Chapter 8

Applications in Linguistics

So far, we have focused our attention on the discussion of scattered context grammars from a rather theoretical viewpoint in this book. As opposed to this theoretical approach, in the present chapter, we describe some of their applications in order to demonstrate their quite realistic use in practice. As a matter of fact, these grammars are useful to every scientific field that formalizes its results by strings in which there exist some scattered context dependencies spread over the strings. Since numerous scientific areas, ranging from cellular biology through neurology up to logic, formalize and study their results by using strings involving dependencies of this kind, describing applications of scattered context grammars in all these areas would be unbearably sketchy and, therefore, didactically inappropriate. Instead of an encyclopedic approach like this, we focus on *linguistics*, which represents a classical application area of formal grammars. We primarily narrow our attention to the investigation of *English syntax*, which describes the rules concerning how words relate to each other in order to form well-formed grammatical English sentences. We have selected syntax of this language because the reader is surely familiar with English very well. Nevertheless, analogical ideas can be applied to members of other language families, including Indo-European, Sino-Tibetan, Niger-Congo, Afro-Asiatic, Altaic, and Japonic families of languages. We explore several common linguistic phenomena involving scattered context in English syntax and explain how to express these phenomena by scattered context grammars.

By no means is this chapter intended to be exhaustive in any way. Rather, we consider only selected topics concerning English syntax and demonstrate how scattered context grammars allow us to explore them clearly, elegantly, and precisely. Compared to the previous parts of this book, which are written in a strictly mathematical way, we discuss and describe scattered context grammars less formally here because we are interested in demonstrating real applications rather than theoretical properties. Specifically, we primarily use scattered context grammars to transform and, simultaneously, verify that the English sentences under discussion are grammatical.

Apart from linguistic applications, discussed in this chapter, the next chapter, which concludes this book, sketches further application areas of these grammars.

8.1 Syntax and Related Linguistic Terminology

In the linguistic study concerning English syntax, we discuss and describe the principles and rules according to which we correctly construct and transform grammatical English sentences. To give an insight into the discussion of English syntax, we open this section by some simple examples that illustrate how we connect the theoretically oriented discussion of scattered context grammars in the previous chapters with the application-oriented discussion of English syntax in the present chapter. Then, we introduce the basic terminology used in syntax-oriented linguistics.

Introduction through Examples

Observe that many common English sentences contain expressions and words that mutually depend on each other although they are not adjacent to each other in the sentences. For example, consider this sentence:

He usually goes to work early.

The subject (*he*) and the predicator (*goes*) are related; sentences

**He usually go to work early.*

and

**I usually goes to work early.*

are ungrammatical because the form of the predicator depends on the form of the subject, according to which the combinations **he...go* and **I...goes* are illegal (throughout this chapter, * denotes ungrammatical sentences or their parts). Clearly, any change of the subject implies the corresponding change of the predicator as well. Linguistic dependencies of this kind can be easily and elegantly captured by scattered context grammars. Let us construct a scattered context grammar that contains this production:

$$(\text{He}, \text{goes}) \rightarrow (\text{We}, \text{go}).$$

This production checks whether the subject is the pronoun *he* and whether the verb *go* is in third person singular. If the sentence satisfies this property, it can be transformed to the grammatically correct sentence

We usually go to work early.

Observe that the related words may occur far away from each other in the sentence in question. In the above example, the word *usually* occurs between the subject and the predicator. While it is fairly easy to use context-sensitive grammars to model context dependencies where only one word occurs between the related words, note that the number of the words appearing between the subject and the predicator can be virtually unlimited. We can say

He almost regularly goes to work early.

but also

He usually, but not always, goes to work early.

and many more grammatical sentences like this. To model these context dependencies by ordinary context-sensitive grammars, many auxiliary productions have to be introduced to send the information concerning the form of a word to another word, which may occur at the opposite end of the sentence. As opposed to this awkward and tedious description, the single scattered context production above is needed to perform the same job regardless of the number of the words appearing between the subject and the predicator.

We next give another example that illustrates the advantage of scattered context grammars over classical context-sensitive grammars under some circumstances. Consider these two sentences:

John recommended it.

and

Did John recommend it?

There exists a relation between the basic clause and its interrogative counterpart. Indeed, we obtain the second, interrogative clause by adding *did* in front of *John* and by changing *recommended* to *recommend* while keeping the rest of the sentence unchanged. In terms of scattered context grammars, this transformation can be described by the scattered context production

$$(\text{John, recommended}) \rightarrow (\text{Did John}, \text{recommend});$$

clearly, when applied to the first sentence, this production performs exactly the same transformation as we have just described. Although this transformation is possible by using an ordinary context production, the inverse transformation is much more difficult to achieve. The inverse transformation can be performed by a scattered context production

$$(\text{Did, recommend}) \rightarrow (\varepsilon, \text{recommended});$$

obviously, by erasing *did* and changing *recommend* to *recommended*, we obtain the first sentence from the second one. Again, instead of *John* the subject may consist of a noun phrase containing several words, which makes it difficult to capture this context dependency by ordinary context-sensitive grammars.

Considering the examples above, the advantage of scattered context grammars is more than obvious: scattered context grammars allow us to change only some words during the transformation while keeping the others unchanged. On the other hand, context-sensitive grammars are inconvenient to perform transformations of this kind. A typical context-sensitive grammar that performs this job usually needs many more context-sensitive productions by which it repeatedly traverses the transformed sentence in question just to change very few context dependent words broadly spread across the sentence.

