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.

\[
E \rightarrow \ E + T | T \\
T \rightarrow T \times F | F \\
F \rightarrow (E) | a
\]

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 $A = \{a^m b^n c^n \mid m, n \geq 0\}$ and $B = \{a^n b^n c^m \mid m, n \geq 0\}$ 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.

2.3 Answer each part for the following context-free grammar $G$.

\[
R \rightarrow XRX | S \\
S \rightarrow aTa | bTa \\
T \rightarrow XTX | X | \varepsilon \\
X \rightarrow a | b
\]

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: $T \Rightarrow \text{aba}$.

g. True or False: $T \Rightarrow \text{aba}$.

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

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

j. True or False: $XX X \Rightarrow \text{aba}$.

k. True or False: $X \Rightarrow \text{aba}$.

l. True or False: $T \Rightarrow XX$.

m. True or False: $T \Rightarrow XXX$.

n. True or False: $S \Rightarrow \varepsilon$.

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

a. $\{w \mid w \text{ contains at least three } 1s\}$

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

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

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

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

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. The set of strings over the alphabet \{a,b\} with more a's than b's

   b. The complement of the language \{a^n b^n | n \geq 0\}

   c. \{w # x | w \cdot R \text{ is a substring of } x \text{ for } w, x \in \{0,1\}^*\}

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

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

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 101. Describe in English the two different meanings of this sentence.

2.9 Give a context-free grammar that generates the language \(A = \{a^i b^i c^k | i = j \text{ or } j = k \text{ where } i, j, k \geq 0\}\).

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:

\[
\begin{align*}
S & \rightarrow TT | U \\
T & \rightarrow 0T | T0 | \# \\
U & \rightarrow 0U00 | \#
\end{align*}
\]

   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.

\[
\begin{align*}
A & \rightarrow BAB | B | \varepsilon \\
B & \rightarrow 00 | \varepsilon
\end{align*}
\]

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 \(S \rightarrow SS\) 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 Problem 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 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 \( A = \{ w \mid w \in \{a, b, c\}^* \text{ and contains equal numbers of a's, b's, and c's} \} \) is not a CFL.

2.19 Let CFG \( G \) be

\[
\begin{align*}
S & \rightarrow aSb \mid bY \mid Ya \\
Y & \rightarrow bY \mid aY \mid \epsilon
\end{align*}
\]

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

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

2.23 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.

2.24 Let \( E = \{ a^ib^i \mid i \neq j \text{ and } 2i \neq j \} \). Show that \( E \) is a context-free language.

2.25 For any language \( A \), let \( SUFFIX(A) = \{ wv \in A \text{ for some string } w \} \). 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 \( G = (V, \Sigma, R, \langle STMT \rangle) \) be the following grammar.

\[
\begin{align*}
\langle STMT \rangle & \rightarrow \langle ASSIGN \rangle \mid \langle IF-THEN \rangle \mid \langle IF-THEN-ELSE \rangle \\
\langle IF-THEN \rangle & \rightarrow \text{if } \text{condition} \text{ then } \langle STMT \rangle \\
\langle IF-THEN-ELSE \rangle & \rightarrow \text{if } \text{condition} \text{ then } \langle STMT \rangle \text{ else } \langle STMT \rangle \\
\langle ASSIGN \rangle & \rightarrow \text{a:=1}
\end{align*}
\]

\( \Sigma = \{\text{if, condition, then, else, a:=1}\} \).

\( V = \{\langle STMT \rangle, \langle IF-THEN \rangle, \langle IF-THEN-ELSE \rangle, \langle ASSIGN \rangle\} \)

\( G \) 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. \( \{ w \mid \text{in every prefix of } w \text{ the number of a's is at least the number of b's} \} \)

b. \( \{ w \mid \text{the number of a's and b's in } w \text{ are equal} \} \)

c. \( \{ w \mid \text{the number of a's is at least the number of b's} \} \)

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. \( \{0^n1^n0^n1^n \mid n \geq 0 \} \)

b. \( \{0^n#0^n#0^n \mid n \geq 0 \} \)

c. \( \{w\#t \mid w \text{ is a substring of } t, \text{ where } w, t \in \{a, b\}^* \} \)

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

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

2.33 Show that \( F = \{a^ib^j \mid i \neq kj \text{ for every positive integer } k \} \) 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. \( |vxy| \leq k \).

