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

Answers to Assignment 2: Due Saturday September 12, 2026

Officially, the Dyck language is over the alphabet $\{(,)\}$, the left and right parentheses, and consists of all "balanced" strings of parentheses, such as "()" "(())()" However, when written without sufficient caution, such as by rushed students taking an exam, the two symbols look so similar that the grader (me, or whoever) is easily confused. Thus, this semester, I will use the alphabet $\{a, b\}$ instead, so that the Dyck language contains, for example, the strings "*ab*" and "*aabbab*."

1. True **T**/False **F**/Open **O**, meaning that the answer is not known to science at this time.

   (i) **O** $\mathcal{P} = \mathcal{NP}$.

   (ii) **O** $\mathcal{P} = \mathcal{NC}$.

   (iii) **T** co-$\mathcal{P} = \mathcal{P}$.

   (iv) **O** co-$\mathcal{NP} = \mathcal{NP}$.

   (v) **T** A language is recursive (decidable) if and only if it is both $\mathcal{RE}$ and co-$\mathcal{RE}$.

   (vi) **T** A language is context-free if and only if it is accepted by some PDA.

   (vii) **T** The game *Rush Hour*, where you try to move the red car out of the "parking lot," is known to be $\mathcal{P}$–SPACE complete. (Hint: look it up on the intenet.)

2. Let G_1 be the following grammar.
   1. $S \to aSb$
   2. $S \to SS$
   3. $S \to \lambda$

   (i) What language does G_1 generate? (It has a spaeific name.)

   The Dyck language.

   (ii) Show that G_1 is ambiguous by writing two different leftmost derivations of some string in that language. (If you draw two different parse trees, but do not give two different leftmost derivations, you have not answered the question.)

   There are many examples, but the simplest string which has two leftmost derivations is the empty string:

   $S \Rightarrow \lambda$
   $S \Rightarrow SS \Rightarrow S \Rightarrow \lambda$

   (iii) Give an unambiguous context-free grammar G_2 which is equivalent to G_1.

   There is more than one answer. Here is one of them:

   1. $S \to aSbS$
   2. $S \to \lambda$

Here is another:

1. $S \to SaSb$
2. $S \to \lambda$

There are others which are more complicated.

3. Let L be the language over $\{a, b\}$ consisting of all strings which have equal numbers of each symbol, that is, $L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$.

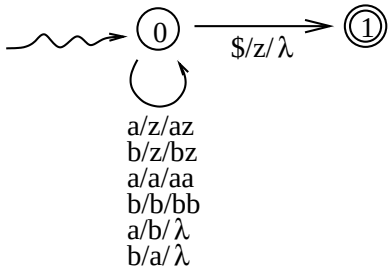
   (i) **F** (T/F/O) G is regular.

   (ii) Give a context-free grammar for L.

   There is more than one answer. Here is an ambiguous grammar.

   1. $S \to aSbS$
   2. $S \to bSaS$
   3. $S \to \lambda$

   (iii) Draw a DPDA which accepts L. Attach an extra sheet if necessary.



4. State the pumping lemma for regular languages. Write it carefully. The statement contains both implications and nested quantifiers, which must be in the correct order.

   For any regular language L
   There exists an integer p such that
   For any string $w \in L$ such that $|w| \geq p$
   There exist strings x, y, z such that the following four statements hold:
   1. $w = xyz$
   2. $|xy| \leq p$
   3. $|y| > 0$
   4. For any $i \geq 0$ $\quad xy^iz \in L$

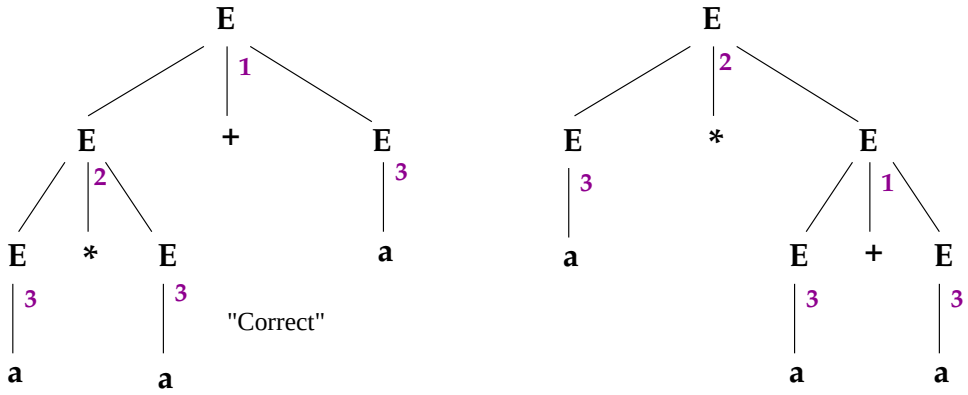
5. I have altered this question. I'll explain in class.

   Here is context-free grammar G which gnerates a language L.. The start symbol of G is E.

   1. $E -> E + E$
   2. $E -> E * E$
   4. $E -> a$

   (i) G is ambiguous. Draw two different parse trees for some $w \in L(G)$.

   I have chosen the string $a * a + a$.



   (ii) Both parse trees are valid as far as the grammar is concerned, but only one of them would be considered "correct" by your high school algebra teacher. Which one?

   (iii) **T** (T/F/O) There exists an unambiguous context-free grammar which generates L and resolves the ambiguity.

6. Give the certificate/verifier definition of the class $\mathcal{NC}$.

   A language $L \subseteq \Sigma^*$ is a member of class $\mathcal{NC}$ if and only if:

   1. There exists a number k and a machine V_L (the verifier) such that
   For every $w \in L$
   There exists a string c, the certificate, or witness of w, such that V accepts the concatenation wc in $O(|w|^k)$ time.

   2. For any string $w \notin L$ and any string c, V does not accept the concatenation wc.

7. List the Chomsky grammar classes, giving their standard type numbers.

   Type 0: General (unrestricted, phase-structure) grammars.

   Tyoe 1: Context-sensitive grammars.

   Type 2: Context-free grammars.

   Type 3: Regular grammars.