Terminology

Taking into account the intuitive insight given above, we see that there are structural rules and regularities underlying syntactically well-formed English sentences and their transformations. Although we have already used some common linguistic notions, such as subject or predicator, we now introduce this elementary linguistic terminology more systematically so we can express these English sentences in terms of their syntactic structure in a more exact and general way. However, we restrict this introduction only to the very basic linguistic notions, most of which are taken from [33, 34]. In the next chapter, which closes this book, we recommend several further excellent linguistic treatments closely related to the discussion of this chapter.

Throughout the rest of this section, we narrow our discussion primarily to verbs and personal pronouns, whose proper use depends on the context in which they occur. For instance, *is, are, was*, and *been* are different forms of the same verb *be*, and their proper use depends on the context in which they appear. We say that words in these categories *inflect* and call this property *inflection*. Verbs and personal pronouns often represent the key elements of a clause—the *subject* and the *predicate*. In simple clauses like

She loves him.

we can understand the notion of the subject and the predicate so that some information is "predicated of" the subject (*she*) by the predicate (*loves him*). In more complicated clauses, the best way to determine the subject and the predicate is the examination of their syntactic properties (see [33] for more details). The predicate is formed by a *verb phrase*—the most important word of this phrase is the verb, also known as the *predicator*. In some verb phrases, there occur several verbs. For example, in the sentence

He has been working for hours.

the verb phrase contains three verbs—*has, been*, and *working*. The predicator is, however, always the first verb of a verb phrase (*has* in the above example). In this study, we focus on the most elementary clauses—*canonical clauses*. In these clauses, the subject always precedes the predicate, and these clauses are positive, declarative, and without subordinate or coordinate clauses.

Next, we describe the basic categorization of verbs and personal pronouns, and further characterize their inflectional forms in greater detail.

Verbs

We distinguish several kinds of verbs based upon their grammatical properties. The set of all verbs is divided into two subsets—the set of *auxiliary verbs*, and the set of *lexical verbs*. Further, the set of auxiliary verbs consists of *modal verbs* and *non-modal verbs*. The set of modal verbs includes the following verbs—*can*, *may*, *must*, *will*, *shall*, *ought*, *need*, *dare*; the verbs *be*, *have*, and *do* are non-modal. All the remaining verbs are lexical. In reality, the above defined classes overlap in certain situations; for example, there are sentences, where *do* appears as an auxiliary verb, and in different situations, *do* behaves as a lexical verb. For simplicity, we do not take into account these special cases in what follows.

Inflectional forms of verbs are called *paradigms*. In English, every verb, except for the verb *be*, may appear in each of the six paradigms described in Table 8.1 (see [33]). Verbs in *primary form* may occur as the only verb in a clause and form the head of its verb phrase (predicator); on the other hand, verbs in *secondary form* have to be accompanied by a verb in primary form.

Form	Paradigm	Person	Example
Primary	Present	3rd sg	*She* walks *home*.
		Other	*They* walk *home*.
	Preterite		*She* walked *home*.
Secondary	Plain form		*They should* walk *home*.
	Gerund-participle		*She is* walking *home*.
	Past participle		*She has* walked *home*.

Table 8.1: Paradigms of English verbs

The verb *be* has nine paradigms in its neutral form. All primary forms have, in addition, their negative contracted counterparts. Compared to other verbs, there is one more verb paradigm called *irrealis*. The irrealis form *were* (and *weren't*) is used in sentences of an unrealistic nature, such as

I wish I were rich.

All these paradigms are presented in Table 8.2.

Personal pronouns

Personal pronouns exhibit a great amount of inflectional variation as well. Table 8.3 summarizes all their inflectional forms. The most important for us is the class of pronouns in *nominative* because these pronouns often appear as the subject of a clause.

Form	Paradigm	Person	Neutral	Negative
Primary	Present	1st sg	*am*	*aren't*
		3rd sg	*is*	*isn't*
		Other	*are*	*aren't*
	Preterite	1st sg, 3rd sg	*was*	*wasn't*
		Other	*were*	*weren't*
	Irrealis	1st sg, 3rd sg	*were*	*weren't*
Secondary	Plain form		*be*	—
	Gerund-participle		*being*	—
	Past participle		*been*	—

Table 8.2: Paradigms of the verb *be*

Non-reflexive				Reflexive
Nominative	Accusative	Genitive		
Plain		Dependent	Independent	
I	*me*	*my*	*mine*	*myself*
you	*you*	*your*	*yours*	*yourself*
he	*him*	*his*	*his*	*himself*
she	*her*	*her*	*hers*	*herself*
it	*it*	*its*	*its*	*itself*
we	*us*	*our*	*ours*	*ourselves*
you	*you*	*your*	*yours*	*yourselves*
they	*them*	*their*	*theirs*	*themselves*

Table 8.3: Personal pronouns

8.2 Transformational Scattered Context Grammars

As we have already mentioned, in this chapter, we primarily apply scattered context grammars to transform grammatical English sentences to other grammatical English sentences. To do so, we next slightly modify scattered context grammars so they start their derivations from a language rather than a single start symbol.

Even more importantly, these grammars define transformations of languages, not just their generation.

