THEOREM 2.66

An endmarked language is generated by an LR(1) grammar iff it is a DCFL.

We've already shown that every DCFL has an LR(0) grammar, because an LR(0) grammar is the same as a DCFG. That proves the reverse direction of the theorem. What remains is the following lemma, which shows how to convert an LR(1) grammar to a DPDA.

LEMMA 2.67

Every LR(1) grammar has an equivalent DPDA.

PROOF IDEA We construct P_1, a modified version of the DPDA P that we presented in Lemma 2.67. P_1 reads its input and simulates DK_1, while using the stack to keep track of the state DK_1 would be in if all reduce steps were applied to this input up to this point. Moreover, P_1 reads 1 symbol ahead and stores this lookahead information in its finite state memory. Whenever DK_1 reaches an accept state, P_1 consults its lookahead to see whether to perform a reduce step, and which step to do if several possibilities appear in this state. Only one option can apply because the grammar is LR(1).

EXERCISES

2.1 Recall the CFG G_4 that we gave in Example 2.4. For convenience, let's rename its variables with single letters as follows.

$$E \to E+T \mid T$$
$$T \to T \times F \mid F$$
$$F \to (E) \mid \texttt{a}$$

Give parse trees and derivations for each string.

a. a

b. a+a

c. a+a+a

d. ((a))

2.2 a. Use the languages $A = \{\texttt{a}^m\texttt{b}^n\texttt{c}^n \mid m,n \geq 0\}$ and $B = \{\texttt{a}^n\texttt{b}^n\texttt{c}^m \mid m,n \geq 0\}$ together with Example 2.36 to show that the class of context-free languages is not closed under intersection.

b. Use part (a) and DeMorgan's law (Theorem 0.20) to show that the class of context-free languages is not closed under complementation.

[A]2.3 Answer each part for the following context-free grammar G.

$$R \to XRX \mid S$$
$$S \to \texttt{a}T\texttt{b} \mid \texttt{b}T\texttt{a}$$
$$T \to XTX \mid X \mid \varepsilon$$
$$X \to \texttt{a} \mid \texttt{b}$$

a. What are the variables of G?

b. What are the terminals of G?

c. Which is the start variable of G?

d. Give three strings in $L(G)$.

e. Give three strings *not* in $L(G)$.

f. True or False: $T \Rightarrow \texttt{aba}$.

g. True or False: $T \overset{*}{\Rightarrow} \texttt{aba}$.

h. True or False: $T \Rightarrow T$.

i. True or False: $T \overset{*}{\Rightarrow} T$.

j. True or False: $XXX \overset{*}{\Rightarrow} \texttt{aba}$.

k. True or False: $X \overset{*}{\Rightarrow} \texttt{aba}$.

l. True or False: $T \overset{*}{\Rightarrow} XX$.

m. True or False: $T \overset{*}{\Rightarrow} XXX$.

n. True or False: $S \overset{*}{\Rightarrow} \varepsilon$.

o. Give a description in English of $L(G)$.

2.4 Give context-free grammars that generate the following languages. In all parts, the alphabet Σ is $\{\texttt{0,1}\}$.

[A]a. $\{w \mid w$ contains at least three 1s$\}$

b. $\{w \mid w$ starts and ends with the same symbol$\}$

c. $\{w \mid$ the length of w is odd$\}$

[A]d. $\{w \mid$ the length of w is odd and its middle symbol is a 0$\}$

e. $\{w \mid w = w^{\mathcal{R}}$, that is, w is a palindrome$\}$

f. The empty set

2.5 Give informal descriptions and state diagrams of pushdown automata for the languages in Exercise 2.4.

2.6 Give context-free grammars generating the following languages.

[A]a. The set of strings over the alphabet $\{\texttt{a,b}\}$ with more a's than b's

b. The complement of the language $\{\texttt{a}^n\texttt{b}^n \mid n \geq 0\}$

[A]c. $\{w\#x \mid w^{\mathcal{R}}$ is a substring of x for $w, x \in \{\texttt{0,1}\}^*\}$

d. $\{x_1\#x_2\#\cdots\#x_k \mid k \geq 1$, each $x_i \in \{\texttt{a,b}\}^*$, and for some i and j, $x_i = x_j^{\mathcal{R}}\}$

[A]2.7 Give informal English descriptions of PDAs for the languages in Exercise 2.6.

