Chapter 8

Properties of Context-Free Languages

The family of context-free languages occupies a central position in a hierarchy of formal languages. On the one hand, context-free languages include important but restricted language families such as regular and deterministic context-free languages. On the other hand, there are broader language families of which context-free languages are a special case. To study the relationship between language families and to exhibit their similarities and differences, we investigate characteristic properties of the various families. As in Chapter 4, we look at closure under a variety of operations, algorithms for determining properties of members of the family, and structural results such as pumping lemmas. These all provide us with a means of understanding relations between the different families as well as for classifying specific languages in an appropriate category.

8.1 Two Pumping Lemmas

The pumping lemma given in Theorem 4.8 is an effective tool for showing that certain languages are not regular. Similar pumping lemmas are known for other language families. Here we will discuss two such results, one for context-free languages in general, the other for a restricted type of context-free language.

A Pumping Lemma for Context-Free Languages

Theorem 8.1

Let L be an infinite context-free language. Then there exists some positive integer m such that any $w \in L$ with $|w| \geq m$ can be decomposed as

$$\boxed{w = u\underbrace{vxy}z,} \tag{8.1}$$

with

$$\boxed{|vxy| \leq m,} \tag{8.2}$$

and

$$|vy| \geq 1, \tag{8.3}$$

such that

$$uv^ixy^iz \in L, \tag{8.4}$$

for all $i = 0, 1, 2, ...$ This is known as the pumping lemma for context-free languages.

Proof: Consider the language $L - \{\lambda\}$, and assume that we have for it a grammar G without unit-productions or λ-productions. Since the length of the string on the right side of any production is bounded, say by k, the length of the derivation of any $w \in L$ must be at least $|w|/k$. Therefore, since L is infinite, there exist arbitrarily long derivations and corresponding derivation trees of arbitrary height.

Consider now such a high derivation tree and some sufficiently long path from the root to a leaf. Since the number of variables in G is finite, there must be some variable that repeats on this path, as shown schematically in Figure 8.1. Corresponding to the derivation tree in Figure 8.1, we have the derivation

$$S \overset{*}{\Rightarrow} uAz \overset{*}{\Rightarrow} uvAyz \overset{*}{\Rightarrow} uvxyz,$$

where u, v, x, y, and z are all strings of terminals. From the above we see that $A \overset{*}{\Rightarrow} vAy$ and $A \overset{*}{\Rightarrow} x$, so all the strings uv^ixy^iz, $i = 0, 1, 2, ...$, can be generated by the grammar and are therefore in L. Furthermore, in the

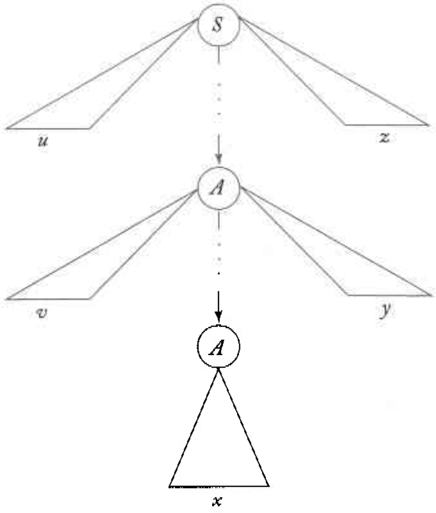


Figure 8.1 Derivation tree for a long string.

derivations $A \overset{*}{\Rightarrow} vAy$ and $A \overset{*}{\Rightarrow} x$, we can assume that no variable repeats (otherwise, we just use the repeating variable as A). Therefore, the lengths of the strings v, x, and y depend only on the productions of the grammar and can be bounded independently of w so that (8.2) holds. Finally, since there are no unit productions and no λ productions, v and y cannot both be empty strings, giving (8.3).