Definition 8.1. A *transformational scattered context grammar* is a quadruple

$$G = (V, T, P, I),$$

where

- V is the *total vocabulary*;
- $T \subset V$ is the set of terminals (or the *output vocabulary*);
- P is a finite set of scattered context productions;
- $I \subset V$ is the *input vocabulary*.

The derivation step is defined as in scattered context grammars (see Definition 2.40). The *transformation T that G defines from $K \subseteq I^*$* is denoted by $T(G,K)$ and defined as

$$T(G,K) = \left\{(x,y) : x \Rightarrow_G^* y, x \in K, y \in T^*\right\}.$$

If $(x,y) \in T(G,K)$, we say that *x is transformed to y* by G; x and y are called the *input* and the *output sentence*, respectively.

As already pointed out, while scattered context grammars generate strings, transformational scattered context grammars translate them. In a sense, however, the language generated by any scattered context grammar $G = (V, T, P, S)$ can be expressed by using a transformational scattered context grammar $H = \big(V, T, P, \{S\}\big)$ as well. Observe that

$$L(G) = \Big\{y : (S,y) \in T\big(H, \{S\}\big)\Big\}.$$

Before we make use of transformational scattered context grammars in terms of English syntax in the next section, we give two examples to demonstrate a close relation of these grammars to the theoretically oriented studies given previously in this book. To link the theoretical discussions given in the previous chapters of this book to the present chapter, the first example presents a transformational scattered context grammar that works with purely abstract languages. In the second example, we discuss a transformational scattered context grammar that is somewhat more linguistically oriented.

Example 8.2. Define the transformational scattered context grammar

$$G = (V,T,P,I),$$

where $V = \{A,B,C,a,b,c\}$, $T = \{a,b,c\}$, $I = \{A,B,C\}$, and

$$P = \big\{(A,B,C) \to (a,bb,c)\big\}.$$

For example, for the input sentence *AABBCC*,

$$AABBCC \Rightarrow_G aABbbcC \Rightarrow_G aabbbbcc.$$

Therefore, the input sentence $AABBCC \in I^*$ is transformed to the output sentence $aabbbbcc \in T^*$, and

$$(AABBCC, aabbbbcc) \in T(G,I^*).$$

If we restrict the input sentences to the language $L = \{A^nB^nC^n : n \geq 1\}$, we get

$$T(G,L) = \big\{(A^nB^nC^n, a^nb^{2n}c^n) : n \geq 1\big\},$$

so every $A^nB^nC^n$, where $n \geq 1$, is transformed to $a^nb^{2n}c^n$.

In the following example, we modify strings consisting of English letters by a transformational scattered context grammar, and in this way, we relate these grammars to lexically oriented linguistics—that is, the area of linguistics that concentrates its study on vocabulary analysis and dictionary design.

Example 8.3. We demonstrate how to lexicographically order alphabetic strings and, simultaneously, convert them from their uppercase versions to lowercase versions. More specifically, we describe a transformational scattered context grammar G that takes any alphabetic strings that consist of English uppercase letters enclosed in angle brackets, lexicographically orders the letters, and converts them to the corresponding lowercases. For instance, G transforms $\langle XXUY\rangle$ to *uxxy*.

More precisely, let J and T be alphabets of English uppercases and English lowercases, respectively. Let $\prec$ denote *lexical order* over J; that is, $A \prec B \prec \cdots \prec Z$. Furthermore, let h be the function that maps the uppercases to the corresponding lowercases; that is, $h(A) = a$, $h(B) = b$, $\ldots$, $h(Z) = z$. Let i denote the inverse of h, so $i(a) = A$, $i(b) = B$, $\ldots$, $i(z) = Z$. Let $N = \{\hat{a} : a \in T\}$. We define the transformational scattered context grammar $G = (V,T,P,I)$, where T is defined as above, $I = J \cup \big\{\langle,\rangle\big\}$, $V = I \cup N \cup T$, and P is constructed as follows:

1. For each A, $B \in I$, where $A \prec B$, add
   $(B,A) \to (A,B)$ to P.

2. For each $a \in T$, add
   $\big(\langle\big) \to (\hat{a})$ to P.

3. For each $a \in T$ and $A \in J$, where $i(a) = A$, add
   $(\hat{a}, A) \to (a, \hat{a})$ to P.
4. For each $a, b \in T$, where $i(a) \prec i(b)$, add
   $(\hat{a}) \to (\hat{b})$ to P.
5. For each $a \in T$, add
   $(\hat{a}, \rangle) \to (\varepsilon, \varepsilon)$ to P.

Set $K = \{\langle\}J^*\{\rangle\}$. For instance, G transforms $\langle ORDER\rangle \in K$ to $deorr \in T^*$ as

$$\begin{aligned}\langle ORDER\rangle &\Rightarrow_G \langle OEDRR\rangle \Rightarrow_G \langle DEORR\rangle \\ &\Rightarrow_G \hat{d}DEORR\rangle \Rightarrow_G d\hat{d}EORR\rangle \Rightarrow_G d\hat{e}EORR\rangle \Rightarrow_G de\hat{e}ORR\rangle \\ &\Rightarrow_G de\hat{o}ORR\rangle \Rightarrow_G deo\hat{o}RR\rangle \Rightarrow_G deo\hat{r}RR\rangle \Rightarrow_G deor\hat{r}R\rangle \\ &\Rightarrow_G deorr\hat{r}\rangle \Rightarrow_G deorr,\end{aligned}$$

so $(\langle ORDER\rangle, deorr) \in T(G, K)$. Clearly, G can make the same transformation in many more ways; on the other hand, notice that the set of all transformations of $\langle ORDER\rangle$ to $deorr$ is finite.

