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

Answers to Assignment 4: Due Saturday October 24, 2026

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

   (i) **F** The set of all regular expressions over a given alphabet is a regular language.

   (ii) **T** Every regular language is $\mathcal{NC}$.

   (iii) **O** The CVP (Circuit Value Problem) is $\mathcal{NC}$.

   (iv) **O** Evaluation of a Boolean expression given an assignment is $\mathcal{NC}$.

   (v) **T** 2-SAT is $\mathcal{P}$–TIME

   (vi) **O** 4-SAT is $\mathcal{P}$–TIME

   (vii) **T** Every context-free language is generated by some Chomsky Normal Form grammar.

   (viii) **T** Every $\mathcal{RE}$ (recursively enumerable) language is generated by some general grammar. Every context-free language is generated by some Chomsky Normal Form grammar.

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

   (x) **T** If L is $\mathcal{P}$–TIME, there is a $\mathcal{NC}$ reduction of L to CVP.

   (xi) **T** Every regular language is $\mathcal{NC}$.

   (xii) **O** Given two regular expressions the problem of whether they are equivalent (that is, descirbe the same regular language) is $\mathcal{P}$-TIME.

2. Each of these languages or, equivalently, 0-1 problems. Mark each them Y (for yes) it it is known to be $\mathcal{NP}$-complete, and N (for no) it is not known to be $\mathcal{NP}$-complete. (You may have to consult the internet.)

   (i) **F** 2-SAT

   (ii) **T** 3-SAT

   (iii) **T** 4-SAT

   (iv) **T** TSP, the Traveling Salesman Problem.

   (v) **F** 2-coloring of a graph.

   (vi) **T** Haniltonian circuit problem. Does a given graph has a Hamiltonian circuit?

   (vii) **T** 3-coloring of a graph.

   (viii) **T** Block sorting. Given a string w can w be sorted within K block steps for a given K?

   (ix) **F** Generalized Checkers. Given a position on an $N \times N$ board, can Black force a win?

   (x) **F** RUSH HOUR. Given a position of the "cars" in a "parking lot" of arbitray size, is there a way to move the red car out?

3. Identify which machine accepts the language defined by each regular expression.

(a) $a^* + b^*$ M_7

(b) λ M_2

(c) a^* M_3

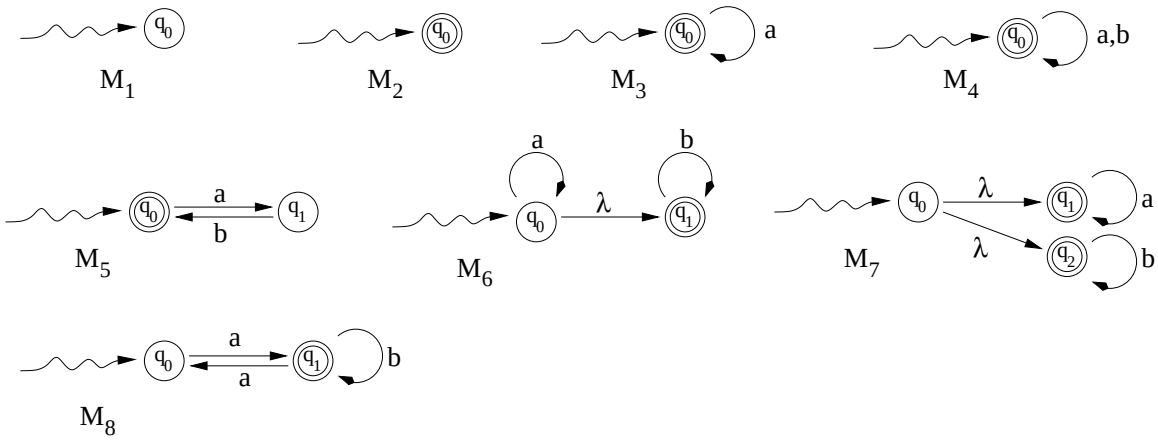
(d) $\emptyset$ M_1

(e) $a(aa + b)^*$ M_8

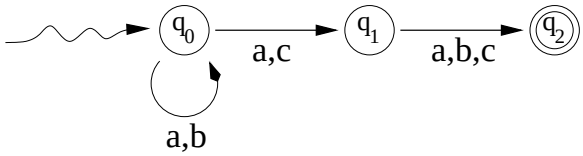
(f) a^*b^* M_6

(g) $(a + b)^*$ M_4

(h) $(ab)^*$ M_5



4. Write a regular expression for the language accepted by the NFA below.



The simplest answer I could find:: $(a + b)^*(a + c)(a + b + c)$

5. Write a Chomsky Normal Form grammar for the Dyck language.

Here is the usual grammar.

$S \to aSbS$

λ

Here is my first attempt.

