PROBLEMS

[A]**2.18** **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. Use part (a) to show that the language $A = \{w|\ w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*$ and contains equal numbers of a's, b's, and c's$\}$ is not a CFL.

*__2.19__ Let CFG G be

$$S \rightarrow \mathtt{a}S\mathtt{b} \mid \mathtt{b}Y \mid Y\mathtt{a}$$
$$Y \rightarrow \mathtt{b}Y \mid \mathtt{a}Y \mid \varepsilon$$

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

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

*__2.21__ Let $\Sigma = \{\mathtt{a},\mathtt{b}\}$. Give a CFG generating the language of strings with twice as many a's as b's. Prove that your grammar is correct.

*__2.22__ Let $C = \{x\#y|\ x, y \in \{\mathtt{0},\mathtt{1}\}^*$ and $x \neq y\}$. Show that C is a context-free language.

*__2.23__ Let $D = \{xy|x, y \in \{\mathtt{0},\mathtt{1}\}^*$ and $|x| = |y|$ but $x \neq y\}$. Show that D is a context-free language.

*__2.24__ Let $E = \{\mathtt{a}^i\mathtt{b}^j|\ i \neq j$ and $2i \neq j\}$. Show that E is a context-free language.

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

2.26 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.

*__2.27__ Let $G = (V, \Sigma, R, \langle\text{STMT}\rangle)$ be the following grammar.

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

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

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.

*__2.28__ Give unambiguous CFGs for the following languages.

a. $\{w|$ in every prefix of w the number of a's is at least the number of b's$\}$

b. $\{w|$ the number of a's and b's in w are equal$\}$

c. $\{w|$ the number of a's is at least the number of b's$\}$

*__2.29__ Show that the language A in Exercise 2.9 is inherently ambiguous.

2.30 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.** $\{w\#t \mid w$ is a substring of t, where $w, t \in \{a, b\}^*\}$

d. $\{t_1\#t_2\# \cdots \#t_k \mid k \geq 2$, each $t_i \in \{a, b\}^*$, and $t_i = t_j$ for some $i \neq j\}$

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

***2.32** Let $\Sigma = \{1, 2, 3, 4\}$ and $C = \{w \in \Sigma^* \mid$ in w, 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.

2.33 Show that $F = \{a^ib^j \mid i \neq kj$ for every positive integer $k\}$ is not context free.

2.34 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.

2.35 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.

2.36 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 1.54.)

***2.37** 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}**2.38** Refer to Problem 1.41 for the definition of the perfect shuffle operation. Show that the class of context-free languages is not closed under perfect shuffle.

2.39 Refer to Problem 1.42 for the definition of the shuffle operation. Show that the class of context-free languages is not closed under shuffle.

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

***2.41** Read the definitions of $NOPREFIX(A)$ and $NOEXTEND(A)$ in Problem 1.40.

a. Show that the class of CFLs is not closed under $NOPREFIX$ operation.

b. Show that the class of CFLs is not closed under $NOEXTEND$ operation.

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

2.43 For strings w and t, write $w \overset{\circ}{=} t$ if the symbols of w are a permutation of the symbols of t. In other words, $w \overset{\circ}{=} t$ if t and w have the same symbols in the same quantities, but possibly in a different order.

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

a. Show that, if $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, then the *SCRAMBLE* of a regular language is context free.

b. What happens in part (a) if Σ contains 3 or more symbols? Prove your answer.

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

***2.45** Let $A = \{wtw^{\mathcal{R}} |\ w, t \in \{0,1\}^* \text{ and } |w| = |t|\}$. Prove that A is not a context-free language.

SELECTED SOLUTIONS

2.3 **(a)** R, X, S, T; **(b)** a, b; **(c)** R; **(d)** Three strings in G are ab, ba, and aab; **(e)** Three strings not in 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)**
$S \to R\mathtt{1}R\mathtt{1}R\mathtt{1}R$
$R \to \mathtt{0}R \mid \mathtt{1}R \mid \varepsilon$

(d) $S \to \mathtt{0} \mid \mathtt{0}S\mathtt{0} \mid \mathtt{0}S\mathtt{1} \mid \mathtt{1}S\mathtt{0} \mid \mathtt{1}S\mathtt{1}$

2.6 **(a)**
$S \to T\mathtt{a}T$
$T \to TT \mid \mathtt{a}T\mathtt{b} \mid \mathtt{b}T\mathtt{a} \mid \mathtt{a} \mid \varepsilon$

T generates all strings with at least as many a's as b's, and S forces an extra a.

(c)
$S \to TX$
$T \to \mathtt{0}T\mathtt{0} \mid \mathtt{1}T\mathtt{1} \mid \#X$
$X \to \mathtt{0}X \mid \mathtt{1}X \mid \varepsilon$

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 0. In more detail, it operates as follows. The PDA scans across the input. If it sees a b and its top stack symbol is a a, it pops the stack. Similarly, if it scans a 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 scans the input, if b 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 # 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.