This completes the argument that (8.1) to (8.4) hold. ■

This pumping lemma is useful in showing that a language does not belong to the family of context-free languages. Its application is typical of pumping lemmas in general; they are used negatively to show that a given language does not belong to some family. As in Theorem 4.8, the correct argument can be visualized as a game against an intelligent opponent. But now the rules make it a little more difficult for us. For regular languages, the substring xy whose length is bounded by m starts at the left end of w. Therefore the substring y that can be pumped is within m symbols of the beginning of w. For context-free languages, we only have a bound on $|vxy|$. The substring u that precedes vxy can be arbitrarily long. This gives additional freedom to the adversary, making arguments involving Theorem 8.1 a little more complicated.

Example 8.1 Show that the language

$$L = \{a^n b^n c^n : n \geq 0\}$$

is not context-free.

Once the adversary has chosen m, we pick the string $\boxed{a^m b^m c^m}$ which is in L. The adversary now has several choices. If he chooses vxy to contain only a's, then the pumped string will obviously not be in L. If he chooses a string containing an equal number of a's and b's, then the pumped string $\boxed{a^k b^k c^m}$ with $k \neq m$ can be generated, and again we have generated a string not in L. In fact, the only way the adversary could stop us from winning is to pick vxy so that vy has the same number of a's, b's, and c's. But this is not possible because of restriction (8.2). Therefore, L is not context-free.

If we try the same argument on the language $L = \{a^n b^n\}$, we fail, as we must, since the language is context-free. If we pick any string in L, such as $w = a^m b^m$, the adversary can pick $v = a^k$ and $y = b^k$. Now, no matter what i we choose, the resulting pumped string w_i is in L. Remember, though, that this does not prove that L is context-free; all we can say is that we have been unable to get any conclusion from the pumping lemma. That L is context-free must come from some other argument, such as the construction of a context-free grammar.

$L = \{a^n b^n\}$

نمی‌توانیم بگوییم L C.f است

فقط می‌توانیم بگوییم از lemma

نتیجه‌ای نگرفتیم

The argument also justifies a claim made in Example 7.9 and allows us to close a gap in that example. The language

$$\boxed{\widehat{L} = \{a^n b^n\} \cup \{a^n b^{2n}\} \cup \{a^n b^n c^n\}}$$

is not context-free. The string $a^m b^m c^m$ is in $\widehat{L}$, but the pumped result is not. ■

Example 8.2

Consider the language

$$\boxed{L = \{ww : w \in \{a, b\}^*\}}.$$

Although this language appears to be very similar to the context-free language of Example 5.1, it is not context-free.

Consider the string

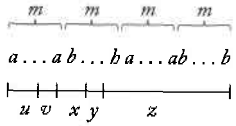
$$a^m b^m a^m b^m.$$

There are many ways in which the adversary can now pick vxy, but for all of them we have a winning countermove. For example, for the choice in Figure 8.2, we can use $i = 0$ to get a string of the form

$$a^k b^j a^m b^m, k < m \text{ or } j < m,$$

which is not in L. For other choices by the adversary, similar arguments can be made. We conclude that L is not context-free. ■

Figure 8.2



Example 8.3 Show that the language

$$L = \{a^{n!} : n \geq 0\}$$

→ not regular

→ not c.f

is not context-free.

In Example 4.11 we showed that this language is not regular. However, for a language over an alphabet with a single symbol, there is little difference between Theorem 8.1 and the pumping lemma for regular languages. In either case, the strings to be pumped consist entirely of a's, and whatever new string can be generated by Theorem 8.1 can also be generated by Theorem 4.8. Therefore, we can use essentially the same arguments as in Example 4.11 to show that L is not context-free. ■

$n = j^2 \Rightarrow$ aa b

aaaa bb

aaaaaaaaa bbb

Example 8.4 Show that the language

$$L = \{a^n b^j : n = j^2\}$$

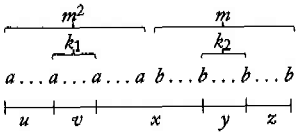
is not context-free.