[A]2.8 Show that the string `the girl touches the boy with the flower` has two different leftmost derivations in grammar G_2 on page 103. Describe in English the two different meanings of this sentence.

2.9 Give a context-free grammar that generates the language

$$A = \{\texttt{a}^i\texttt{b}^j\texttt{c}^k \mid i = j \text{ or } j = k \text{ where } i,j,k \geq 0\}.$$

Is your grammar ambiguous? Why or why not?

2.10 Give an informal description of a pushdown automaton that recognizes the language A in Exercise 2.9.

2.11 Convert the CFG G_4 given in Exercise 2.1 to an equivalent PDA, using the procedure given in Theorem 2.20.

2.12 Convert the CFG G given in Exercise 2.3 to an equivalent PDA, using the procedure given in Theorem 2.20.

2.13 Let $G = (V, \Sigma, R, S)$ be the following grammar. $V = \{S, T, U\}$; $\Sigma = \{0, \#\}$; and R is the set of rules:

$$\begin{aligned} S &\to TT \mid U \\ T &\to 0T \mid T0 \mid \# \\ U &\to 0U00 \mid \# \end{aligned}$$

a. Describe $L(G)$ in English.
b. Prove that $L(G)$ is not regular.

2.14 Convert the following CFG into an equivalent CFG in Chomsky normal form, using the procedure given in Theorem 2.9.

$$\begin{aligned} A &\to BAB \mid B \mid \varepsilon \\ B &\to 00 \mid \varepsilon \end{aligned}$$

2.15 Give a counterexample to show that the following construction fails to prove that the class of context-free languages is closed under star. Let A be a CFL that is generated by the CFG $G = (V, \Sigma, R, S)$. Add the new rule $S \to SS$ and call the resulting grammar G'. This grammar is supposed to generate A^*.

2.16 Show that the class of context-free languages is closed under the regular operations, union, concatenation, and star.

2.17 Use the results of Exercise 2.16 to give another proof that every regular language is context free, by showing how to convert a regular expression directly to an equivalent context-free grammar.

PROBLEMS

2.18 Consider the following CFG G:

$$\begin{aligned} S &\to SS \mid T \\ T &\to \mathtt{a}T\mathtt{b} \mid \mathtt{ab} \end{aligned}$$

Describe $L(G)$ and show that G is ambiguous. Give an unambiguous grammar H where $L(H) = L(G)$ and sketch a proof that H is unambiguous.

·2.19 We defined the rotational closure of language A to be $RC(A) = \{yx \mid xy \in A\}$. Show that the class of CFLs is closed under rotational closure.

·2.20 We defined the CUT of language A to be $CUT(A) = \{yxz \mid xyz \in A\}$. Show that the class of CFLs is not closed under CUT.

2.21 Show that every DCFG is an unambiguous CFG.

A·2.22 Show that every DCFG generates a prefix-free language.

·2.23 Show that the class of DCFLs is not closed under the following operations:

a. Union
b. Intersection
c. Concatenation
d. Star
e. Reversal

2.24 Let G be the following grammar:

$$\begin{aligned} S &\to T\dashv \\ T &\to T\mathtt{a}T\mathtt{b} \mid T\mathtt{b}T\mathtt{a} \mid \varepsilon \end{aligned}$$

a. Show that $L(G) = \{w\dashv \mid w$ contains equal numbers of a's and b's$\}$. Use a proof by induction on the length of w.
b. Use the DK-test to show that G is a DCFG.
c. Describe a DPDA that recognizes $L(G)$.

