**University of Nevada, Las Vegas Computer Science 456/656 Fall 2023**

**Assignment 7: Due Sunday November 19, 2023, 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, Sepideh Farivar, telling you how to turn in the assignment.

1. True or False. 5 points each. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

   (a) _____ The context-free grammar equivalence problem is decidable.

   (b) _____ The context-free grammar equivalence problem is co-$\mathcal{RE}$.

   (c) _____ If $L_1$ is a regular language and $L_2$ is a context-free language, then $L_1 \cap L_2$ is context-free.

   (d) _____ If there is a recursive reduction of $L_1$ to $L_2$, where $L_1$ is an undecidable language, then $L_2$ must be undecidable.

   (e) _____ The factoring problem is in $\mathcal{P}$.

   (f) _____ If $L$ is a recursive language, there must be a machine which enumerates $L$ in canonical order.

   (g) _____ If there is a machine which enumerates a language $L$ in canonical order, then $L$ must be recursive.

   (h) _____ The set of all real numbers is countable.

   (i) _____ The set of all recursive real numbers is countable.

   (j) _____ For any alphabet $\Sigma$, the set of all languages over $\Sigma$ is countable.

   (k) _____ For any alphabet $\Sigma$, the set of all recursive languages over $\Sigma$ is countable.

   (l) _____ For any alphabet $\Sigma$, the set of all recursively enumerable languages over $\Sigma$ is countable.

   (m) _____ Every subset of a recursive language is recursive.

2. Give a definition of a *recursive* real number. (There is more than one correct definition.)

3. Which of these languages (problems) are **known** to be $\mathcal{NP}$-complete? If a language, or problem, is known to be $\mathcal{NP}$-complete, fill in the first circle. If it is either known not be be $\mathcal{NP}$-complete, or if whether it is $\mathcal{NP}$-complete is not known at this time, fill in the second circle.

   ○ ○ Boolean satisfiability.
   ○ ○ 2SAT.
   ○ ○ 3SAT.
   ○ ○ 4SAT.
   ○ ○ Subset sum problem.
   ○ ○ Generalized checkers, *i.e.* on a board of arbitrary size.
   ○ ○ Independent set problem.
   ○ ○ Traveling salesman problem.
   ○ ○ Regular expression equivalence.
   ○ ○ C++ program equivalence.
   ○ ○ Rush Hour: https://www.youtube.com/watch?v=HI0rlp7tiZ0
   ○ ○ Circuit value problem, CVP.
   ○ ○ Regular grammar equivalence.
   ○ ○ Dominating set problem.
   ○ ○ Partition.

4. State the pumping lemma for regular languages.

5. Let $L$ be the language over $\{a, b\}$ consisting of all strings which have the same number of $a$'s as $b$'s, such as aabb, abba, aaabbb, bbbaaa, .... Design a DPDA which accepts $L$.

6. Give a polynomial time reduction of the subset sum problem to partition.

7. Give a polynomial time reduction of 3SAT to the independent set problem.

8. Prove that a language is recursively enumerable, $\mathcal{RE}$, if and only if it is accepted by some machine.

9. Consider $G$, the following context-free gammar with start symbol $E$. Stack states are indicated.

   1. $E \to E_{1,11} +_2 E_3$
   2. $E \to E_{1,11} -_4 E_5$
   3. $E \to E_{1,3,5,11} *_6 E_7$
   4. $E \to -_8 E_9$
   5. $E \to (_{10} E_{11})_{12}$
   6. $E \to x_{13}$

   What follows is an ACTION table followed by a GOTO table for an LALR parser for $G$.

   (a) Which entry guarantees that negation has higher priority than multiplication?