Given m in Theorem 8.1, we pick as our string $a^{m^2} b^m$. The adversary now has several choices. The only one that requires much thought is the one shown in Figure 8.3. Pumping i times will yield a new string with $m^2 + (i-1)\, k_1$ a's and $m + (i-1)\, k_2$ b's. If the adversary takes $k_1 \neq 0$, $k_2 \neq 0$, we can pick $i = 0$. Since

$$\begin{aligned}(m - k_2)^2 &\leq (m-1)^2 \\ &= m^2 - 2m + 1 \\ &< m^2 - k_1,\end{aligned}$$

the result is not in L. If the opponent picks $k_1 = 0$, $k_2 \neq 0$ or $k_1 \neq 0$, $k_2 = 0$, then again with $i = 0$, the pumped string is not in L. We can conclude from this that L is not a context-free language. ■

Figure 8.3


m^2
m
k_1
k_2
$a \ldots a \ldots a \ldots a\ b \ldots b \ldots b \ldots b$
u
v
x
y
z

A Pumping Lemma for Linear Languages

We previously made a distinction between linear and nonlinear context-free grammars. We now make a similar distinction between languages.

Definition 8.1

A context-free language L is said to be linear if there exists a linear context-free grammar G such that $L = L(G)$.

هر زبان خطی یک C.f است

Clearly, every linear language is context-free, but we have not yet established whether or not the converse is true.

Example 8.5 The language $L = \{a^n b^n : n \geq 0\}$ is a linear language. A linear grammar for it is given in Example 1.10. The grammar given in Example 1.12 for the language $L = \{w : n_a(w) = n_b(w)\}$ is not linear, so the second language is not necessarily linear. ■

Of course, just because a specific grammar is not linear does not imply that the language generated by it is not linear. If we want to prove that a language is not linear, we must show that there exists no equivalent linear grammar. We approach this in the usual way, establishing structural properties for linear languages, then showing that some context-free languages do not have a required property.

Theorem 8.2 Let L be an infinite linear language. Then there exists some positive integer m, such that any $w \in L$, with $|w| \geq m$ can be decomposed as $w = uvxyz$ with

$$|uvyz| \leq m, \tag{8.5}$$

$$|vy| \geq 1, \tag{8.6}$$

such that

$$uv^ixy^iz \in L, \tag{8.7}$$

for all $i = 0, 1, 2, ...$

Note that the conclusions of this theorem differ from those of Theorem 8.1, since (8.2) is replaced by (8.5). This implies that the strings v and y to be pumped must now be located within m symbols of the left and right ends of w, respectively. The middle string x can be of arbitrary length.

Proof: Our reasoning follows the proof of Theorem 8.1. Since the language is linear, there exists some linear grammar G for it. To use the argument in Theorem 8.1, we also need to claim that G contains no unit-productions and no λ-productions. An examination of the proofs of Theorem 6.3 and Theorem 6.4 will show that removing λ-productions and unit-productions does not destroy the linearity of the grammar. We can therefore assume that G has the required property.

Consider now the derivation tree as shown in Figure 8.1. Because the grammar is linear, variables can appear only on the path from S to the first A, on the path from the first A to the second one, and on the path from the second A to some leaf of the tree. Since there are only a finite number of variables on the path from S to the first A, and since each of these generates a finite number of terminals, u and z must be bounded. By a similar argument, v and y are bounded, so (8.5) follows.

The rest of the argument is as in Theorem 8.1. ■

Example 8.6 The language

$$\boxed{L = \{w : n_a(w) = n_b(w)\}}$$

is not linear.

To show this, assume that the language is linear and apply Theorem 8.2 to the string

$$w = a^m b^{2m} a^m.$$

Inequality (8.7) shows that in this case the strings u, v, y, z must all consist entirely of a's. If we pump this string, we get $a^{m+k}b^{2m}a^{m+l}$, with either $k \geq 1$ or $l \geq 1$, a result that is not in L. This contradiction of Theorem 8.2 proves that the language is not linear. ■

This example answers the general question raised on the relation between the families of context-free and linear languages. The family of linear languages is a proper subset of the family of context-free languages.

EXERCISES

1. Use reasoning similar to that in Example 4.11 to give a complete proof that the language in Example 8.3 is not context-free.

