntactic structures)
finite set of combination rules
repair rules
```

We have seen before that the various systems of grammar consist of three things. First, each
component contains a list of basic building blocks (primitives). Second, there is a set of rules or
constraints for combining these basic expressions into complex expressions. Third, there is, or
can be, a set of rules operating on the complex expressions to make adjustments. In the case of
the syntactic system we will discuss repair rules which are called transformational rules.

It is important to understand that the syntactic–categorical system is simply a
continuation of the morphological–categorical system. The main difference is that the latter
produces units of the size word, whereas the former system starts with words and produces units
of the size phrase and sentence.

Not all linguists consider the word-sentence division significant and it is certainly the case that
there isn’t a lot of difference between the types of structures that are constructed within these two
domains. Nor will it always be immediately obvious whether a certain complex expression is a
complex word or a sentential construction. Another reason for not maintaining a strict separation
between a word and sentence module is that such a separation seems to imply that complex
words cannot be formed out of, for example, a phrase (a piece of sentence) and an affix. This
raises the question of how to analyze an expression like:

American history teacher

It would seem that the part American history is a phrase consisting of an adjective and a noun.
But if the whole thing is a compound we have a phrase inside a word. Note that this same string
of words can also be interpreted as consisting of the compound history teacher to which we have
added an adjective. This gives a different meaning, namely a history teacher who is an
American:
It is indeed the case that this expression is ambiguous and could mean either a teacher of American history or an American teacher of history, but the point of importance here is that the first reading presupposes a structure with a phrase inside a word. We could say that syntactic constructions can simply be fed back into the categorical word module, but that would allow for all sorts of constructions that would appear to be absent in the languages of the world. Another questioning the word / phrase distinction is that, as we have argued, inflection occurs after syntax. Yet, inflected units seem to be words! I will not try to solve these issues here and simply proceed on the assumption that to maintain the division between words and phrases/sentences is a useful one, and one that captures a significant property of human languages.

The Primitives of the Syntactic System

The primitives of the syntactic system are the category labels of words. I’ve given you a list of these in the previous chapter in the context of categorical properties of morphemes. Here I assume that since the categorical properties of words are projection from the categorical properties of morphemes, both the morphological and the syntactic system have the same primitives.

The Combination Rules

Here are some examples of syntactic–categorical combination (rewrite) rules:

<table>
<thead>
<tr>
<th>Combination rules</th>
<th>Explanation of symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. $S \Rightarrow NP \ VP$</td>
<td>$S =$ Sentence</td>
</tr>
<tr>
<td>b. $NP \Rightarrow det \ N$</td>
<td>$NP =$ Noun Phrase</td>
</tr>
<tr>
<td>c. $VP \Rightarrow V \ NP$</td>
<td>$VP =$ Verb phrases</td>
</tr>
<tr>
<td>d. $VP \Rightarrow V$</td>
<td>$V =$ Verb</td>
</tr>
<tr>
<td>e. $VP \Rightarrow V \ S$</td>
<td>det = determiner</td>
</tr>
</tbody>
</table>

A special name for these rules is phrase structure rules. As before, these rules produce tree structures:
When checking the categorical properties of the sentence The boy throws the ball we establish:

1. **Insertion Requirement**: Words must be combined in accordance with their insertion requirements.
2. **Projection Requirement**: The label of a phrase is identical to the label of its head.

The form (phonotactic) aspects of the words, as well as their meaning aspects, need to be organized in wellformed structures as well for the sentence, as a whole, to be wellformed.

Now look at the following structure:

This structure is possible because the rule system above is recursive. Rules (a) and (e) are a recursive set: Both rules have a symbol on the left-hand side of the rule that the other rule has on the right-hand side; these rules are mutually feeding.
More Examples of Recursion in Syntax

Let us consider some more examples of recursion in syntax. Consider the following sentence:

\[
S \\
VP \\
NP \quad V \\
I \quad saw \quad the \quad man \quad pull \quad a \quad rabbit \quad out \quad of \quad his \quad sleeve
\]

The structure is recursive because we have a sentence inside another sentence, because the rules (a) and (e) form a recursive set. (The closed triangle hides the structure of the embedded sentence, because we wish to suppress unnecessary details.)