More formally, we claim that G transforms every $\langle A_1 \dots A_n\rangle \in K$ to $b_1 \dots b_n \in T^*$, for some $n \geq 0$, so that $i(b_1) \dots i(b_n)$ represents a permutation of $A_1 \dots A_n$, and for all $1 \leq j \leq n-1$, $i(b_j) \prec i(b_{j+1})$ (the case when $n = 0$ means that $A_1 \dots A_n = b_1 \dots b_n = \varepsilon$). To see why this claim holds, notice that $T \cap I = \emptyset$, so every successful transformation of a string from K to a string from T^* is performed so that all symbols are rewritten during the computation. By productions introduced in (1), G lexicographically orders the input uppercases. By a production of the form $(\langle) \to (\hat{a})$ introduced in (2), G changes the leftmost symbol $\langle$ to $\hat{a}$. By productions introduced in (3) and (4), G verifies that the alphabetic string is properly ordered and, simultaneously, converts its uppercase symbols into the corresponding lowercases in a strictly left-to-right one-by-one way. Observe that a production introduced in (2) is applied precisely once during every successful transformation because the left-to-right conversion necessities its application, and on the other hand, no production can produce $\langle$. By a production from (5), G completes the transformation; notice that if this completion is performed prematurely with some uppercases left, the transformation is necessary unsuccessful because the uppercases cannot be turned to the corresponding lowercases. Based upon these observations, it should be obvious that G performs the desired transformation.

Having illustrated the lexically oriented application, we devote the next section solely to the applications of transformational scattered context grammars in English syntax.

8.3 Scattered Context in English Syntax

In this section, we apply transformational scattered context grammars to English syntax. Before opening this topic, let us make an assumption regarding the set of all English words. We assume that this set, denoted by T, is finite and fixed. From a practical point of view, this is obviously a reasonable assumption because we all commonly use a finite and fixed vocabulary of words in everyday English (purely hypothetically, however, this may not be the case as illustrated by the study that closes this section). Next, we subdivide this set into subsets with respect to the classification of verbs and pronouns described in Section 8.1:

- T is the set of all words including all their inflectional forms;
- $T_V \subset T$ is the set of all verbs including all their inflectional forms;
- $T_{VA} \subset T_V$ is the set of all auxiliary verbs including all their inflectional forms;
- $T_{Vpl} \subset T_V$ is the set of all verbs in plain form;
- $T_{PPn} \subset T$ is the set of all personal pronouns in nominative.

To describe all possible paradigms of a verb $v \in T_{Vpl}$, we use the following notation:

- $\pi_{3rd}(v)$ is the verb v in third person singular present;
- $\pi_{pres}(v)$ is the verb v in present (other than third person singular);
- $\pi_{pret}(v)$ is the verb v in preterite.

There are several conventions we use throughout this section in order to simplify the presented case studies:

- We do not take into account capitalization and punctuation. Therefore, according to this convention,

  He is your best friend.

  and

  he is your best friend

  are equivalent.
- To make the following studies as simple and readable as possible, we expect every input sentence to be a canonical clause. In some examples, however, we make slight exceptions to this rule; for instance, sometimes we permit the input sentence to be negative. The first example and the last example also demonstrate a simple type of coordinated canonical clauses.
- The input vocabulary is the set $I = \{\langle x \rangle : x \in T\}$, where T is the set of all English words as stated above. As a result, every transformational scattered context grammar in this section takes an input sentence over I and transforms

it to an output sentence over T. For instance, in the case of the declarative-to-interrogative transformation,

$$\langle \text{he} \rangle \langle \text{is} \rangle \langle \text{your} \rangle \langle \text{best} \rangle \langle \text{friend} \rangle$$

is transformed to

is he your best friend

As we have already mentioned, we omit punctuation and capitalization, so the above sentence corresponds to

Is he your best friend?

Next, we give several studies that describe how to transform various kinds of grammatical sentences to other grammatical sentences by using transformational scattered context grammars.

Clauses with *neither* and *nor*

The first example shows how to use transformational scattered context grammars to negate clauses that contain the pair of the words *neither* and *nor*, such as

Neither Thomas nor his wife went to the party.

Clearly, the words *neither* and *nor* are related, but there is no explicit limit of the number of the words appearing between them. The following transformational scattered context grammar G converts the above sentence to

Both Thomas and his wife went to the party.

In fact, the constructed grammar G is general enough to negate every grammatical clause that contains the pair of the words *neither* and *nor*.

