CS 4510

Spring 2015

Test 2 Practice questions

Notes:

- Please write neat and legible answers.
- You can use any of the theorems/facts/lemmas that we covered in *class* without re-proving them unless explicitly stated otherwise. You can also cite homework problems and test practice problems from this course.
- Please state clearly any assumptions you make.
- Topics:
  - Context-free grammars: simplifications, normal forms such as CNF, ambiguity, special forms such as linear grammars.
  - Push-down automata.
  - Pumping lemma for CFLs.
  - Closure properties of CFLs.
  - Algorithms involving CFGs.

1. Circle all the violations that make this grammar one that is not in Chomsky-Normal Form.

$$\begin{aligned} S &\rightarrow bB \\ A &\rightarrow AS \mid \epsilon \end{aligned}$$

2. Describe an algorithm to decide if a given CFG G generates the empty string.

   Solution: We saw in class a CFG simplification algorithm to remove all ϵ-rules from G and produce a CFG G' such that:

   - if $\epsilon \notin L(G)$ then G' has no ϵ-rules, and
   - if $\epsilon \in L(G)$ then G' has only the following ϵ-rule: $S' \rightarrow \epsilon$.

   Use this algorithm to produce the grammar G' and test to see if there is a rule of the form $S' \rightarrow \epsilon$.

3. A *linear* grammar is a context-free grammar in which each rule is in one of the four forms below:

$$\begin{aligned} A &\rightarrow a \\ A &\rightarrow aB \\ A &\rightarrow Ba \\ A &\rightarrow \epsilon \end{aligned}$$

where A, B are variables and a is a terminal symbol. That is, the right hand side of each rule has at most one variable in it.

Answer TRUE/FALSE and justify your answer: A linear grammar is unambiguous.

Solution: FALSE. The grammar G below is a linear grammar but it is ambiguous.

$$\begin{aligned}
S &\rightarrow A \mid B \\
A &\rightarrow 0C \mid 1C \\
C &\rightarrow 0A \mid 1A \mid \epsilon \\
B &\rightarrow D0 \mid D1 \\
D &\rightarrow B0 \mid B1 \mid \epsilon
\end{aligned}$$

4. Answer TRUE/FALSE with a brief justification: Every regular language is generated by an unambiguous context-free grammar.

   Solution: TRUE. Let L be a regular language. Let M be a DFA for L. On input $w \in L$, there is a unique path in the DFA from the start state to a final accepting state labeled by the symbols in w. Let G be the CFG constructed from M as shown in class/text. The derivation for w corresponds to the accepting computation path in the DFA and thus is unique.

5. Answer TRUE/FALSE with a brief justification: Every context-free language without ϵ can be generated by a context-free grammar in which every rule is of the form:

$$\begin{aligned}
A &\rightarrow BCD \\
A &\rightarrow a
\end{aligned}$$

   where a is an alphabet symbol and A,B,C, and D are variables.

   Solution: FALSE. Such a grammar can generate only odd length strings.

6. What is the language generated by the CFG below. Informally justify your answer.

$$\begin{aligned}
S &\rightarrow ASB \mid AB \\
A &\rightarrow 0 \\
B &\rightarrow 11
\end{aligned}$$

   Solution: $L = \{0^k 1^{2k} \mid k \geq 1\}$.

   The second S-rule generates $011 \in L$. Assuming inductively that the variable S in the first S-rule generates a string in L, the first S-rule generates the "next" string in L. That is, all strings generated by G are in L.

Let $w = 0^k1^{2k}$ be a string in L. If $k = 1$ then w is generated as follows:

$$S \Rightarrow AB \Rightarrow 0B \Rightarrow 011.$$

Let $k > 1$. Then, w is generated as follows:

$$S \Rightarrow ASB \Rightarrow 0SB \Rightarrow \cdots \Rightarrow 00^{k-1}1^{2k-2}B \Rightarrow 0^k1^{2k}.$$

7. Construct a context-free grammar for the language: $\{a^nb^{2m}c^md^{2n} \mid n, m > 0\}$. Briefly justify your construction.

   Solution:

$$\begin{array}{lll} S & \rightarrow & aAdd \mid B \\ A & \rightarrow & aAdd \mid B \\ B & \rightarrow & bbBc \mid bbc \end{array}$$

8. A variable A is said to be *terminating* if there is a rule of the form $A \rightarrow \alpha$ such that either α is a string of terminals or all the variables in α are terminating variables.

   (a) What are the terminating variables in the grammar below? (The alphabet is $\{a\}$.)

$$\begin{array}{lll} S & \rightarrow & BD \mid CS \mid CC \\ A & \rightarrow & AC \mid aF \\ B & \rightarrow & aB \mid aA \\ C & \rightarrow & DA \mid a \\ D & \rightarrow & aD \mid E \\ E & \rightarrow & aE \mid CE \mid D \\ F & \rightarrow & aB \mid CSB \end{array}$$

   (b) Delete from the grammar above all variables that are not terminating and all rules that involve these variables. What is the language generated by this grammar?

9. Show that the language $\{a^ib^jc^id^j \mid i, j \geq 1\}$ is not context-free using the pumping lemma.

   Solution: Let $s = a^pb^pc^pd^p$ where p is the pumping lemma constant.

10. Construct a push-down automaton for the language: $\{0^i1^j \mid i \leq j \leq 3i\}$.

    Solution: For every 0 in the input, nondeterministically push one or two or three 1's onto the stack. For every 1 in the input, match it with a 1 in the stack.

11. Construct a PDA for the language $\{x^R\#y \mid x, y \in \{0,1\}^* \text{ and } x \text{ is a substring of y}\}$.

    A string $x = x_1x_2\cdots x_k$ is a substring of a string $y = y_1y_2\cdots y_n$, where $x_i, y_i \in \Sigma$, if there exist $1 \leq j \leq n - k + 1$ such that $y_{j+p-1} = x_p$ for $1 \leq p \leq k$.

    Solution: Read x and push it onto the stack. Guess the position j in y where the string x occurs as a substring. From this position onwards, read a symbol in y and match it with the stack symbol.

12. Show that the complement of the the context-free language $\{a^ib^ic^j \mid i, j \geq 0\}$ is also context-free.

    Solution: The complement is the union of the two context-free languages below:

    - The set of all strings in which that characters a, b, c are not in order.
    - The set of all strings in $a^*b^*c^*$ in which the number of $a's$ and $b's$ are different.

    The first language is regular and hence context free. The second language is context-free. A PDA for this language stacks all the a's, pops an a for each b in the input, rejects if the stack becomes empty when the b''s are exhausted, ignores the c's.

13. A *restricted pushdown automaton* is defined as a pushdown automaton with the restriction that the stack alphabet has exactly one symbol, say A other than the stack bottom marker \$. That is, the stack in a pushdown automaton is always of the form $A^n\$$ for some $n \geq 0$.

    Construct a *restricted pushdown automaton* for the following languages:

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

    (b) $\{w \in \{0,1\}^* \mid \#_0(w) = \#_1(w)\}$.