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 \( \text{NOPREFIX}(A) \) and \( \text{NOEXTEND}(A) \) in Problem 1.40.

a. Show that the class of CFLs is not closed under \( \text{NOPREFIX} \) operation.

b. Show that the class of CFLs is not closed under \( \text{NOEXTEND} \) operation.

2.42 Let \( \Sigma = \{1, \#\} \) and \( Y = \{w \mid w = t_1\#t_2\#\cdots\#t_k \text{ for } k \geq 0, \text{ each } t_i \in 1^*, \text{ and } t_i \neq t_j \text{ whenever } i \neq j \} \). Prove that \( Y \) is not context free.
2.43 For strings \( w \) and \( t \), write \( w \preceq t \) if the symbols of \( w \) are a permutation of the symbols of \( t \). In other words, \( w \preceq t \) if \( t \) and \( w \) have the same symbols in the same quantities, but possibly in a different order.

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

a. Show that, if \( \Sigma = \{0, 1\} \), then the \( \text{SCRAMBLE} \) of a regular language is context free.

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

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

*2.45 Let \( A = \{ w^R t w \mid w, t \in \{0, 1\}^* \text{ and } |w| = |t| \} \). 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 \( \varepsilon \); (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) \( S \to R1R1R1R \) \( R \to 0R \mid 1R \mid \varepsilon \)

(b) \( S \to 0 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \)

2.6 (a) \( S \to TaT \) \( T \to TT \mid aTb \mid bTa \mid a \mid \varepsilon \)

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

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

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.

(b) 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.
2.8 Here is one derivation:
\[(\text{SENTENCE}) \Rightarrow (\text{NOUN-PHRASE})(\text{VERB-PHRASE}) \Rightarrow \]
\[(\text{CMPLX-noun})(\text{CMPLX-VERB})(\text{PREP-PHRASE}) \Rightarrow \]
\[(\text{ARTICLE})(\text{NOUN})(\text{CMPLX-VERB})(\text{PREP-PHRASE}) \Rightarrow \]
The boy (\text{VERB})(\text{NOUN-PHRASE})(\text{PREP})(\text{CMPLX-noun}) \Rightarrow \]
The boy touches (\text{NOUN-PHRASE})(\text{PREP})(\text{CMPLX-noun}) \Rightarrow \]
The boy touches (\text{CMPLX-noun})(\text{PREP})(\text{CMPLX-noun}) \Rightarrow \]
The boy touches (\text{ARTICLE})(\text{NOUN})(\text{PREP})(\text{CMPLX-noun}) \Rightarrow \]
The boy touches the girl with (\text{CMPLX-noun}) \Rightarrow \]
The boy touches the girl with (\text{ARTICLE})(\text{NOUN}) \Rightarrow \]
The boy touches the girl with the flower

Here is another derivation:
\[(\text{SENTENCE}) \Rightarrow (\text{NOUN-PHRASE})(\text{VERB-PHRASE}) \Rightarrow \]
\[(\text{CMPLX-noun})(\text{VERB-PHRASE}) \Rightarrow (\text{ARTICLE})(\text{NOUN})(\text{VERB-PHRASE}) \Rightarrow \]
The boy (\text{VERB-PHRASE}) \Rightarrow \text{The boy (CMPLX-VERB)} \Rightarrow \]
The boy (\text{VERB})(\text{NOUN-PHRASE}) \Rightarrow \]
The boy touches (\text{NOUN-PHRASE}) \Rightarrow \]
The boy touches (\text{CMPLX-noun})(\text{PREP-PHRASE}) \Rightarrow \]
The boy touches (\text{ARTICLE})(\text{NOUN})(\text{PREP-PHRASE}) \Rightarrow \]
The boy touches the girl (\text{PREP-PHRASE}) \Rightarrow \]
The boy touches the girl (\text{PREP})(\text{CMPLX-noun}) \Rightarrow \]
The boy touches the girl with (\text{CMPLX-noun}) \Rightarrow \]
The boy touches the girl with (\text{ARTICLE})(\text{NOUN}) \Rightarrow \]
The boy touches the girl with the flower