2. Show that the language $L = \{a^n : n \text{ is a prime number}\}$ is not context-free.

3. Show that $L = \{ww^Rw : w \in \{a,b\}^*\}$ is not a context-free language. ●

4. Show that $L = \{w \in \{a,b,c\}^* : n_a^2(w) + n_b^2(w) = n_c^2(w)\}$ is not context-free.

5. Is the language $L = \{a^n b^m : n = 2^m\}$ context-free?

6. Show that the language $L = \left\{a^{n^2} : n \geq 0\right\}$ is not context-free.

4 7. Show that the following languages on $\Sigma = \{a,b,c\}$ are not context-free.

   (a) $L = \{a^n b^j : n \leq j^2\}$ ●

   (b) $L = \{a^n b^j : n \geq (j-1)^3\}$

   (c) $L = \{a^n b^j c^k : k = jn\}$

   (d) $L = \{a^n b^j c^k : k > n, k > j\}$

   (e) $L = \{a^n b^j c^k : n < j, n \leq k \leq j\}$

   (f) $L = \{w : n_a(w) < n_b(w) < n_c(w)\}$ (S)

   (g) $L = \{w : n_a(w)/n_b(w) = n_c(w)\}$

   (h) $L = \{w \in \{a,b,c\}^* : n_a(w) + n_b(w) = 2n_c(w)\}$.

7 8. Determine whether or not the following languages are context-free.

   (a) $L = \{a^n ww^R a^n : n \geq 0, w \in \{a,b\}^*\}$

   (b) $L = \{a^n b^j a^n b^j : n \geq 0, j \geq 0\}$ ●

   (c) $L = \{a^n b^j a^j b^n : n \geq 0, j \geq 0\}$

   (d) $L = \{a^n b^j a^k b^l : n + j \leq k + l\}$

   (e) $L = \{a^n b^j a^k b^l : n \leq k, j \leq l\}$

   (f) $L = \{a^n b^n c^j : n \leq j\}$

9. In Theorem 8.1, find a bound for m in terms of the properties of the grammar G.

X 10. Determine whether or not the following language is context-free.

$$L = \{w_1 c w_2 : w_1, w_2 \in \{a,b\}^*, w_1 \neq w_2\}$$ ●

8 **11.** Show that the language $L = \{a^n b^n a^m b^m : n \geq 0, m \geq 0\}$ is context-free but not linear.

9 **12.** Show that the following language is not linear.

$$L = \{w : n_a(w) \geq n_b(w)\} \quad ●$$

13. Show that the language $L = \{w \in \{a, b, c\}^* : n_a(w) + n_b(w) = n_c(w)\}$ is context-free, but not linear.

10 **14.** Determine whether or not the language $L = \{a^n b^j : j \leq n \leq 2j - 1\}$ is linear.

15. Determine whether or not the language in Example 5.12 is linear. ●

16. In Theorem 8.2, find a bound on m in terms of the properties of the grammar G.

17. Justify the claim made in Theorem 8.2 that for any linear language (not containing λ) there exists a linear grammar without λ-productions and unit-productions.

18. Consider the set of all strings a/b, where a and b are positive decimal integers such that $a < b$. The set of strings then represents all possible decimal fractions. Determine whether or not this is a context-free language.

★ **19.** Show that the complement of the language in Exercise 6 is not context-free.

16 **20.** Is the following language context-free?

$$L = \{a^{nm} : n \text{ and } m \text{ are prime numbers}\} \quad ●$$

8.2 Closure Properties and Decision Algorithms for Context-Free Languages

In Chapter 4 we looked at closure under certain operations and algorithms to decide on the properties of the family of regular languages. On the whole, the questions raised there had easy answers. When we ask the same questions about context-free languages, we encounter more difficulties. First, closure properties that hold for regular languages do not always hold for context-free languages. When they do, the arguments needed to prove them are often quite complicated. Second, many intuitively simple and important questions about context-free languages cannot be answered. This statement may seem at first surprising and will need to be elaborated as we proceed. In this section, we provide only a sample of some of the most important results.

Closure of Context-Free Languages

Theorem 8.3

The family of context-free languages is closed under union, concatenation, and star-closure.