Recursivity can also be introduced by rules that contain conjunctions:

d. NP \Rightarrow NP \text{ conjunct } NP \quad \text{(the man or the woman)}

e. VP \Rightarrow VP \text{ conjunct } VP \quad \text{(eats and sleeps)}

f. AP \Rightarrow AP \text{ conjunct } AP \quad \text{(beautiful and rich)}

g. S \Rightarrow S \text{ conjunct } S \quad \text{(he drinks and she eats)}

The mere fact that we can stick phrases inside phrases is a remarkable property of language but does not in itself constitute recursion. Recursion arises when we can stick phrases inside phrases of the same type. (e.g., an NP inside an NP, and so on). Recursion is a very specific, simple, yet powerful device that essentially gives language its infinite range of words and sentences.

Recursion is a way to compress different sentences into one sentence, either basically maintaining the sentence structure for the embedded unit \( \text{(that stands on a hill)} \) or shortening it to a phrase \( \text{(on a hill)} \):

\[
\text{Did you see that? Mary fell! } \Rightarrow \text{ Did you see that Mary fell?}
\]

\[
\text{I see a house. The house stands on a hill } \Rightarrow \text{ I see a house that stands on a hill}
\]

or: \[
\text{I see a house on a hill}
\]

\[
\text{John came. Mary went } \Rightarrow \text{ John came and Mary went}
\]
I now turn to a discussion of some important general properties of phrase structure rules, explaining a notation that continues the integer notation that we introduced for morphological–categorical rules.

We have said that words are organized into phrases. Like words, phrases belong to different categories. We have noun phrases (NPs), which are built around a noun, verb phrases (VPs), which are built around a verb, and so on. Thus, words have the power to collect other words around them and form a phrase in which they are the central element. In line with our discussion in the preceding chapter, we call this central element the head, because it is this element that determines the category of the phrase.

Extending the integer notation introduced in the previous chapter (-er is $N^{-1}$, sleep is $V^0$), we will now say that a phrase is a $+1$ (or simply 1) unit. After all, it goes one level up from the zero-level for words. Here’s an example of a Prepositional Phrase (PP), a phrase that has a preposition (the word on) as its head:

As of now, I will adopt the convention to draw a vertical line over the head unit. A general term for non-heads is dependents, which are below a slant line in our tree notation.

I just said that a head forms a phrase by combining with other words. This is slightly inaccurate. A head word combines with, on the one hand single words (like the in the house) and, on the other hand, with phrases. Thus house in the following example combines with the phrase on the hill.

We now face the question of how we represent the structure of the noun phrase the house on the hill in which house combines both with the and with on the hill. Here are three possibilities:
In the first structure we have created a ternary node: the node has three branches. But as we have argued before, linguists have claimed that there is a virtue in working with trees in which nodes can only branch in two. Thus we reject structure (a). The choice between (b) and (c) is less obvious. I will below provide an argument for believing that structure (c) is the best one.

But we now have to wonder how many layers we can have within phrases. Some linguists have claimed that all phrases maximally have a two-layered structure, i.e. as in structure (c). Let us refer to this as the Two Layer Principle (TLP). They all have a head that combines with a phrase (called the complement) to form layer 1. Layer 1 then combines with another unit (typically a single word) called the specifier to form layer 2. This creates two layers that we distinguish by our integer superscripts. So the universal structure for phrases is claimed to be

\[
\begin{align*}
X^2 & \quad X^1 \\
\text{Specifier} (Z^0) & \quad X^0 \quad \text{Complement} (Y^2)
\end{align*}
\]
The symbol $X$ here is a variable standing for $N$, $V$, $A$, that is, whatever the category of the head is. Both $X^1$ and $X^2$ can be read as XPhrase (Noun Phrase, Verb Phrase, etc.). The head word is a zero-level unit.

For the remainder of this chapter I will assume that the two-layer hypothesis is correct, even though there are phrases in most languages that pose a problem for this idea. I will now show some examples of the more common types of phrases in which the heads are nouns, verbs, adjectives, or prepositions:

$X^2$  
Specifier $X^0$ Complement  

- the house on the hill (head is $N$)  
- very fond of horses (head is $A$)  
- right under the table (head is $P$)  
- nearly hit the dog (head is $V$)

In the first two phrases the complements are $P^2$s; in the other two the complements are $N^2$s.