2.25 Let G_1 be the following grammar that we introduced in Example 2.45. Use the DK-test to show that G_1 is not a DCFG.

$$\begin{aligned} R &\to S \mid T \\ S &\to \mathtt{a}S\mathtt{b} \mid \mathtt{ab} \\ T &\to \mathtt{a}T\mathtt{bb} \mid \mathtt{abb} \end{aligned}$$

·2.26 Let $A = L(G_1)$ where G_1 is defined in Problem 2.25. Show that A is not a DCFL. (Hint: Assume that A is a DCFL and consider its DPDA P. Modify P so that its input alphabet is $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. When it first enters an accept state, it pretends that c's are b's in the input from that point on. What language would the modified P accept?)

·2.27 Let $B = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i, j, k \geq 0$ and $i = j$ or $i = k\}$. Prove that B is not a DCFL.

·2.28 Let $C = \{ww^{\mathcal{R}} \mid w \in \{0,1\}^*\}$. Prove that C is not a DCFL. (Hint: Suppose that when some DPDA P is started in state q with symbol x on the top of its stack, P never pops its stack below x, no matter what input string P reads from that point on. In that case, the contents of P's stack at that point cannot affect its subsequent behavior, so P's subsequent behavior can depend only on q and x.)

·2.29 If we disallow ε-rules in CFGs, we can simplify the DK-test. In the simplified test, we only need to check that each of DK's accept states has a single rule. Prove that a CFG without ε-rules passes the simplified DK-test iff it is a DCFG.

A2.30 a. Let C be a context-free language and R be a regular language. Prove that the language $C \cap R$ is context free.
b. Let $A = \{w \mid w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*$ and w contains equal numbers of a's, b's, and c's$\}$. Use part (a) to show that A is not a CFL.

·2.31 Let CFG G be the following grammar.

$$\begin{aligned} S &\to \mathtt{a}S\mathtt{b} \mid \mathtt{b}Y \mid Y\mathtt{a} \\ Y &\to \mathtt{b}Y \mid \mathtt{a}Y \mid \varepsilon \end{aligned}$$

Give a simple description of $L(G)$ in English. Use that description to give a CFG for $\overline{L(G)}$, the complement of $L(G)$.

2.32 Let $A/B = \{w \mid wx \in A \text{ for some } x \in B\}$. Show that if A is context free and B is regular, then A/B is context free.

*2.33 Let $\Sigma = \{\texttt{a},\texttt{b}\}$. Give a CFG generating the language of strings with twice as many a's as b's. Prove that your grammar is correct.

*2.34 Let $C = \{x\#y \mid x, y \in \{0,1\}^* \text{ and } x \neq y\}$. Show that C is a context-free language.

*2.35 Let $D = \{xy \mid x, y \in \{0,1\}^* \text{ and } |x| = |y| \text{ but } x \neq y\}$. Show that D is a context-free language.

*2.36 Let $E = \{\texttt{a}^i\texttt{b}^j \mid i \neq j \text{ and } 2i \neq j\}$. Show that E is a context-free language.

2.37 For any language A, let $SUFFIX(A) = \{v \mid uv \in A \text{ for some string } u\}$. Show that the class of context-free languages is closed under the $SUFFIX$ operation.

2.38 Show that if G is a CFG in Chomsky normal form, then for any string $w \in L(G)$ of length $n \geq 1$, exactly $2n - 1$ steps are required for any derivation of w.

*2.39 Let $G = (V, \Sigma, R, \langle\text{STMT}\rangle)$ be the following grammar.

$$\begin{aligned}
\langle\text{STMT}\rangle &\to \langle\text{ASSIGN}\rangle \mid \langle\text{IF-THEN}\rangle \mid \langle\text{IF-THEN-ELSE}\rangle \\
\langle\text{IF-THEN}\rangle &\to \texttt{if condition then } \langle\text{STMT}\rangle \\
\langle\text{IF-THEN-ELSE}\rangle &\to \texttt{if condition then } \langle\text{STMT}\rangle \texttt{ else } \langle\text{STMT}\rangle \\
\langle\text{ASSIGN}\rangle &\to \texttt{a:=1}
\end{aligned}$$

$$\Sigma = \{\texttt{if}, \texttt{condition}, \texttt{then}, \texttt{else}, \texttt{a:=1}\}$$
$$V = \{\langle\text{STMT}\rangle, \langle\text{IF-THEN}\rangle, \langle\text{IF-THEN-ELSE}\rangle, \langle\text{ASSIGN}\rangle\}$$

G is a natural-looking grammar for a fragment of a programming language, but G is ambiguous.