   (b) Oops! I somehow forgot to fill in the column headed by "−." Fill it in. Hint: unlike the others, this column contains no empty cells. Remember that the minus sign is used for both subtraction and negation.

| | $x$ | + | − | ∗ | ( | ) | \$ | $E$ |
|---|---|---|---|---|---|---|---|---|
| 0 | s13 | | | | s10 | | | 1 |
| 1 | | s2 | | s6 | | | **halt** | |
| 2 | s13 | | | | s10 | | | 3 |
| 3 | | r1 | | s6 | | r1 | r1 | |
| 4 | s13 | | | | s10 | | | 5 |
| 5 | | r2 | | s6 | | r2 | r2 | |
| 6 | s13 | | | | s10 | | | 7 |
| 7 | | r3 | | r3 | | r3 | r3 | |
| 8 | s13 | | | | s10 | | | 9 |
| 9 | | r4 | | r4 | | r4 | r4 | |
| 10 | s13 | | | | s10 | | | 11 |
| 11 | | s2 | | s6 | | s12 | | |
| 12 | | r5 | | r5 | | r5 | r5 | |
| 13 | | r6 | | r6 | | r6 | r6 | |

(c) Give a complete computation of the parser if the input string is $x + x * -(-x * x)$.

10. Consider the following problem. Given binary numerals $\langle u \rangle$ and $\langle v \rangle$ of length $n$, decide whether $u < v$. Give an $\mathcal{NC}$ algorithm for solving this problem.

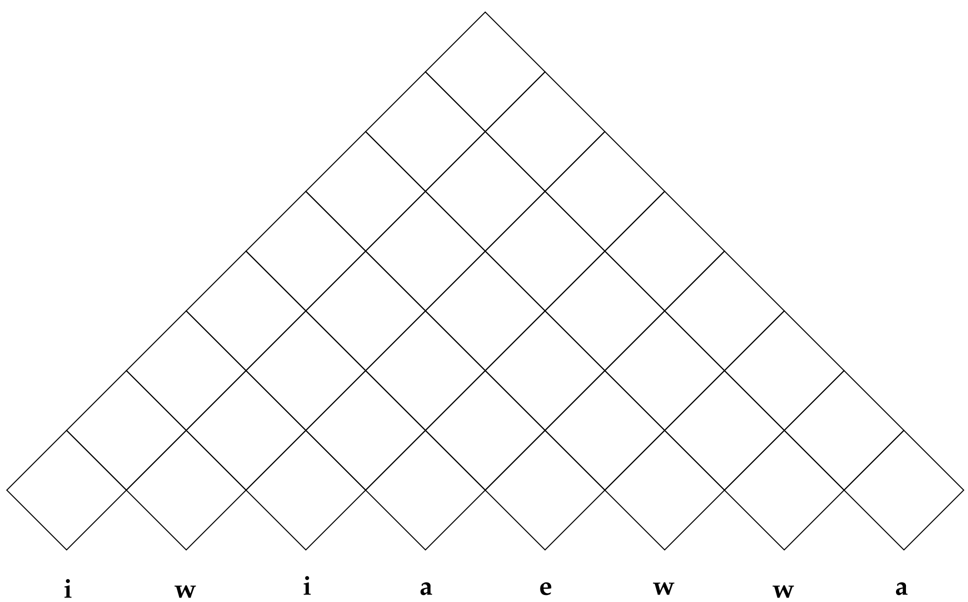
11. Prove that a language is enumerable in canonical order by some machine if and only if it is decidable.

12. Consider the Chomsky Normal Form grammar $G$ given below.

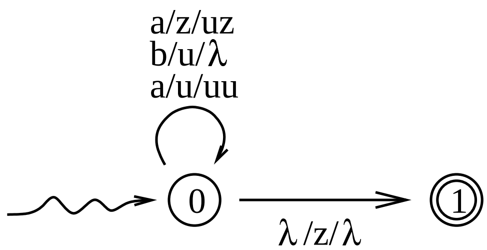
$S \to IS$
$S \to WS$
$S \to XY$
$X \to IS$
$Y \to ES$
$S \to a$
$I \to i$
$W \to w$
$E \to e$

