University of Nevada, Las Vegas Computer Science 456/656 Spring 2024

Assignment 3: Due Friday February 16, 2024, 11:59 PM

Name:_____

You are permitted to work in groups, get help from others, read books, and use the internet. You will receive a message from the graduage assistant, Zachary Edwards, telling you how to turn in the assignment.

Recall that there is no such thing as the "set" of regular languges, only the "class" of regular languages. But if we pick an alphabet, we can have sets. For example, we can talk about the set of binary languages, the set of lenguages over the binary alphabet $\{0, 1\}$.

1. True or False, write T or F. If the answer is unknown to science at this time, write O, for Open.

   (a) **O** $\mathcal{P}$–TIME $= \mathcal{NP}$.

   (b) **O** $\mathcal{P}$–TIME $= \mathcal{P}$–SPACE.

   (c) **T** Every $\mathcal{NP}$ language is decidable.

   (d) **T** The set of recursive real numbers is countable.

   (e) **F** Every recursively enumerable language is decidable.

   (f) **F** The problem of whether two context-free grammars are equivalent is recursively enumerable.

   (g) **T** The set of all C++ programs is countable.

   (h) **T** The set of recursively enumerable binary languages is countable.

   (i) **F** The set of binary languages is countable.

   (j) **F** If a CF grammar G is ambiguous, there cannot be any deterministic machine which constructs a parse tree for every string in $L(G)$.

   (k) **F** The busy beaver function is recursive. (Look it up.)

   (l) **T** The language $\{a^n b^n c^n d^n \ : \ n \geq 0\}$ is context-sensitive.

   (m) **O** There is a polynomial time algorithm for deciding whether a given Boolean expression is a contradiction. (An example of a contradiction is $x*!x$, where x is a variable of Boolean type and "$*$" means "and.")

   (n) **F** The halting problem is decidable.

   (o) **T** The halting problem is recursively enumerable.

2. State the pumping lemma for regular languages correctly.

   If L is any regular language, there is an integer p such that, for any string $w \in L$ of length at least p, there are string x, y, z such that the following four statements are true:
   1. $w = xyz$,
   2. $|xy| \leq p$,
   3. $|y| > 0$,
   4. For any integer $i \geq 0$, $xy^i z \in L$.

3. For each of the grammars given below, each variable symbol is a capital Roman letter, each terminal symbol is a lower case Roman letter, and the start symbol is S.

(a) Which of these grammars generates $\{a^n b^n : n \geq 0\}$?

G_1

(b) Which of these grammars generates $\{a^n b^m : n, m \geq 0, n \neq m\}$?

G_2

(c) Which two of these grammars are equivalent?

G_3 and G_4

(d) Which of these grammars are ambiguous?

G_2 and G_4

Grammar G_1:
$S \to aSb$
$S \to \lambda$

Grammar G_2:
$S \to aA \,|\, Bb$
$A \to aA \,|\, aAb \,|\, \lambda$
$B \to Bb \,|\, aBb \,|\, \lambda$

Grammar G_3:
$S \to aSbS$
$S \to \lambda$

Grammar G_4:
$S \to aSb$
$S \to SS$
$S \to \lambda$

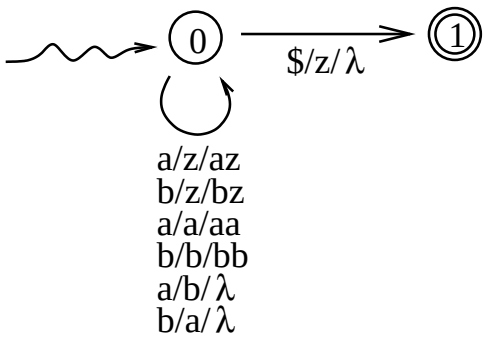
4. The correct order of the three labels on a transition of a PDA are written in this order: "read/pop/push." Commas can be used instead of slashes.

When we push two or more symbols, the topmost one is on the left of the string. For example: if the label on the transition is $a/z/az$, then z is popped, a is read, and az is pushed, but that means z is pushed first, then a is pushed on top of it. A PDA transition must pop exactly one symbol, reads either zero or one symbol, and can push any number of symbols.

Assume that initially the stack is not empty, but contains just the bottom-of-stack symbol, z.

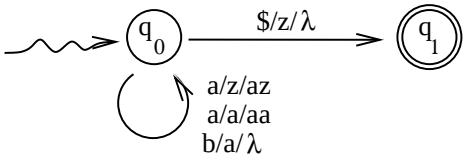
The input of a DPDA must end with the end-of-file symbol, which we write \$. The input of an NPDA does not need the end of file symbol, since it can correctly guess that it has reached the end.

(a) What is the language accepted by the DPDA illustrated below?



The language of all strings over $\{a, b\}$ which have an equal number of each symbol.

(b) Design a DPDA which accepts the Dyck language. For ease of grading, use the version where the terminal symbols are a and b, not parentheses.



(c) The language L generated by the gammar $S \to iS|iSeS|\lambda$ is ambiguous. Is there a DPDA that accepts L? If so, draw it. If not, draw an NPDA that accepts L.

Yes, that language is accepted by a DPDA. However, I have not covered enough material for you to be able to design it. So, we will "push" this problem into the future.

(d) Draw a PDA which accepts the language of all palindromes over $\{a, b\}$. Your machine cannot be a DPDA, as you know. (Note: a palindrome can have odd or even length.)