- **a.** Show that G is ambiguous.
- **b.** Give a new unambiguous grammar for the same language.

*2.40 Give unambiguous CFGs for the following languages.

- **a.** $\{w \mid \text{in every prefix of } w \text{ the number of a's is at least the number of b's}\}$
- **b.** $\{w \mid \text{the number of a's and the number of b's in } w \text{ are equal}\}$
- **c.** $\{w \mid \text{the number of a's is at least the number of b's in } w\}$

*2.41 Show that the language A in Exercise 2.9 is inherently ambiguous.

2.42 Use the pumping lemma to show that the following languages are not context free.

- **a.** $\{0^n1^n0^n1^n \mid n \geq 0\}$
- **^{A}b.** $\{0^n\#0^{2n}\#0^{3n} \mid n \geq 0\}$
- **^{A}c.** $\{w\#t \mid w \text{ is a substring of } t, \text{ where } w, t \in \{\texttt{a},\texttt{b}\}^*\}$
- **d.** $\{t_1\#t_2\#\cdots\#t_k \mid k \geq 2, \text{ each } t_i \in \{\texttt{a},\texttt{b}\}^*, \text{ and } t_i = t_j \text{ for some } i \neq j\}$

2.43 Let B be the language of all palindromes over $\{0,1\}$ containing equal numbers of 0s and 1s. Show that B is not context free.

2.44 Let $\Sigma = \{1,2,3,4\}$ and $C = \{w \in \Sigma^* \mid \text{in } w, \text{ the number of 1s equals the number of 2s, and the number of 3s equals the number of 4s}\}$. Show that C is not context free.

*2.45 Show that $F = \{\texttt{a}^i\texttt{b}^j \mid i = kj \text{ for some positive integer } k\}$ is not context free.

2.46 Consider the language $B = L(G)$, where G is the grammar given in Exercise 2.13. The pumping lemma for context-free languages, Theorem 2.34, states the existence of a pumping length p for B. What is the minimum value of p that works in the pumping lemma? Justify your answer.

2.47 Let G be a CFG in Chomsky normal form that contains b variables. Show that if G generates some string with a derivation having at least 2^b steps, $L(G)$ is infinite.

2.48 Give an example of a language that is not context free but that acts like a CFL in the pumping lemma. Prove that your example works. (See the analogous example for regular languages in Problem 1.49.)

*2.49 Prove the following stronger form of the pumping lemma, wherein *both* pieces v and y must be nonempty when the string s is broken up.

If A is a context-free language, then there is a number k where, if s is any string in A of length at least k, then s may be divided into five pieces, $s = uvxyz$, satisfying the conditions:

- **a.** for each $i \geq 0$, $uv^ixy^iz \in A$,
- **b.** $v \neq \varepsilon$ and $y \neq \varepsilon$, and
- **c.** $|vxy| \leq k$.

^{A}2.50 Refer to Problem 1.31 for the definition of the perfect shuffle operation. Show that the class of context-free languages is not closed under perfect shuffle.

2.51 Refer to Problem 1.32 for the definition of the shuffle operation. Show that the class of context-free languages is not closed under shuffle.

*2.52 Say that a language is ***prefix-closed*** if all prefixes of every string in the language are also in the language. Let C be an infinite, prefix-closed, context-free language. Show that C contains an infinite regular subset.

*2.53 Read the definitions of $NOPREFIX(A)$ and $NOEXTEND(A)$ in Problem 1.45.

- **a.** Show that the class of CFLs is not closed under $NOPREFIX$.
- **b.** Show that the class of CFLs is not closed under $NOEXTEND$.