Set $G = (V, T, P, I)$, where $V = T \cup I$, and P is defined as follows:

$$\begin{aligned} P = & \Big\{ \big(\langle \text{neither} \rangle, \langle \text{nor} \rangle \big) \rightarrow (\text{both}, \text{and}) \Big\} \\ \cup & \Big\{ \big(\langle x \rangle \big) \rightarrow (x) : x \in T - \{\text{neither, nor}\} \Big\}. \end{aligned}$$

For example, for the above sentence, the transformation can proceed in this way:

$$\begin{aligned} & \langle \text{neither} \rangle \langle \text{thomas} \rangle \langle \text{nor} \rangle \langle \text{his} \rangle \langle \text{wife} \rangle \langle \text{went} \rangle \langle \text{to} \rangle \langle \text{the} \rangle \langle \text{party} \rangle \\ \Rightarrow_G \; & \text{both } \langle \text{thomas} \rangle \text{ and } \langle \text{his} \rangle \langle \text{wife} \rangle \langle \text{went} \rangle \langle \text{to} \rangle \langle \text{the} \rangle \langle \text{party} \rangle \\ \Rightarrow_G \; & \text{both thomas and } \langle \text{his} \rangle \langle \text{wife} \rangle \langle \text{went} \rangle \langle \text{to} \rangle \langle \text{the} \rangle \langle \text{party} \rangle \\ \Rightarrow_G \; & \text{both thomas and his } \langle \text{wife} \rangle \langle \text{went} \rangle \langle \text{to} \rangle \langle \text{the} \rangle \langle \text{party} \rangle \\ \Rightarrow_G^5 \; & \text{both thomas and his wife went to the party.} \end{aligned}$$

The production

$$\big(\langle \text{neither} \rangle, \langle \text{nor} \rangle \big) \rightarrow (\text{both}, \text{and})$$

replaces *neither* and *nor* with *both* and *and*, respectively. Every other word $\langle w \rangle \in I$ is changed to $w \in T$. Therefore, if we denote all possible input sentences, described in the introduction of this example, by K, $T(G,K)$ represents the set of all negated sentences from K, and

$$\big(\langle\text{neither}\rangle\langle\text{thomas}\rangle\langle\text{nor}\rangle\langle\text{his}\rangle\langle\text{wife}\rangle\langle\text{went}\rangle\langle\text{to}\rangle\langle\text{the}\rangle\langle\text{party}\rangle, \text{both thomas and his wife went to the party}\big) \in T(G,K).$$

Existential clauses

In English, clauses that indicate an existence are called *existential*. These clauses are usually formed by the dummy subject *there*; for example,

There was a nurse present.

However, this dummy subject is not mandatory in all situations. For instance, the above example can be rephrased as

A nurse was present.

We construct a transformational scattered context grammar G that converts any canonical existential clause without the dummy subject *there* to an equivalent existential clause with *there*.

Set $G = (V,T,P,I)$, where $V = T \cup I \cup \{X\}$ (X is a new symbol such that $X \notin T \cup I$), and P is defined as follows:

$$\begin{aligned} P = \Big\{ & \big(\langle x\rangle, \langle\text{is}\rangle\big) \rightarrow (\text{there is } xX, \varepsilon), \\ & \big(\langle x\rangle, \langle\text{are}\rangle\big) \rightarrow (\text{there are } xX, \varepsilon), \\ & \big(\langle x\rangle, \langle\text{was}\rangle\big) \rightarrow (\text{there was } xX, \varepsilon), \\ & \big(\langle x\rangle, \langle\text{were}\rangle\big) \rightarrow (\text{there were } xX, \varepsilon) : x \in T \Big\} \\ \cup & \Big\{ \big(X, \langle x\rangle\big) \rightarrow (X, x) : x \in T \Big\} \\ \cup & \big\{ (X) \rightarrow (\varepsilon) \big\}. \end{aligned}$$

For the above sample sentence, we get the following derivation:

$$\begin{aligned} & \langle\text{a}\rangle\langle\text{nurse}\rangle\langle\text{was}\rangle\langle\text{present}\rangle \\ \Rightarrow_G\ & \text{there was a } X\langle\text{nurse}\rangle\langle\text{present}\rangle \\ \Rightarrow_G\ & \text{there was a } X \text{ nurse } \langle\text{present}\rangle \\ \Rightarrow_G\ & \text{there was a } X \text{ nurse present} \\ \Rightarrow_G\ & \text{there was a nurse present.} \end{aligned}$$

A production from the first set has to be applied first because initially there is no symbol X in the sentential form and all other productions require X to be present in the sentential form. In our case, the production

$$\big(\langle\text{a}\rangle, \langle\text{was}\rangle\big) \rightarrow (\text{there was a } X, \varepsilon)$$

is applied; the use of other productions from this set depends on what tense is used in the input sentence and whether the subject is in singular or plural. The production non-deterministically selects the first word of the sentence, puts *there was* in front of it, and the symbol X behind it; in addition, it erases *was* in the middle of the sentence. Next, all words $\langle w \rangle \in I$ are replaced with $w \in T$ by productions from the second set. These productions also verify that the previous non-deterministic selection was made at the beginning of the sentence; if not, there remains a word $\langle w \rangle \in I$ in front of X that cannot be rewritten. Finally, the derivation ends by erasing X from the sentential form.

This form of the derivation implies that if we denote the input existential clauses described in the introduction of this example by K, $T(G,K)$ represents the set of these clauses with the dummy subject *there*. As a result,

$$\big(\langle \text{a} \rangle \langle \text{nurse} \rangle \langle \text{was} \rangle \langle \text{present} \rangle, \text{there was a nurse present}\big) \in T(G,K).$$

Interrogative Clauses

In English, there are two ways of transforming declarative clauses into interrogative clauses depending on the predicator. If the predicator is an auxiliary verb, the interrogative clause is formed simply by swapping the subject and the predicator. For example, we get the interrogative clause

Is he mowing the lawn?

by swapping *he*, which is the subject, and *is*, which is the predicator, in

He is mowing the lawn.