In some cases we note that the complement is obligatory. You cannot say that someone is very fond or that something is on. You have to add a phrase to make these expressions complete. Some verbs, too, seem to obligatorily come with complements. These things have to be specified in the lexicon for the words that are the heads. Verbs that must come with a complement (that functions as a direct object) are called transitive (like hit). Verbs that don’t take a complement are called intransitive (like sleep). These two types of verbs are called subcategories (or subclasses) of the category (or class) of verbs. As we have seen before, if heads have a property that they must occur with some other unit, this is indicate by giving them an insertion frame. The insertion frame for transitive verbs is:

\[ [-N^2] \]

The fact that in some cases a head has an obligatory phrase due to an insertion requirement provides us with the argument for grouping phrases and heads first, before we add the specifier. Insertion frames indicate that the relationship between heads and phrases is more ‘intimate’ than that between heads and specifiers. Specifiers never are obligatory—they are always optional—and all heads are supposed to have the option of taking one, which means that no head can be subcategorized as being unable to combine with a specifier.

At this point we need to discuss a subtle point that concerns our use of the superscript integers. Consider the phrase: [mice] (as in my mother hates mice). We have several ways of representing this phrase:
The question is, do we assume the various layers to be universally present even when there is no specifier and/or complement, as in (a)? Or do we take something simpler as in (c). In principle I assume (a) to be correct, but I will adopt structure (b) in case there is no specifier, which I regard as a short hand for structure (a).

**Word Order Across Languages**

An important point that we can make with reference to the preceding examples of phrases is that English appears to choose a fixed order of specifier >> head >> complement for all phrase types (">>" means precedes). If all languages worked this way, one might refer to this point as a **language universal**. This universal, if valid, would be a good candidate for a language-specific innate characteristic because there is no apparent reason for why the order would be fixed in this way. Why couldn’t English have the following structure:

The on the hill house big

If all languages would have the same structure for phrases as English, such an apparently contingent (i.e., not logically necessarily) property could only come from an innate trait.

When linguists started collecting data from many different languages they made some very interesting discoveries, which show that phrase structure isn’t *that* universal. There are languages that have different phrase structures than those in English. However, it would still appear that there are some interesting restrictions on the variation. Very briefly, let me suggest some important discoveries that bear on phrase structure.

The first observation is that the order *specifier >> head >> complement* is not found in all languages. Again, focusing on the order of head and complement, it was found that in some languages the order is complement >> head, in *Japanese* for example. It is striking, however, that *Japanese* places the complement before the head *in all phrase types*. I’ll illustrate this for the verb phrase and the prepositional phrase:

<table>
<thead>
<tr>
<th>English</th>
<th>Japanese</th>
</tr>
</thead>
<tbody>
<tr>
<td>John [hit Mary]</td>
<td>John-ga [Mary-o butta]</td>
</tr>
<tr>
<td>VP: head first</td>
<td>VP: head last</td>
</tr>
<tr>
<td>English: by car</td>
<td>Japanese: kuruma de</td>
</tr>
<tr>
<td>PP: head first</td>
<td>PP: head last</td>
</tr>
</tbody>
</table>

10
(The terminology head first/head last focuses on the X^1 level, and thus ignores the specifier.)

Although not shown here, the specifier in both languages comes first in the phrase. The difference, then, lies in the order of head and complement. If languages can differ like this, the order of head and complement cannot be universally fixed. Rather, it appears that there is a choice. A choice like this is called a parameter.

A second important observation is the following. Despite the fact that languages vary in the ordering of units within phrases, there still appear to be constraints. The most important constraint is that within each language the same ordering is maintained in all phrase types. We do not find cases in which the NP has a complement-head order while the VP has the opposite. If this finding is robust, it is very interesting because there is no clear (logical) reason for why this limitation would exist. And if that is so, it seems likely that this property that languages have must be due to an innate fixation.