Proof: Let L_1 and L_2 be two context-free languages generated by the context-free grammars $G_1 = (V_1, T_1, S_1, P_1)$ and $G_2 = (V_2, T_2, S_2, P_2)$, respectively. We can assume without loss of generality that the sets V_1 and V_2 are disjoint.

Consider now the language $L(G_3)$, generated by the grammar

$$G_3 = (V_1 \cup V_2 \cup \{S_3\}, T_1 \cup T_2, S_3, P_3),$$

where S_3 is a variable not in $V_1 \cup V_2$. The productions of G_3 are all the productions of G_1 and G_2, together with an alternative starting production that allows us to use one or the other grammars. More precisely,

$$P_3 = P_1 \cup P_2 \cup \{S_3 \to S_1 | S_2\}.$$

Obviously, G_3 is a context-free grammar, so that $L(G_3)$ is a context-free language. But it is easy to see that

$$L(G_3) = L_1 \cup L_2. \tag{8.8}$$

Suppose for instance that $w \in L_1$. Then

$$S_3 \Rightarrow S_1 \overset{*}{\Rightarrow} w$$

is a possible derivation in grammar G_3. A similar argument can be made for $w \in L_2$. Also, if $w \in L(G_3)$ then either

$$S_3 \Rightarrow S_1 \tag{8.9}$$

or

$$S_3 \Rightarrow S_2 \tag{8.10}$$

must be the first step of the derivation. Suppose (8.9) is used. Since sentential forms derived from S_1 have variables in V_1, and V_1 and V_2 are disjoint, the derivation

$$S_1 \overset{*}{\Rightarrow} w$$

can involve productions in P_1 only. Hence w must be in L_1. Alternatively, if (8.10) is used first, then w must be in L_2 and it follows that $L(G_3)$ is the union of L_1 and L_2.

Next, consider

$$G_4 = (V_1 \cup V_2 \cup \{S_4\}, T_1 \cup T_2, S_4, P_4).$$

Here again S_4 is a new variable and

$$P_4 = P_1 \cup P_2 \cup \{S_4 \to S_1 S_2\}.$$

Then

$$L(G_4) = L(G_1)\,L(G_2)$$

follows easily.

Finally, consider $L(G_5)$ with

$$G_5 = (V_1 \cup \{S_5\}, T_1, S_5, P_5),$$

where S_5 is a new variable and

$$P_5 = P_1 \cup \{S_5 \to S_1 S_5 | \lambda\}.$$

Then

$$L(G_5) = L(G_1)^*.$$

Thus we have shown that the family of context-free languages is closed under union, concatenation, and star-closure. ■

Theorem 8.4 The family of context-free languages is not closed under intersection and complementation.

Proof: Consider the two languages

$$L_1 = \{a^n b^n c^m : n \geq 0, m \geq 0\}$$

and

$$L_2 = \{a^n b^m c^m : n \geq 0, m \geq 0\}.$$

There are several ways one can show that L_1 and L_2 are context-free. For instance, a grammar for L_1 is

$$\begin{aligned} S &\to S_1 S_2, \\ S_1 &\to a S_1 b | \lambda, \\ S_2 &\to c S_2 | \lambda. \end{aligned}$$

Alternatively, we note that L_1 is the concatenation of two context-free languages, so it is context-free by Theorem 8.3. But

$$L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\},$$

which we have already shown not to be context-free. Thus, the family of context-free languages is not closed under intersection.

The second part of the theorem follows from Theorem 8.3 and the set identity

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}.$$

If the family of context-free languages were closed under complementation, then the right side of the above expression would be a context-free language for any context-free L_1 and L_2. But this contradicts what we have just shown, that the intersection of two context-free languages is not necessarily context-free. Consequently, the family of context-free languages is not closed under complementation. ■

While the intersection of two context-free languages may produce a language that is not context-free, the closure property holds if one of the languages is regular.

Theorem 8.5 Let L_1 be a context-free language and L_2 be a regular language. Then $L_1 \cap L_2$ is context-free.