On the other hand, if the predicator is a lexical verb, the interrogative clause is formed by adding the dummy *do* to the beginning of the declarative clause. The dummy *do* has to be of the same paradigm as the predicator in the declarative clause and the predicator itself is converted to its plain form. For instance,

She usually gets up early.

is a declarative clause with the predicator *gets*, which is in third person singular, and the subject *she*. By inserting *do* in third person singular to the beginning of the sentence and converting *gets* to its plain form, we obtain

Does she usually get up early?

To simplify the following transformational scattered context grammar G, which performs this conversion, we assume that the subject is a personal pronoun in nominative.

Set $G = (V, T, P, I)$, where $V = T \cup I \cup \{X\}$ (X is a new symbol such that $X \notin T \cup I$), and P is defined as follows:

$$\begin{aligned} P = & \Big\{ \big(\langle p \rangle, \langle v \rangle\big) \rightarrow (vp, X) : v \in T_{\mathrm{VA}}, p \in T_{PPn} \Big\} \\ \cup & \Big\{ \big(\langle p \rangle, \langle \pi_{pret}(v) \rangle\big) \rightarrow (\text{did } p, vX), \\ & \big(\langle p \rangle, \langle \pi_{3rd}(v) \rangle\big) \rightarrow (\text{does } p, vX), \\ & \big(\langle p \rangle, \langle \pi_{pres}(v) \rangle\big) \rightarrow (\text{do } p, vX) : v \in T_{Vpl} - T_{\mathrm{VA}}, p \in T_{PPn} \Big\} \\ \cup & \Big\{ \big(\langle x \rangle, X\big) \rightarrow (x, X), \\ & \big(X, \langle y \rangle\big) \rightarrow (X, y) : x \in T - T_{\mathrm{V}}, y \in T \Big\} \\ \cup & \big\{ (X) \rightarrow (\varepsilon) \big\}. \end{aligned}$$

For sentences whose predicator is an auxiliary verb, the transformation made by G proceeds as follows:

$$\begin{aligned} & \langle \text{he} \rangle \langle \text{is} \rangle \langle \text{mowing} \rangle \langle \text{the} \rangle \langle \text{lawn} \rangle \\ \Rightarrow_G\; & \text{is he } X \langle \text{mowing} \rangle \langle \text{the} \rangle \langle \text{lawn} \rangle \\ \Rightarrow_G\; & \text{is he } X \text{ mowing } \langle \text{the} \rangle \langle \text{lawn} \rangle \\ \Rightarrow_G\; & \text{is he } X \text{ mowing the } \langle \text{lawn} \rangle \\ \Rightarrow_G\; & \text{is he } X \text{ mowing the lawn} \\ \Rightarrow_G\; & \text{is he mowing the lawn.} \end{aligned}$$

The derivation starts by the application of a production from the first set, which swaps the subject and the predicator, and puts X behind them. Next, productions from the third set rewrite every word $\langle w \rangle \in I$ to $w \in T$. Finally, X is removed from the sentential form.

The transformation of the sentences in which the predicator is a lexical verb is more complicated:

$$\begin{aligned} & \langle \text{she} \rangle \langle \text{usually} \rangle \langle \text{gets} \rangle \langle \text{up} \rangle \langle \text{early} \rangle \\ \Rightarrow_G\; & \text{does she } \langle \text{usually} \rangle \text{ get } X \langle \text{up} \rangle \langle \text{early} \rangle \\ \Rightarrow_G\; & \text{does she usually get } X \langle \text{up} \rangle \langle \text{early} \rangle \\ \Rightarrow_G\; & \text{does she usually get } X \text{ up } \langle \text{early} \rangle \\ \Rightarrow_G\; & \text{does she usually get } X \text{ up early} \\ \Rightarrow_G\; & \text{does she usually get up early.} \end{aligned}$$

As the predicator is in third person singular, a production from

$$\Big\{ \big(\langle p \rangle, \langle \pi_{3rd}(v) \rangle\big) \rightarrow (\text{does } p, vX) : v \in T_{Vpl} - T_{\mathrm{VA}}, p \in T_{PPn} \Big\}$$

is applied at the beginning of the derivation. It inserts *does* to the beginning of the sentence, converts the predicator *gets* to its plain form *get*, and puts X behind it.

Next, productions from

$$\Big\{\big(\langle x\rangle, X\big) \to (x, X) : x \in T - T_V\Big\}$$

rewrite every word $\langle w\rangle \in I$ appearing in front of the predicator to $w \in T$. Notice that they do not rewrite verbs—in this way, the grammar verifies that the first verb in the sentence was previously selected as the predicator. For instance, in the sentence

He has been working for hours.

has must be selected as the predicator; otherwise, the derivation is unsuccessful. Finally, the grammar rewrites all words behind X, and erases X in the last step as in the previous case.