We have seen that in complex words in English the head is on the right. Now we have learned that in phrases (considering head and complement), the head is on the left. In both cases the choice of the side at which the head is located is a choice for English, because other languages can have heads on the other side in complex words or phrases or both. The fact that complex words and phrase can locate their head on opposite side, as English does, can be taken as an additional argument for making a distinction between words and phrases.

**Syntactic Repair Rules**

We have now concluded our discussion of the phrase structure rules (i.e., the combination rules in the sentence level categorical module). Next, we will discuss the need for repair rules in this module. Let’s see what it is that we are looking for. What was it again that repair rules do? Recall what the repair rules in the word phonological system do.

Phonological repair rules:
- Replace phonemes by other phonemes, or they delete or insert phonemes

The syntactic theory that has been proposed by Noam Chomsky relies crucially on a class of syntactic repair rules that are very similar in function to the phonological repair rules. Chomsky proposed that we need to have rules that move, insert, and delete words and phrases. He called these rules **transformations**.

**Transformations**

We start the discussion about transformational rules with an examination of normal statement sentences and corresponding question sentences. Compare the following two sentences:

(1) a. John will buy flowers.
    b. What will John buy?
In both sentences we have the verb *buy*, which takes a subject (*John*) and an obligatory object: *flowers* in (1a) and *what* in (1b). Also, in both cases, *buy* is accompanied by an auxiliary verb *will*. You may find it strange to think of the word *what* in (1b) as being the object of *buy*. However, you will agree, I hope, that *what* in some sense *stands for* the thing that will be bought, just like *flowers* in (1a) is the thing that will be bought.

What we have here, then, is a pair of sentences that are very similar, except for the fact that one is a **statement**, whereas the other is a **question**. Both sentences have the same parts (i.e., subject, object, verb, auxiliary), but they occur in different orders:

(2)    John  will  buy  flowers

What  will  John  buy?

The subject *John* and the auxiliary *will* occur in opposite orders, and the object *what* in the question occurs all the way at the beginning of the sentence, whereas in the statement the object *flowers* immediately follows the verb. It seems reasonable to suppose that being a statement or being a question is what causes these differences in word order.

We might now suppose that both sentences are derived from one basic **sentence template** and that when that structure occurs in the environment of a **question marker**, repair rules (transformations) apply. This is the basic idea that Chomsky had in the 1950s.

Perhaps this approach reminds you of a phenomenon that we discussed in Chapter 10 (on phonology). We noticed there that a particular morpheme sometimes has two different phonemic shapes depending on its context, that is, on morphemes that it is combined with. We discussed the following example:

(3)   electri[k]  -  electri[s]ity

In that case, we analyzed the alternation between *electri[k]* and *electri[s]* by formulating an repair rule that changes *k* into *s* before *-ity*. It would seem that the different shapes of statements and questions can also be regarded as alternations of a basic sentence template, which suggests that the alternation between the two sentences can be handled in a similar way. Noting the analogue between allomorphy (different surface shape of a morpheme) and the variation between the structure of statements and question sentences brings out that phonological repair rules (in this case a rule replacing /k/ by /s/) are very similar in function to the class of transformational rules that we are now ready to introduce. We had to decide which of the two variants is basic (*electrik* or *electris* in the case of allomorphy. In syntax, we need to decide on the basic organization of the sentence template (the statement or the question). Instead of putting you through the exercise of trying out both options, I’ll make what I know to be the correct choice and say that **the statement order is basic**. If that is so, we need two adjustment rules:
(4)  

a. *Subject-Aux inversion:* Subject Aux $\Rightarrow$ Aux Subject in a Q-sentence  

b. *Q-movement:* Move a Q-word to the beginning in a Q-sentence  

Rule (4a) switches the linear order of the subject NP and the auxiliary verb, whereas rule (4b) moves the question word *what* to the beginning of the sentence.

Before we see how this works, I must introduce what we will take to be the overall structure of a sentence. Earlier, we adopted something as in (5a), but now we will use (5b). The motivation is, first, that the sentence must have a head (like all phrases) and, second, the auxiliary must have a place in the sentence:

(5)  

a. 