*2.54 Let $Y = \{w \mid w = t_1\#t_2\#\cdots\#t_k \text{ for } k \geq 0, \text{ each } t_i \in 1^*, \text{ and } t_i \neq t_j \text{ whenever } i \neq j\}$. Here $\Sigma = \{1, \#\}$. Prove that Y is not context free.

2.55 For strings w and t, write $w \doteq t$ if the symbols of w are a permutation of the symbols of t. In other words, $w \doteq t$ if t and w have the same symbols in the same quantities, but possibly in a different order.

For any string w, define $SCRAMBLE(w) = \{t \mid t \doteq w\}$. For any language A, let $SCRAMBLE(A) = \{t \mid t \in SCRAMBLE(w) \text{ for some } w \in A\}$.

- **a.** Show that if $\Sigma = \{0,1\}$, then the $SCRAMBLE$ of a regular language is context free.
- **b.** What happens in part (a) if Σ contains three or more symbols? Prove your answer.

2.56 If A and B are languages, define $A \diamond B = \{xy \mid x \in A \text{ and } y \in B \text{ and } |x| = |y|\}$. Show that if A and B are regular languages, then $A \diamond B$ is a CFL.

*2.57 Let $A = \{wtw^{\mathcal{R}} \mid w, t \in \{0,1\}^* \text{ and } |w| = |t|\}$. Prove that A is not a CFL.

2.58 Let $\Sigma = \{0,1\}$ and let B be the collection of strings that contain at least one 1 in their second half. In other words, $B = \{uv \mid u \in \Sigma^*, v \in \Sigma^*1\Sigma^* \text{ and } |u| \geq |v|\}$.

a. Give a PDA that recognizes B.

b. Give a CFG that generates B.

2.59 Let $\Sigma = \{0,1\}$. Let C_1 be the language of all strings that contain a 1 in their middle third. Let C_2 be the language of all strings that contain two 1s in their middle third. So $C_1 = \{xyz \mid x, z \in \Sigma^* \text{ and } y \in \Sigma^*1\Sigma^*, \text{ where } |x| = |z| \geq |y|\}$ and $C_2 = \{xyz \mid x, z \in \Sigma^* \text{ and } y \in \Sigma^*1\Sigma^*1\Sigma^*, \text{ where } |x| = |z| \geq |y|\}$.

a. Show that C_1 is a CFL.

b. Show that C_2 is not a CFL.

SELECTED SOLUTIONS

2.3 (a) R, X, S, T; (b) a, b; (c) R; (d) Three strings in $L(G)$ are ab, ba, and aab; (e) Three strings not in $L(G)$ are a, b, and ε; (f) False; (g) True; (h) False; (i) True; (j) True; (k) False; (l) True; (m) True; (n) False; (o) $L(G)$ consists of all strings over a and b that are not palindromes.

2.4 (a) $S \to R1R1R1R$
$R \to 0R \mid 1R \mid \varepsilon$

(d) $S \to 0 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1$

2.6 (a) $S \to TaT$
$T \to TT \mid aTb \mid bTa \mid a \mid \varepsilon$
T generates all strings with at least as many a's as b's, and S forces an extra a.

(c) $S \to TX$
$T \to 0T0 \mid 1T1 \mid \#X$
$X \to 0X \mid 1X \mid \varepsilon$

2.7 (a) The PDA uses its stack to count the number of a's minus the number of b's. It enters an accepting state whenever this count is positive. In more detail, it operates as follows. The PDA scans across the input. If it sees a b and its top stack symbol is an a, it pops the stack. Similarly, if it scans an a and its top stack symbol is a b, it pops the stack. In all other cases, it pushes the input symbol onto the stack. After the PDA finishes the input, if a is on top of the stack, it accepts. Otherwise it rejects.

