University of Nevada, Las Vegas Computer Science 456/656 Spring 2026

Additional Problems to Work in Preparation for the Second Test.

If w is any string, we write w^{-1} for the *inverse* of w. For example, if $w = abcca$ then $w^{-1} = accba$. A *palindrome* is a string which is equal to its own inverse, such as "level." We write $\#_0(w)$ for the number of zeros in w, and $\#_1(w)$ for the number of ones.

1. True or False. If the answwer is unknown to science at this time, write **O** for Open.

   (a) **F** Let L be the language over $\{a, b, c\}$ consisting of all strings which have more a's than b's and more b's than c's. There is some PDA that accepts L.

   (b) **T** The union of any two context-free languages must be context-free

   (c) **T** The language $\{a^n b^n \mid n \geq 0\}$ is context-free.

   (d) **F** The language $\{a^n b^n c^n \mid n \geq 0\}$ is context-free.

   (e) **T** The language $\left\{a^i b^j c^k \mid j = i + k\right\}$ is context-free.

   (f) **T** The intersection of any regular language with any context-free language is context-free.

   (g) **F** The intersection of any two context-free languages is context-free.

   (h) **T** If L is a context-free language over an alphabet with just one symbol, then L is regular.

   (i) **F** There is an LALR parser for any context-free grammar.

   (j) **T** The set of strings that your high school algebra teacher would accept as legitimate expressions is a context-free language.

   (k) **T** The Kleene closure of any context-free language is context-free.

   (l) **T** Every regular language is context-free.

   (m) **T** Every context-free language is in $\mathcal{P}$.

   (n) **T** Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

   (o) **T** The problem of whether a given string is generated by a given context-free grammar is decidable.

   (p) **T** If G is a context-free grammar, the question of whether $L(G) = \emptyset$ is decidable.

   (q) **F** If L_1 is $\mathcal{NP}$ complete and L_2 is $\mathcal{NP}$, there is a $\mathcal{P}$–TIME reduction from L_1 to L_2.

   (r) **T** If L_1 is $\mathcal{NP}$-hard and there is a $\mathcal{P}$–TIME reduction of L_1 to L_2, then L_2 must be $\mathcal{NP}$-hard.

   (s) **T** The partition problem is $\mathcal{NP}$ complete.

2. Give two context-free languages whose intersection is not context-free.

   One easy example is $L_1 = \{a^n b^n c^m\}$ and $L_2 = \{a^m b^n c^n\}$.

3. Identify which grammar generates each language. In each case, the language is over the binary alphabet $\Sigma = \{0, 1\}$.

   (a) The language of all palindromes of even length over Σ.

   (b) The language of all palindromes of odd length over Σ.

   (ii)

   (c) The language of all strings w over Σ such that $\#_0(w) = \#_1(w)$.

   (iv)

   (d) The language of all strings w over Σ such that $\#_0(w) = \#_1(w)$ and each prefix of w has at least as many zeros as ones.

   (i)

   (e) The set of all binary numerals for multiples of three, where leading zeros are allowed.

   (v)

(i)

$S \to 0S1S$

$S \to \lambda$

(ii)

$S \to 0S0$

$S \to 1S1$

$S \to \lambda$

(iii)

$S \to 0S0$

$S \to 1S1$

$S \to 0$

$S \to 1$

(iv)

$S \to 0S1S$

$S \to 1S0S$

$S \to \lambda$

(v)

$S \to 0S$

$S \to 1A$

$A \to 1S$

$A \to 0B$

$B \to 1B$

$B \to 0A$

$S \to 0$

4. The following grammar G_1 generates a simple algebraic language. Prove that G_1 is ambiguous by writing two different rightmost derivations for some string in $L(G_1)$. The start symbol of G_1 is E.

   1. $E \to E + E$
   2. $E \to E * E$
   3. $E \to (E)$
   4. $E \to x$
   5. $E \to y$
   6. $E \to z$

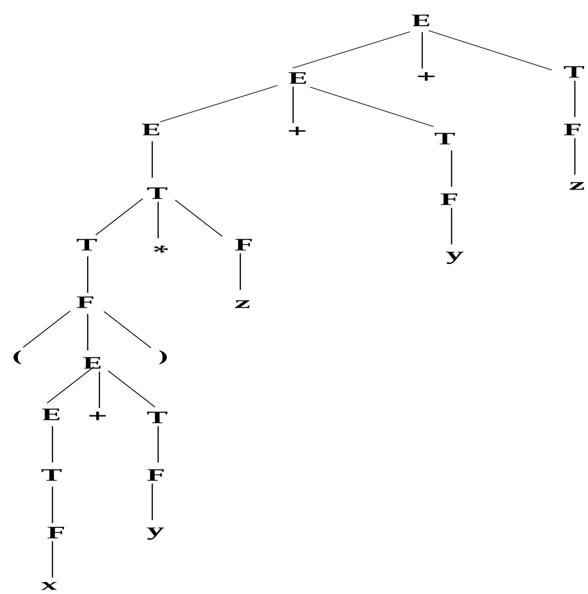
   Let $w = x + y * z$. Two different rightmost derivations of w are:

   $E \Rightarrow E * E \Rightarrow E * z \Rightarrow E + E * z \Rightarrow E + y * z \Rightarrow x + y * z$

   $E \Rightarrow E + E \Rightarrow E + E * E \Rightarrow E + E * z \Rightarrow E + y * z \Rightarrow x + y * z$

5. The grammar G_2, given below, is equivalent to G_1, but is unambiguous. Draw a parse tree for $(x + y) * z$ using G_2. (E stands for "expression," T stands for "term," and F stands for "factor.")

   1. $E \to E + T$
   2. $E \to T$
   3. $T \to T * F$
   4. $T \to F$
   5. $F \to (E)$
   6. $F \to x$
   7. $F \to y$
   8. $F \to z$



6. Consider the following list of languages. In each case, the alphabet is $\Sigma = \{a, b\}$.

(a) The language L_1 where each prefix of any string in L_2 has at least as many $a's$ as $b's$.

I don't believe that any of the machines below accept this language. Sorry.

(b) The language $L_2 \subseteq L_1$ consisting of all strings which have an equal number of each symbol.

The second machine.

(c) The language of all even length palindromes.

The first machine.

Each of the above languages is accepted by one of the PDA shown below. Which one?