```
(\text{\text{S}})
```

```
(\text{\text{N}^2})
```

```
(\text{\text{V}^2})
```

b. 

```
(\text{\text{Aux}^2})
```

```
(\text{\text{N}^2})
```

```
(\text{\text{Aux}^1})
```

```
(\text{\text{Aux}^2})
```

```
(\text{\text{V}^2})
```

When we introduced (5a) we did not consider sentences that, next to the main verb, have a so-called auxiliary verb (*John will come*). These auxiliary verbs need a place too, and the proposal in (5b) is, in fact, that these verbs constitute the head of the sentence (a rather fancy spot for verbs that are called *auxiliary*). In other words, we treat the sentence as a two-layer phrase with the auxiliary as the head, the subject NP as the specifier, and the object NP as the complement. Thus Aux$^2$ is equivalent to the symbol S.

Let's now look at the application of the two transformational rules:

(6) *Phrase Structure Rules* 

```
(\text{\text{Aux}^2}(=\text{\text{Sentence}}))
```

```
(\text{\text{N}^2})
```

```
(\text{\text{Aux}^1})
```

```
(\text{\text{N}^0})
```

```
(\text{\text{Aux}^0})
```

```
(\text{\text{V}^2})
```

```
(\text{\text{V}^0})
```

```
(\text{\text{N}^2})
```

```
(\text{\text{N}^0})
```

There are two problems here regarding these transformation. Firstly, what is the structure after the transformations have been applied? Specifically, are we saying that *John* ends up in the Auxiliary spot and *will* in the Noun Phrase spot? That would be rather odd because that would
create a mismatch between the labeling of the terminal nodes in the tree and the word class properties of the words involved. Secondly, when we move what to the front, where does it end up? It doesn’t seem to be part of the sentence structure anymore. That cannot be right either. If we compare this to the phonological repair rules that substitutes /s/ for /k/, we have assumed the /s/ is integrated into the syllable structure of the word by occupying an onset position in fourth syllable of the word. Analogously, we want to make sure that the word what will be somehow integrated into the sentence structure.

To solve the first problem, let us say that John does not go anywhere and that instead will moves to a position in front of John. This, of course, only solves half of the problem because we now need to determine how will can be integrated into the sentence structure. By moving both will and what to the front of the sentence we seem to have two words hanging in midair, as shown in (7):

(7)  

So let us see whether we can arrange things in such a way that both what and will end up being incorporated into the phrase structure of the sentence. It would seem that we need to formulate our phrase structure rules in such a way that two extra positions at the beginning of the sentence will be present, but the question is how to motivate such extra positions independently. By this I mean: Is there any independent evidence for having these extra positions? Yes, there is. The evidence will come from comparing so-called main sentences with embedded sentences. This means that we need to make a little detour here and look at the structure of embedded sentences. The difference between a main sentence and an embedded sentence is shown in the following sentence:

(8)  

What we have here are actually two sentences:

(9)  

Main sentence
Embedded sentence
[Harry said X] is a sentence comparable to [Harry said nothing]. But instead of the object NP nothing, we fill the object slot by a whole sentence [that John will buy flowers]. (We can do that, i.e., have a sentence inside a sentence, because of recursion.) In other words, the verb to say can take as its complement either a noun phrase or a whole sentence. Let us see which phrase structure rules we need now. (10a) and (10c) are the equivalent of some of the rules that I introduced earlier. Rule (10b) is needed to know that we have an auxiliary phrase. Note that (10c) allows the verb to say to take a sentence as its complement:

\[
(10)\begin{align*}
a. \text{Aux}_2 & \Rightarrow \text{N}_2 \text{Aux}_2 \\
b. \text{Aux}_1 & \Rightarrow \text{Aux}_0 \text{V}_2 \\
c. \text{V}_1 & \Rightarrow \text{V}_0 \text{Aux}_2
\end{align*}
\]

(cf. S ⇒ NP VP) (cf. VP ⇒ V S)

(Observe that rules (a) and (c) are mutually feeding and thus make the rule system recursive.)

At the same time we must assume that the verb to say occurs in the lexicon with two insertion frames, one for its occurrence with a Noun Phrase (say nothing) and the other for its occurrence with a sentence (say that…):