(c) The PDA scans across the input string and pushes every symbol it reads until it reads a #. If a # is never encountered, it rejects. Then, the PDA skips over part of the input, nondeterministically deciding when to stop skipping. At that point, it compares the next input symbols with the symbols it pops off the stack. At any disagreement, or if the input finishes while the stack is nonempty, this branch of the computation rejects. If the stack becomes empty, the machine reads the rest of the input and accepts.

2.8 Here is one derivation:
⟨SENTENCE⟩ → ⟨NOUN-PHRASE⟩⟨VERB-PHRASE⟩ →
⟨CMPLX-NOUN⟩⟨VERB-PHRASE⟩ →
⟨ARTICLE⟩⟨NOUN⟩⟨VERB-PHRASE⟩ →
The ⟨NOUN⟩⟨VERB-PHRASE⟩ →
`The girl` ⟨VERB-PHRASE⟩ →
`The girl` ⟨CMPLX-VERB⟩⟨PREP-PHRASE⟩ →
`The girl` ⟨VERB⟩⟨NOUN-PHRASE⟩⟨PREP-PHRASE⟩ →
`The girl touches` ⟨NOUN-PHRASE⟩⟨PREP-PHRASE⟩ →
`The girl touches` ⟨CMPLX-NOUN⟩⟨PREP-PHRASE⟩ →
`The girl touches` ⟨ARTICLE⟩⟨NOUN⟩⟨PREP-PHRASE⟩ →
`The girl touches the` ⟨NOUN⟩⟨PREP-PHRASE⟩ →
`The girl touches the boy` ⟨PREP-PHRASE⟩ →
`The girl touches the boy` ⟨PREP⟩⟨CMPLX-NOUN⟩ →
`The girl touches the boy with` ⟨CMPLX-NOUN⟩ →
`The girl touches the boy with` ⟨ARTICLE⟩⟨NOUN⟩ →
`The girl touches the boy with the` ⟨NOUN⟩ →
`The girl touches the boy with the flower`

Here is another leftmost derivation:
⟨SENTENCE⟩ → ⟨NOUN-PHRASE⟩⟨VERB-PHRASE⟩ →
⟨CMPLX-NOUN⟩⟨VERB-PHRASE⟩ →
⟨ARTICLE⟩⟨NOUN⟩⟨VERB-PHRASE⟩ →
The ⟨NOUN⟩⟨VERB-PHRASE⟩ →
`The girl` ⟨VERB-PHRASE⟩ →
`The girl` ⟨CMPLX-VERB⟩ →
`The girl` ⟨VERB⟩⟨NOUN-PHRASE⟩ →
`The girl touches` ⟨NOUN-PHRASE⟩ →
`The girl touches` ⟨CMPLX-NOUN⟩⟨PREP-PHRASE⟩ →
`The girl touches` ⟨ARTICLE⟩⟨NOUN⟩⟨PREP-PHRASE⟩ →
`The girl touches the` ⟨NOUN⟩⟨PREP-PHRASE⟩ →
`The girl touches the boy` ⟨PREP-PHRASE⟩ →
`The girl touches the boy` ⟨PREP⟩⟨CMPLX-NOUN⟩ →
`The girl touches the boy with` ⟨CMPLX-NOUN⟩ →
`The girl touches the boy with` ⟨ARTICLE⟩⟨NOUN⟩ →
`The girl touches the boy with the` ⟨NOUN⟩ →
`The girl touches the boy with the flower`

Each of these derivations corresponds to a different English meaning. In the first derivation, the sentence means that the girl used the flower to touch the boy. In the second derivation, the boy is holding the flower when the girl touches her.

2.22 We use a proof by contradiction. Assume that w and wz are two unequal strings in $L(G)$, where G is a DCFG. Both are valid strings so both have handles, and these handles must agree because we can write $w = xhy$ and $wz = xhyz = xh\hat{y}$ where h is the handle of w. Hence, the first reduce steps of w and wz produce valid strings u and uz, respectively. We can continue this process until we obtain S_1 and S_1z where S_1 is the start variable. However, S_1 does not appear on the right-hand side of any rule so we cannot reduce S_1z. That gives a contradiction.