Based on this intuitive explanation, we can see that the set of all input sentences K described in the introduction of this example is transformed by G to $T(G,K)$, which is the set of all interrogative sentences constructed from K. Therefore,

$$\big(\langle\text{he}\rangle\langle\text{is}\rangle\langle\text{mowing}\rangle\langle\text{the}\rangle\langle\text{lawn}\rangle, \text{is he mowing the lawn}\big) \in T(G,K),$$
$$\big(\langle\text{she}\rangle\langle\text{usually}\rangle\langle\text{gets}\rangle\langle\text{up}\rangle\langle\text{early}\rangle, \text{does she usually get up early}\big) \in T(G,K).$$

Question Tags

Question tags are special constructs that are primarily used in spoken language. They are used at the end of declarative clauses, and we customarily use them to ask for agreement or confirmation. For instance, in

Your sister is married, isn't she?

isn't she is a question tag, and we expect an answer stating that she is married. The polarity of question tags is always opposite to the polarity of the main clause—if the main clause is positive, the question tag is negative, and vice versa. If the predicator is an auxiliary verb, the question tag is formed by the same auxiliary verb. For lexical verbs, the question tag is made by using *do* as

He plays the violin, doesn't he?

There are some special cases that have to be taken into account. First, the verb *be* has to be treated separately because it has more paradigms than other verbs and the question tag for first person singular is irregular:

I am always right, aren't I?

Second, for the verb *have*, the question tag depends on whether it is used as an auxiliary verb, or a lexical verb. In the first case, *have* is used in the question tag as

He has been working hard, hasn't he?

in the latter case, the auxiliary *do* is used as

They have a dog, don't they?

To explain the basic concepts as simply as possible, we omit the special cases of the verb *have* in the following transformational scattered context grammar G, which supplements a canonical clause with a question tag. For the same reason, we only sketch its construction and do not mention all the created productions explicitly. In addition, we suppose that the subject is represented by a personal pronoun.

Set $G = (V, T, P, I)$, where $V = T \cup I \cup \{X, Y\}$ (X, Y are new symbols such that X, $Y \notin T \cup I$), and P is defined as follows:

$$
\begin{aligned}
P = \Big\{ & \big(\langle p\rangle, \langle \text{will}\rangle, \langle x\rangle\big) \to (p, \text{will } X, Yx \text{ won't } p), \\
& \big(\langle p\rangle, \langle \text{won't}\rangle, \langle x\rangle\big) \to (p, \text{won't } X, Yx \text{ will } p), \\
& \ldots : p \in T_{PPn}, x \in T \Big\} \\
\cup \Big\{ & \big(\langle \text{I}\rangle, \langle \text{am}\rangle, \langle x\rangle\big) \to (\text{I}, \text{am } X, Yx \text{ aren't I}), \\
& \big(\langle \text{you}\rangle, \langle \text{are}\rangle, \langle x\rangle\big) \to (\text{you}, \text{are } X, Yx \text{ aren't you}), \\
& \ldots : x \in T \Big\} \\
\cup \Big\{ & \big(\langle p\rangle, \langle v\rangle, \langle x\rangle\big) \to (p, vX, Yx \text{ doesn't } p), \\
& \big(\langle q\rangle, \langle v\rangle, \langle x\rangle\big) \to (q, vX, Yx \text{ don't } q) : \\
& p \in \{\text{he, she, it}\}, q \in T_{PPn} - \{\text{he, she, it}\}, v \in T_V - T_{VA}, x \in T \Big\} \\
& \vdots \\
\cup \Big\{ & \big(\langle x\rangle, X\big) \to (x, X), \\
& \big(X, \langle y\rangle, Y\big) \to (X, y, Y) : x \in T - T_V, y \in T \Big\} \\
\cup \Big\{ & (X, Y) \to (\varepsilon, \varepsilon) \Big\}.
\end{aligned}
$$

First, we describe the generation of question tags for clauses whose predicator is an auxiliary verb:

$$
\begin{aligned}
& \langle \text{I}\rangle\langle \text{am}\rangle\langle \text{always}\rangle\langle \text{right}\rangle \\
\Rightarrow_G\ & \text{I am } X\langle \text{always}\rangle Y \text{ right aren't I} \\
\Rightarrow_G\ & \text{I am } X \text{ always } Y \text{ right aren't I} \\
\Rightarrow_G\ & \text{I am always right aren't I.}
\end{aligned}
$$

Here, the production

$$\big(\langle \text{I}\rangle, \langle \text{am}\rangle, \langle \text{right}\rangle\big) \to (\text{I}, \text{am } X, Y \text{ right aren't I})$$

initiates the derivation. When it finds *I am* at the beginning of the sentence, it generates the question tag *aren't I* at its end. In addition, it adds X behind *I am* and Y in front of *right aren't I*. Next, it rewrites all words from $\langle w \rangle \in I$ to $w \in T$. It makes sure that the predicator was chosen properly by productions from

$$\Big\{ \big(\langle x \rangle, X \big) \rightarrow (x, X) : x \in T - T_V \Big\}$$

similarly to the previous example. In addition, productions from

$$\Big\{ \big(X, \langle y \rangle, Y \big) \rightarrow (X, y, Y) : x \in T - T_V, y \in T \Big\}$$

check whether the question tag was placed at the very end of the sentence. If not, there remains some symbol from the input vocabulary behind Y that cannot be rewritten. Finally, the last production removes X and Y from the sentential form.