\[
(1a) [ - \text{N}_2 ] \text{v}_1 \quad (b) \quad [ - \text{Aux}_2 ] \text{v}_1
\]

Now, the point to notice is that the embedded sentence starts with a special word, that. Such words belong to the word class of complementizers. Given that all words must be inserted in a terminal position in the phrase structure, an embedded sentence must have a position for complementizers. This necessitates another phrase structure rule that will be written as in (11):

\[
(11) \quad \text{Comp}_1 \Rightarrow \text{Comp}_0 \text{Aux}_2
\]

(By the way, the term complementizer is distinct from the term complement. Both terms are related in as far as complementizers introduce sentences that are complements of a verb.)

With the rule in (11), and the other rules that we already used, we can generate the following structure for an embedded sentence where we have added an extra shell to the sentence. This shell is headed by the complementizer position Comp\(^0\). The sentence (Aux\(^2\)) is thus the complement of this complementizer:
In a way, we can say that the word *that* has an insertion frame that indicates that this word can take a sentence as a complement: \([ - \text{Aux}^2 ]_{\text{Comp}^1}\).

Given that embedded sentences start with a complementizer such as *that*, it would seem that embedded sentences show us yet another layer of the basic sentence template which we need to take into account if we want it to underlie a three-way variation in sentence structure:

(13)  

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>John will buy flowers.</td>
<td>statements</td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td>What will John buy?</td>
<td>questions</td>
<td></td>
</tr>
<tr>
<td>c.</td>
<td>that John will buy flowers</td>
<td>embedded sentence</td>
<td></td>
</tr>
</tbody>
</table>

Still pursuing the idea that the different shapes that sentences can take must *all* be derived from one common structure, I will show that we can solve the problems raised by (7) if we adopt the idea that something like the structure in (12) underlies all three sentence types in (13).

At first sight it would seem that the Comp shell provides only *one* extra position, the position that is occupied by *that* in an embedded sentence. However, if we regard this shell as a phrase in its own right (a Comp phrase or Comp\(^2\)), we expect its head to not only have a complement (i.e., the sentence or Aux\(^2\)), but also a specifier. With two extra positions we can now say that *what* moves to the specifier of Comp, whereas the auxiliary *will* moves to the head of Comp:
We have now shown that both what and will can be moved to a position in the tree structure so that they no longer have to float in mid air. Notice that both movements leave a gap in the structure, namely at the original site of these two words, but we will say that it is acceptable that not all terminal positions in a sentence structure are occupied by an actual word. (I’ll come back to this point below.) The basic form of a sentence is usually called its deep structure and the surface form (which results from applying the transformations) its surface structure.

There is one glitch: if will moves to the Comp⁰ position, we place an Aux⁰ unit in a position that mismatches with that category label. The same kind of problem does not arise with the movement of what since we do not have to assume that the specifier position has an inherent category.

Let me finally update the phrase structure rules:

(15) a. Comp² \Rightarrow Spec Comp¹
    b. Comp¹ \Rightarrow Comp⁰ Aux²
    c. Aux² \Rightarrow N² Aux²
    d. Aux¹ \Rightarrow Aux⁰ V²
    e. V¹ \Rightarrow V⁰ Comp²

So far I’ve talked about transformations as if they move words, which are terminal elements in the tree. However, in the following sentence Q-movement has moved a phrase rather than a single word:
Thus, transformations must be able to make reference to structure (i.e., groups of words). We refer to this as the **Principle of Structure Dependency**. Transformations are structure dependent. They don’t operate on linear strings of words; they take into account how these words are organized into phrases. This means that they can target non-terminal nodes as well. This principle, in fact, supports the idea that sentences are not just linear arrangements of words, but are instead grouped into phrases.