Proof: Let $M_1 = (Q, \Sigma, \Gamma, \delta_1, q_0, z, F_1)$ be an npda which accepts L_1 and $M_2 = (P, \Sigma, \delta_2, p_0, F_2)$ be a dfa that accepts L_2. We construct a pushdown automaton $\widehat{M} = \left(\widehat{Q}, \Sigma, \Gamma, \widehat{\delta}, \widehat{q}_0, z, \widehat{F}\right)$ which simulates the parallel action of M_1 and M_2: whenever a symbol is read from the input string, $\widehat{M}$ simultaneously executes the moves of M_1 and M_2. To this end we let

$$\widehat{Q} = Q \times P,$$
$$\widehat{q}_0 = (q_0, p_0),$$
$$\widehat{F} = F_1 \times F_2,$$

and define $\widehat{\delta}$ such that

$$((q_k, p_l), x) \in \widehat{\delta}((q_i, p_j), a, b),$$

if and only if

$$(q_k, x) \in \delta_1(q_i, a, b),$$

and

$$\delta_2(p_j, a) = p_l.$$

In this, we also require that if $a = \lambda$, then $p_j = p_l$. In other words, the states of $\widehat{M}$ are labeled with pairs (q_i, p_j), representing the respective states in which M_1 and M_2 can be after reading a certain input string. It is a straightforward induction argument to show that

$$((q_0, p_0), w, z) \overset{*}{\vdash}_{\widehat{M}} ((q_r, p_s), x),$$

with $q_r \in F_1$ and $p_s \in F_2$ if and only if

$$(q_0, w, z) \overset{*}{\vdash}_{M_1} (q_r, x),$$

and

$$\delta^* (p_0, w) = p_s.$$

Therefore, a string is accepted by $\widehat{M}$ if and only if it is accepted by M_1 and M_2, that is, if it is in $L(M_1) \cap L(M_2) = L_1 \cap L_2$. ■

The property addressed by this theorem is called closure under **regular intersection**. Because of the result of the theorem, we say that the family of context-free languages is closed under regular intersection. This closure property is sometimes useful for simplifying arguments in connection with specific languages.

Example 8.7 Show that the language

$$L = \{a^n b^n : n \geq 0, n \neq 100\}$$

is context-free.

It is possible to prove this claim by constructing a pda or a context-free grammar for the language, but the process is tedious. We can get a much neater argument with Theorem 8.5.

Let

$$L_1 = \left\{a^{100} b^{100}\right\}.$$

Then, because L_1 is finite, it is regular. Also, it is easy to see that

$$L = \{a^n b^n : n \geq 0\} \cap \overline{L}_1.$$

Therefore, by the closure of regular languages under complementation and the closure of context-free languages under regular intersection, the desired result follows.

■

Example 8.8 Show that the language

$$L = \left\{ w \in \{a, b, c\}^* : n_a(w) = n_b(w) = n_c(w) \right\}$$

is not context-free.

The pumping lemma can be used for this, but again we can get a much shorter argument using closure under regular intersection. Suppose that L were context-free. Then

$$L \cap L(a^* b^* c^*) = \{a^n b^n c^n : n \geq 0\}$$

would also be context-free. But we already know that this is not so. We conclude that L is not context-free. ■

Closure properties of languages play an important role in the theory of formal languages and many more closure properties for context-free languages can be established. Some additional results are explored in the exercises at the end of this section.

Some Decidable Properties of Context-Free Languages

By putting together Theorems 5.2 and 6.6, we have already established the existence of a membership algorithm for context-free languages. This is of course an essential feature of any language family useful in practice. Other simple properties of context-free languages can also be determined. For the purpose of this discussion, we assume that the language is described by its grammar.

Theorem 8.6 Given a context-free grammar $G = (V, T, S, P)$, there exists an algorithm for deciding whether or not $L(G)$ is empty.

Proof: For simplicity, assume that $\lambda \notin L(G)$. Slight changes have to be made in the argument if this is not so. We use the algorithm for removing useless symbols and productions. If S is found to be useless, then $L(G)$ is empty; if not, then $L(G)$ contains at least one element. ■

Theorem 8.7 Given a context-free grammar $G = (V, T, S, P)$, there exists an algorithm for determining whether or not $L(G)$ is infinite.