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

2.18 (a) Let \( C \) be a context-free language and \( R \) be a regular language. Let \( P \) be the PDA that recognizes \( C \), and \( D \) be the DFA that recognizes \( R \). If \( Q \) is the set of states of \( P \) and \( Q' \) is the set of states of \( D \), we construct a PDA \( P' \) that recognizes \( C \cap R \) with the set of states \( Q \times Q' \). \( P' \) will do what \( P \) does and also keep track of the states of \( D \). It accepts a string \( w \) if and only if it stops at a state \( q \in F_P \times F_D \), where \( F_P \) is the set of accept states of \( P \) and \( F_D \) is the set of accept states of \( D \). Since \( C \cap R \) is recognized by \( P' \), it is context free.

(b) Let \( R \) be the regular language \( a^*b^*c^* \). If \( A \) were a CFL then \( A \cap R \) would be a CFL by part (a). However, \( A \cap R = \{a^n b^n c^n | n \geq 0 \} \), and Example 2.36 proves that \( A \cap R \) is not context free. Thus \( A \) is not a CFL.

2.30 (b) Let \( B = \{0^n\#0^{2n}\#0^{3n} | n \geq 0 \} \). Let \( p \) be the pumping length given by the pumping lemma. Let \( s = 0^p\#0^{2p}\#0^{3p} \). We show that \( s = uvxyz \) cannot be pumped.

Neither \( v \) nor \( y \) can contain \#, otherwise \( x^2 y^2 z \) contains more than two \#'s. Therefore, if we divide \( s \) into three segments by \#'s: \( 0^n, 0^{2p}, \) and \( 0^{3p} \), at least one of the segments is not contained within either \( v \) or \( y \). Hence \( x^2 y^2 z \) is not in \( B \) because the 1 : 2 : 3 length ratio of the segments is not maintained.
(c) Let \( C = \{ w#t \mid w \text{ is a substring of } t, \text{ where } w, t \in \{ a, b \}^* \} \). Let \( p \) be the pumping length given by the pumping lemma. Let \( s = a^p b^p \# a^p b^p \). We show that the string \( s = uvxyz \) cannot be pumped. Neither \( v \) nor \( y \) can contain \( \# \), otherwise \( uv^0xy^0z \) does not contain \( \# \) and therefore is not in \( C \). If both \( v \) and \( y \) are nonempty and occur on the left-hand side of the \( \# \), the string \( uv^0xy^0z \) cannot be in \( C \) because it is longer on the left-hand side of the \( \# \). Similarly, if both strings occur on the right-hand side of the \( \# \), the string \( uv^0xy^0z \) cannot be in \( C \) because it is again longer on the left-hand side of the \( \# \). If only one of \( v \) and \( y \) is nonempty (both cannot be nonempty), treat them as if both occurred on the same side of the \( \# \) as above.

The only remaining case is where both \( v \) and \( y \) are nonempty and straddle the \( \# \). But then \( v \) consists of \( b \)'s and \( y \) consists of \( a \)'s because of the third pumping lemma condition \( |vxy| \leq p \). Hence, \( uv^2xy^2z \) contains more \( b \)'s on the left-hand side of the \( \# \), so it cannot be a member of \( C \).

2.38 Let \( A \) be the language \( \{ 0^k1^k \mid k \geq 0 \} \) and let \( B \) be the language \( \{ a^k b^{2k} \mid k \geq 0 \} \). The perfect shuffle of \( A \) and \( B \) is the language \( C = \{ (0a)^k (0b)^k (1b)^{2k} \mid k \geq 0 \} \). Languages \( A \) and \( B \) are easily seen to be CFGs, but \( C \) is not a CFL, as follows. If \( C \) were a CFL, let \( p \) be the pumping length given by the pumping lemma, and let \( s \) be the string \( (0a)^p (0b)^p (1b)^{2p} \). Because \( s \) is longer than \( p \) and \( s \in C \), we can divide \( s = uvxyz \) satisfying the pumping lemma's three conditions. Strings in \( C \) contain twice as many \( 1 \)'s as \( a \)'s. In order for \( uv^2xy^2z \) to have that property, the string \( vxy \) must contain both \( 1 \)'s and \( a \)'s. But that is impossible, because they are separated by \( 2p \) symbols yet the third condition says that \( |vxy| \leq p \). Hence \( C \) is not context free.