In fact we can assume that transformation always operate on phrase because single words (with no specifier or complement) are also considered to be phrase. The fact that transformations operate on phrase confirm that the categorical checking system is part of the merge component. After the need for transformation had just been proposed, syntacticians also formulated transformations that would insert phrases or delete them. Later, they decided that only movement transformations are needed, so I will not bother to discuss data that at first appeared to motivate these other kinds of operations.

It has been suggested that that transformational rules do not move units, but instead **copy** them. This leads to two occurrences of the units that are copied, which means that one of them must be deleted, or left unpronounced. Differences among languages could then lie in whether the original or the copy is pronounced.

| Deep structure: John will buy what | Copying what will John will buy what |
| Copying what will John will | Deletion what John will buy |
| Deletion what will | Deletion will John will buy what |

The various deletions are options that make languages different. Here I will not discuss the merits or problems for this copying-plus-deletion theory of movement.

**Why Moves Things Around?**

One might wonder why it is that words have to be moved (or copied) in a Q-sentence. Why should the Q-word come at the beginning, and why should the Aux-element move to the complement in questions but not in statements? The most direct reason for transformations is that the structures that they apply to violate a categorical constraint of the grammar. Hence there is a constraint which states that a Qphrases cannot occur anywhere else than at the beginning of the sentence. The phrase structure rules, however, allow Qphrases in positions where NPs normally occur and that is never a problem, except when the phrase is a Qphrase. Most important is that there is a pretty compelling reason for why Qphrases have to be located in places where they cannot stay. I will discuss that point below.
Another question is why English would have such a constraint on where Qphrases can occur. Some languages other than English (such as Chinese languages) do not move question words to the front of the sentence. Apparently, languages can differ in terms of the constraints that they have and thus in terms of the transformational rules that they have, just as they can differ in terms of their phonotactic constraints and repair rules. Perhaps the original motivation for moving question phrases to the front was to put them in ‘the spot light’, i.e. to draw extra attention to them.

Note, by the way, that even in English one can leave the question word in its original position:

You bought what?

Such questions are called *echo questions*. In English you can only use these expressions in response to someone saying what he bought. So, in fact, an echo question is not a question because you already know the answer. A regular question (What did you buy?), on the contrary, is used when someone has not yet informed you what he bought.

*Why do we first put Qphrases in the ‘wrong’ place*

Let us now ask why we put Qphrases in places from where they subsequently have to be moved? Let us assume for a moment that questions and statements are not derived from a common phrase structure but instead are produced independently from each other. This could be motivated by the preference to do away with transformational rules. If we make that move we do not even have to add any phrase structure rules because we could continue to assume that all sentences have the same general Comp shell structure and simply allow positions to remain empty, which we have already allowed anyway. In this approach we would produce the sentence *What will John buy?* directly, without any transformations.
Fair enough, but now consider the needs of the verb *buy*. *Buy* is a transitive verb, and that means that its insertion in a sentence structure is controlled by a **subcategorization property** expressed in its insertion frame that says that this verb **must** be followed by a direct object: \[ - N^2 \]. It would seem that this property is not satisfied in the structure in (17) because there is no direct object immediately following *buy*. In other words, as far as the grammar is concerned, the preceding sentence should be ruled out in the same way that the following sentence is ruled out:

(17) Phrase structure rules & primitives:

```
Comp^2
    Comp^1
        Comp^0  Aux^2 (=Sentence)
            N^2  Aux^1
                N^0  Aux^0
                    V^2  V^0
what  will  John  buy
```

(18) *John will buy

We know that the question word *what* in (17), which is all the way in the front of the sentence, acts as the direct object of *buy*! But if we generate the sentence with the question word in the front we will have a hard time explaining how the grammar knows that *what* functions as the object of *buy*. We can’t just say, “Look for a Q-word at the beginning, because the following sentence is not grammatical either:

(19) *what buys?

In fact, another problem that we have when we insert the primitive *what* directly in front (rather than moving it there) is that we can then also make sentences like this one:

(20) *what will John buy flowers?

You will start to see, I hope, that it is going to be very difficult to come up with a precise procedure that will ensure that *buy* has one, and not more than one, object. We don’t have that problem if we follow the analysis that we provided earlier. We insert the question word to the
right of the verb. Then we check whether everything is wellformed, in particular whether buy has a direct object. This will be so; the direct object will be what. Then, because this is required by a constraint on the location of Qphrases, we move what to the specifier of Comp² and all is well.

Stepwise (Cyclic) Movement

Finally, let me mention one remarkable property of the movement of question words. Consider the following sentence:

(21) [When did the boy say [ he fell from the tree]]

This sentence is ambiguous because the when could inquire about the time of saying or the time of falling. This means that there are two underlying structures:

(22) a. [The boy said when [ he fell from the tree]]

b. [The boy said [ he fell from the tree when ]]

To derive the question from underlying structure (a), when moves from a position within the matrix sentence to the front of the matrix sentence. To derive the question from underlying structure (b), when moves from a position within the embedded sentence to the front of the matrix sentence.

It is interesting in its own right that a question word can move from within an embedded sentence up to the beginning of the matrix sentence. This shows that this kind of movement can cross sentence boundaries. In fact, it can cross more than one boundary:

(23) [When did the boy say [ he saw [ Mary fall from the tree ]]]

The underlying structure for this sentence is:

(24) [The boy said [ he saw [ Mary fall from the tree when ]]]

But the following are also possible underlying structures:

(25) a. [The boy said [ he saw when [ Mary fall from the tree ]]]

b. [The boy said when [ he saw [ Mary fall from the tree ]]]

This means that the sentence has three meanings.

One might wonder whether the movement from an embedded sentence moves the when word in one big step, or in smaller steps:

(26) [ - The boy said [ he saw [ Mary fall from the tree when ] ]]

21
The stepwise movement method may seem more complicated, but there is evidence that this is how things work. If we assume that movement is cyclic, it seems reasonable to expect that a movement cannot take place in cases where the landing site for the to-be-moved entity is already occupied so that the presence of a word in that position would block movement. Consider the following sentence:

(28) [When did the boy say [how he fell from the tree]]

In this sentence how occupies the position for question words. This means that when cannot have come from within the embedded sentence, as we can see in (29):
If the *when* in the embedded sentence cannot move to the front within its own sentence, it cannot move further to the front of the matrix sentence either. We can make this even more interesting. Consider the sentence:

\[(30) \quad [\text{When did the boy say} [\text{that he fell from the tree}]]\]

At first sight you would think that the word *that* must block the reading with *when* coming from the embedded sentence, just like the word *how* in (28) did. But there is a subtle difference between (28) and (30): the word *that* does not occupy the specifier positions of the embedded sentence, it stands in the Comp<sup>0</sup> position:

\[(31)\]

\[
\begin{array}{c}
\text{Comp}^2 \\
\text{Spec} \\
\text{Comp}^1 \\
\text{Comp}^0 \\
\text{Spec} \\
\text{Comp}^1 \\
N^2 \\
N^0 \\
\text{det} \\
did \\
\text{the} \\
\text{boy} \\
\text{say} \\
\text{spec} \\
\text{Comp}^1 \\
\text{Comp}^2 \\
\text{Aux}^2 \quad (=\text{Sentence}) \\
\text{V}^0 \\
\text{V}^2 \\
\text{Comp}^0 \\
\text{Aux}^1 \\
\text{Aux}^0 \\
\end{array}
\]
These facts support the idea that question words like *when* and complementizer words like *that* indeed occupy different positions. At the same time, this array of facts supports the idea that the movement of questions words takes place in a step-wise manner.

**Conclusions**

The syntactic–categorical system has the same general design as the phonological and morphological–categorical system (primitives, combination rules, and repair rules). The combination rules have the property of recursion, as in the morphological system.

We have come to the important conclusion that all major sentence types that we have looked at (embedded, questions, or statements) can be derived from one common phrase structure:

(32)

We then discussed the need for syntactic repair rules, called transformational rules. Syntactic rules apply first and derive the various sentence types (statements, questions, embedded sentences) from the common underlying structure in (32). If, given the location of specific kinds of phrases (such as Qphrases) certain constraints are violated, there is a need for transformation rules to apply.

Adding what we have learned in this chapter to the diagram at the end of the previous chapter, we now arrive at the following picture:

<table>
<thead>
<tr>
<th>Level: word</th>
<th>Primitives</th>
<th>Combinations rules</th>
<th>Adjustment rules</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phonological system</strong></td>
<td>Phonemes</td>
<td>Phonotactic rules</td>
<td>Phoneme substitution rules</td>
</tr>
<tr>
<td><strong>Categorical system</strong></td>
<td>Categorical labels on morphemes</td>
<td>Word formation rules</td>
<td>(none discussed)</td>
</tr>
<tr>
<td><strong>Level: sentence</strong></td>
<td>Categorical labels on words</td>
<td>Phrase structure rule</td>
<td>Transformational rules</td>
</tr>
</tbody>
</table>
You will recall that at the sentence level we have only briefly touched on the phonotactic side (in Chapter 10). In the next chapter, we will take a look at the semantic side of both words and sentences. As promised, when all these aspects are added up, we end up with a total of six grammatical modules, three at the word level and three at the sentence level.

We have learned in chapter 7 that after the syntax, there is further level of structure building, namely inflection. For reasons of time and space, I will not discuss this level here.