Proof: We assume that G contains no λ-productions, no unit-productions, and no useless symbols. Suppose the grammar has a repeating variable in the sense that there exists some $A \in V$ for which there is a derivation

$$A \stackrel{*}{\Rightarrow} xAy.$$

Since G is assumed to have no λ-productions and no unit-productions, x and y cannot be simultaneously empty. Since A is neither nullable nor a useless symbol, we have

$$S \overset{*}{\Rightarrow} uAv \overset{*}{\Rightarrow} w$$

and

$$A \overset{*}{\Rightarrow} z,$$

where, u, v, and z are in T^*. But then

$$S \overset{*}{\Rightarrow} uAv \overset{*}{\Rightarrow} ux^nAy^nv \overset{*}{\Rightarrow} ux^nzy^nv$$

is possible for all n, so that $L(G)$ is infinite.

If no variable can ever repeat, then the length of any derivation is bounded by $|V|$. In that case, $L(G)$ is finite.

Thus, to get an algorithm for determining whether $L(G)$ is finite, we need only to determine whether the grammar has some repeating variables. This can be done simply by drawing a dependency graph for the variables in such a way that there is an edge (A, B) whenever there is a corresponding production

$$A \rightarrow xBy.$$

Then any variable that is at the base of a cycle is a repeating one. Consequently, the grammar has a repeating variable if and only if the dependency graph has a cycle.

Since we now have an algorithm for deciding whether a grammar has a repeating variable, we have an algorithm for determining whether or not $L(G)$ is infinite. ■

Somewhat surprisingly, other simple properties of context-free languages are not so easily dealt with. As in Theorem 4.7, we might look for an algorithm to determine whether two context-free grammars generate the same language. But it turns out that there is no such algorithm. For the moment, we do not have the technical machinery for properly defining the meaning of "there is no algorithm," but its intuitive meaning is clear. This is an important point to which we will return later.

EXERCISES

1. Is the complement of the language in Example 8.8 context-free?

2. Consider the language L_1 in Theorem 8.4. Show that this language is linear.

3. Show that the family of context-free languages is closed under homomorphism.

4. Show that the family of linear languages is closed under homomorphism.

5. Show that the family of context-free languages is closed under reversal. ●

6. Which of the language families we have discussed are not closed under reversal?

7. Show that the family of context-free languages is not closed under difference in general, but is closed under regular difference, that is, if L_1 is context-free and L_2 is regular, then $L_1 - L_2$ is context-free.

8. Show that the family of deterministic context-free languages is closed under regular difference.

9. Show that the family of linear languages is closed under union, but not closed under concatenation. ●

10. Show that the family of linear languages is not closed under intersection.

11. Show that the family of deterministic context-free languages is not closed under union and intersection.

12. Give an example of a context-free language whose complement is not context-free.

★13. Show that if L_1 is linear and L_2 is regular, then $L_1 L_2$ is a linear language. ●

14. Show that the family of unambiguous context-free languages is not closed under union.

15. Show that the family of unambiguous context-free languages is not closed under intersection. ●

16. Let L be a deterministic context-free language and define a new language $L_1 = \{w : aw \in L, a \in \Sigma\}$. Is it necessarily true that L_1 is a deterministic context-free language?

17. Show that the language $L = \{a^n b^n : n \geq 0, n \text{ is not a multiple of } 5\}$ is context-free.

18. Show that the following language is context-free.

$$L = \{w \in \{a,b\}^* : n_a(w) = n_b(w), w \text{ does not contain a substring } aab\}$$

19. Is the family of deterministic context-free languages closed under homomorphism?

20. Give the details of the inductive argument in Theorem 8.5.

21. Give an algorithm which, for any given context-free grammar G, can determine whether or not $\lambda \in L(G)$. ●

22. Show that there exists an algorithm to determine whether the language generated by some context-free grammar contains any words of length less than some given number n.

23. Let L_1 be a context-free language and L_2 be regular. Show that there exists an algorithm to determine whether or not L_1 and L_2 have a common element.