$S \to XY$

$S \to \lambda$

$X \to AS$

$Y \to BS$

$A \to a$

$B \to b$

That grammar generates the Dyck language, but it is not in CNF because the start symbol appears on the right side of a production.

Here is a hopefully correct, answer. One tricky technique is to introduce a variable S', a proxy for the start symbol.

$S \to XY.$

$S \to \lambda$

$X \to AS'$

$Y \to BS'$

$A \to a$

$B \to b$

$X \to a$

$Y \to b$

We need those last two productions because we can't allow $S' \to \lambda$.

6. Consider the Chomsky Normal Form grammar G given below, where S is the start symbol.

$S \to IS$
$S \to WS$
$S \to XY$
$X \to IS$
$Y \to ES$
$S \to a$
$I \to i$
$W \to w$
$E \to e$

(i) Show that G is ambiguous by giving two different **leftmost** derivations for the string $iiaea$.

Using the usual CF grammar equivalent to G, we have the following two leftmost derivations of $iiaea$:

$S \Rightarrow iS \Rightarrow iiSeS \Rightarrow iiaeS \Rightarrow iiaea$

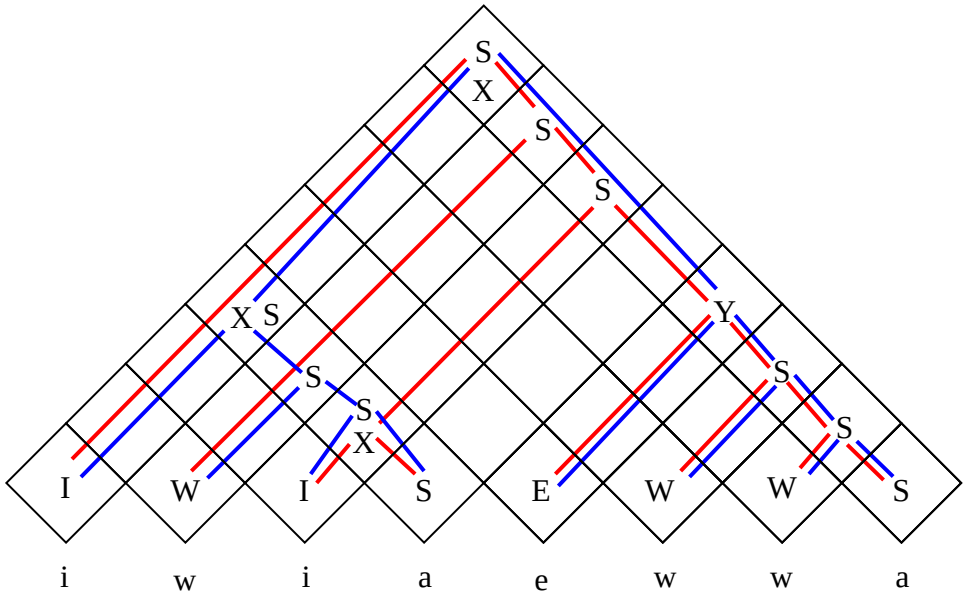
$S \Rightarrow iSeS \Rightarrow iiSeS \Rightarrow iiaeS \Rightarrow iiaea$

We then "translate" those two derivations into leftmost derivations of $iiaea$ using G:

$S \Rightarrow IS \Rightarrow iS \Rightarrow iXY \Rightarrow iiSY \Rightarrow iiaY \Rightarrow iiaES \Rightarrow iiaeS \Rightarrow iiaea$

$S \Rightarrow XY \Rightarrow iiSY \Rightarrow iiaY \Rightarrow iiaES \Rightarrow iiaeS \Rightarrow iiaea$

(ii) Use the CYK algorithm to prove that $iwiaewwa \in L(G)$.



The above figure shows the edges of each of the two parse trees for $iwiaewwa$. I did not require you to write the edges of a parse tree into the figure, but I'm showing them here, anyway. But you **were** required to write every variables generated by the CYK algorithm, even if it is not used to generate the root S. For example, the variable X in the topmost cell of the figure is generated by CYK but is not part of either parse tree.

8. Use the pumping lemma for regular languages to prove that $L = \{a^n b^n : n \geq 0\}$ is not regular.

Proof. Assume L is regular. Let p be a pumping length of L. Let $w = a^p b^p \in L$ Note that $|w| \geq p$. By the pumping lemma, there are strings x, y, z such that:

1. $w = xyz$
2. $|xy| \leq p$
3. $|y| > 0$.
4. For any $i \geq 0$, $xy^i z \in L$

Let $i = 0$. Then $xz \in L$. Let $k = |y|$, which is positive. Since $|xy| \leq p$, xy is a substring of a^p, which implies that $y = a^k$, hence $xz = a^{p-k} b^p \notin L$, contradiction. We conclude that L is not regular. □