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

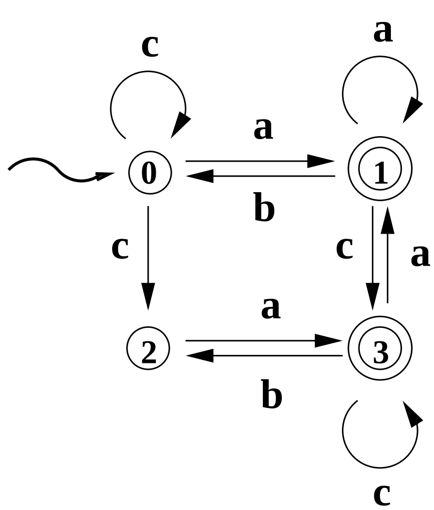
(b) Use the CYK algorithm to prove that $iwiaewwa \in L(G)$.



13. Let $L$ be the language accepted by the PDA diagrammed below. What is $L$? You can either describe $L$ in a **few** words, or give a context-free grammar for $L$.




14. Find a minimal DFA equivalent to the NFA shown below.



15. Fill in the following table, showing which operations are closed for each class of languages. In each box, write **T** if it is known that that language class is closed under that operation, **F** it is known that that class is not closed under that operation, and **O** if neither of those is known.

| language class | union | intersection | concatenation | Kleene closure | complementation |
|---|---|---|---|---|---|
| $\mathcal{NC}$ | | | | | |
| regular | | | | | |
| context-free | | | | | |
| $\mathcal{P}$ | | | | | |
| $\mathcal{NP}$ | | | | | |
| co-$\mathcal{NP}$ | | | | | |
| recursive | | | | | |
| $\mathcal{RE}$ | | | | | |
| co-$\mathcal{RE}$ | | | | | |
| undecidable | | | | | |

16. Prove that the halting problem is undecidable.

17. Consider the following well-known complexity classes.

$$\mathcal{NC} \subseteq \mathcal{P} - \text{TIME} \subseteq \mathcal{NP} - \text{TIME} \subseteq \mathcal{P} - \text{SPACE} \subseteq \mathbf{EXP} - \text{TIME} \subseteq \mathbf{EXP} - \text{SPACE}$$

The *mover's problem* is, given a room with a door and pieces of furniture of various shapes and sizes, can the furniture be moved into the room through the door?

The *crane operator's problem* is, given a room and pieces of furniture of various shapes and sizes, can the furniture be placed into the room after the roof is removed?

For both furniture problems, we assume that no piece of furniture can ever be fully or partially on top of another.

(a) Which of the above complexity classes is the smallest class which is known to contain the mover's problem?

(b) Which of the above complexity classes is the smallest class which is known to contain the crane operator's problem?

(c) Which of the above complexity classes is the smallest class which is known to contain the context-free grammar membership problem?

(d) Which of the above complexity classes is the smallest class which is known to contain the circuit valuation problem, which is the problem of determining the output of a Boolean circuit with given inputs?

(e) *Generalized checkers* is the game of checkers played on an $n \times n$ board. (The standard game uses an $8 \times 8$ board.) Which of the above complexity classes is the smallest class which is known to contain the problem of determining whether the first player to move, from a given configuration, can win?

18. $f$ is a *one-way function* if $f(x)$ can be computed in polynomial time for any string $x$, but there is no polynomial time randomized algorithm which can invert $f$; that is, given $f(x)$, find, with high probability, a string $x'$ such that $f(x) = f(x')$. Such a function would be useful in cryptography.

    The formal definition is given at `https://en.wikipedia.org/wiki/One-way_function` There are some functions that are generally believed to be one-way, but no one knows for sure. Prove that if $\mathcal{P} = \mathcal{NP}$, no one-way function exists.

19. The binary factorization problem is, given a binary numeral for an integer $n$, and another "benchmark" numeral for an integer $a$, determine whether $n$ has a factor greater than 1 and less than $a$. Prove that the binary integer factorization problem is in $\mathcal{NP}$. (It is also in co-$\mathcal{NP}$, but that is harder to prove.)