When the predicator is a lexical verb in present, the question tag is formed by *does* or *do* depending on person in which the predicator occurs:

$$\begin{aligned} & \langle \text{he} \rangle \langle \text{plays} \rangle \langle \text{the} \rangle \langle \text{violin} \rangle \\ \Rightarrow_G\ & \text{he plays } X \langle \text{the} \rangle Y \text{ violin doesn't he} \\ \Rightarrow_G\ & \text{he plays } X \text{ the violin } Y \text{ doesn't he} \\ \Rightarrow_G\ & \text{he plays the violin doesn't he.} \end{aligned}$$

The rest of the derivation is analogous to the first case.

Based on these derivations, we can see that the set of all input sentences K described in the introduction of this example is transformed by G to $T(G, K)$, which is the set of all sentences constructed from K that are supplemented with question tags. Therefore,

$$\big(\langle \text{I} \rangle \langle \text{am} \rangle \langle \text{always} \rangle \langle \text{right} \rangle, \text{I am always right aren't I} \big) \in T(G, K),$$
$$\big(\langle \text{he} \rangle \langle \text{plays} \rangle \langle \text{the} \rangle \langle \text{violin} \rangle, \text{he plays the violin doesn't he} \big) \in T(G, K).$$

Generation of Grammatical Sentences

The purpose of the next discussion, which closes this section, is six-fold—(1) through (6), stated below.

1. We want to demonstrate that ordinary scattered context grammars, discussed in the previous chapters of this book, can be seen as a special case of transformational scattered context grammars, whose applications are discussed in the present section.

2. As pointed out in the notes following the general definition of a transformational scattered context grammar (see Definition 8.1), there exists a close relation between ordinary scattered context grammars and transformational scattered context grammars. That is, for every scattered context grammar

$G = (V, T, P, S)$, there is a transformational scattered context grammar $H = \big(V, T, P, \{S\}\big)$ satisfying

$$L(G) = \Big\{y : (S, y) \in T\big(H, \{S\}\big)\Big\},$$

and in this way, $L(G)$ is defined by H. Next, we illustrate this relation by a specific example.

3. From a syntactical point of view, we want to show that scattered context grammars can generate an infinite non-context-free grammatical subset of English language in a very succinct way.

4. In terms of *morphology*—that is, the area of linguistics that studies the structure of words and their generation—we demonstrate how to use transformational scattered context grammars to create complicated English words within English sentences so that the resulting words and sentences are grammatically correct.

5. As stated in the beginning of the present section, so far we have assumed that the set of common English words is finite. Next, we want to demonstrate that from a strictly theoretical point of view, the set of all possible well-formed English words, including extremely rare words in everyday English, is infinite. Indeed, L, given next, includes infinitely many words of the form $(\textit{great-})^i\textit{grandparents}$, $(\textit{great-})^i\textit{grandfathers}$, and $(\textit{great-})^i\textit{grandmothers}$, for all $i \geq 0$, and purely theoretically speaking, they all represent well-formed English words. Of course, most of them, such as

   great-great-great-great-great-great-great-great-great-grandfathers

   cannot be considered as common English words because most people never use them during their lifetime.

6. We illustrate that the language generation based upon scattered context grammars may have significant advantages over the generation based upon classical grammars, such as context-sensitive grammars.

Without further ado, consider the language L consisting of these grammatical English sentences:

Your grandparents are all your grandfathers and all your grandmothers.

Your great-grandparents are all your great-grandfathers and all your great-grandmothers.

Your great-great-grandparents are all your great-great-grandfathers and all your great-great-grandmothers.

⋮

In brief,

$$L = \big\{\text{your } \{\text{great-}\}^i\text{grandparents are all your } \{\text{great-}\}^i\text{grandfathers}$$
$$\text{and all your } \{\text{great-}\}^i\text{grandmothers } : i \geq 0\big\}.$$

Introduce the scattered context grammar $G = (V, T, P, S)$, where

$$T = \{\text{all}, \text{and}, \text{are}, \text{grandfathers}, \text{grandmothers}, \text{grandparents}, \text{great-}, \text{your}\},$$

$V = T \cup \{S, \#\}$, and P consists of these three productions:

$$(S) \to (\text{your \#grandparents are all your \#grandfathers}$$
$$\text{and all your \#grandmothers}),$$
$$(\#, \#, \#) \to (\#\text{great-}, \#\text{great-}, \#\text{great-}),$$
$$(\#, \#, \#) \to (\varepsilon, \varepsilon, \varepsilon).$$

Obviously, this scattered context grammar generates L; formally, $L = L(G)$. Consider the transformational scattered context grammar $H = \big(V, T, P, \{S\}\big)$. Notice that

$$L(G) = \Big\{y : (S, y) \in T\big(H, \{S\}\big)\Big\}.$$

Clearly, L is not context-free, so its generation is beyond the power of context-free grammars. It would be possible to construct a context-sensitive grammar that generates L. However, a context-sensitive grammar like this would have to keep traversing across its sentential forms to guarantee the same number of occurrences of *great-* in the generated sentences. Compared to this awkward way of generating L, the scattered context grammar G generates L in a more elegant, economical, and effective way.

In this chapter, we have illustrated how to transform and generate grammatical sentences in English by using transformational scattered context grammars, which represent a very natural linguistic apparatus straightforwardly based on scattered context grammars. However, from a more general perspective, we can apply these grammars basically in any area of science that formalizes its results by strings containing some scattered context dependencies. This general perspective brings us to the concluding chapter of this book, in which we make remarks about some selected scientific areas that involve a formalization of scattered context, and we also suggest how to make use of scattered context grammars in them.