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

Practice Problems for the Examination on October 25, 2023

1. Review answers to homework3:
   `http://web.cs.unlv.edu/larmore/Courses/CSC456/S23/Assignments/hw3ans.pdf`
2. Review answers to homework4:
   `http://web.cs.unlv.edu/larmore/Courses/CSC456/S23/Assignments/hw4ans.pdf`
3. Review answers to homework5:
   `http://web.cs.unlv.edu/larmore/Courses/CSC456/S23/Assignments/hw5ans.pdf`
4. True or False. If the question is currently open, write "O" or "Open."

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

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

   (iii) ________ Every regular language is $\mathcal{NC}$.

   (iv) ________ Every context-free language is $\mathcal{NC}$.

   (v) ________ The Boolean circuit problem is $\mathcal{NC}$

   (vi) ________ The complement of any $\mathcal{P}$-TIME language is $\mathcal{P}$-TIME.

   (vii) ________ The complement of any $\mathcal{NP}$ language is $\mathcal{NP}$.

   (viii) ________ The complement of any $\mathcal{P}$-SPACE language is $\mathcal{P}$-SPACE.

   (ix) ________ The complement of every recursive language is recursive.

   (x) ________ The complement of every recursively enumerable language is recursively enumerable.

   (xi) ________ If p is the pumping length of a regular language L, then $p + 1$ is also the pumping length of L.

   (xii) ________ If a language L is accepted by an NFA with p states, then p is the pumping length of L.

   (xiii) ________ Every language which is generated by a general grammar is recursively enumerable.

   (xiv) ________ The context-free membership problem is undecidable.

   (xv) ________ A Given any context-free grammar G and any string $w \in L(G)$, there is always a unique leftmost derivation of w using G.

   (xvi) ________ A For any non-deterministic finite automaton, there is always a unique minimal deterministic finite automaton equivalent to it.

   (xvii) ________ The union of any two context-free languages is context-free.

   (xviii) ________ The question of whether a given Turing Machine halts with empty input is decidable.

   (xix) ________ The class of languages accepted by non-deterministic finite automata is the same as the class of languages accepted by deterministic finite automata.

(xx) ________ The class of languages accepted by non-deterministic push-down automata is the same as the class of languages accepted by deterministic push-down automata.

(xxi) ________ Let π be the ratio of the circumference of a circle to its diameter. The problem of whether the n^{th} digit of the decimal expansion of π for a given n is equal to a given digit is decidable.

(xxii) ________ There cannot exist any computer program that can decide whether any two C++ programs are equivalent.

(xxiii) ________ An undecidable language is necessarily $\mathcal{NP}$-complete.

(xxiv) ________ Every context-free language is in the class $\mathcal{P}$-TIME.

(xxv) ________ Every regular language is in the class $\mathcal{NC}$

(xxvi) ________ Every Function that can be mathematically defined is recursive.

(xxvii) ________ Every bounded function from integers to integers is Turing-computable. (We say that f is bounded if there is some B such that $|f(n)| \leq B$ for all n.)

(xxviii) ________ The language of all palindromes over $\{0, 1\}$ is inherently ambiguous.

(xxix) ________ The boolean satisfiability problem is undecidable.

(xxx) ________ If $\mathcal{P} = \mathcal{NP}$, then all one-way encoding systems are breakable in polynomial time.

(xxxi) ________ A language L is in $\mathcal{NP}$ if and only if there is a polynomial time reduction of L to SAT.

(xxxii) ________ Every subset of a regular language is regular.

(xxxiii) ________ The intersection of any context-free language with any regular language is context-free.

(xxxiv) ________ The question of whether two context-free grammars generate the same language is undecidable.

(xxxv) ________ There exists some proposition which is true but which has no proof.

(xxxvi) ________ A If L_1 reduces to L_2 in polynomial time, and if L_2 is $\mathcal{NP}$, and if L_1 is $\mathcal{NP}$-complete, then L_2 must be $\mathcal{NP}$-complete.

(xxxvii) ________ B Given any context-free grammar G and any string $w \in L(G)$, there is always a unique leftmost derivation of w using G.

(xxxviii) ________ The question of whether two regular expressions are equivalent is $\mathcal{NP}$-complete. (Do not guess. Look it up.)

(xxxix) ________ No language which has an ambiguous context-free grammar can be accepted by a DPDA.

(xl) ________ The intersection of any two regular languages is regular.

(xli) ________ The intersection of any two context-free languages is context-free.

(xlii) ________ A If L_1 reduces to L_2 in polynomial time, and if L_2 is $\mathcal{NP}$, then L_1 must be $\mathcal{NP}$.

(xliii) ________ Let $F(0) = 1$, and let $F(n) = 2^{F(n-1)}$ for $n > 0$. Then F is recursive.

(xliv) ________ Every language which is accepted by some non-deterministic machine is accepted by some deterministic machine.

(xlv) ________ A The language of all regular expressions over the binary alphabet is a regular language.

(xlvi) ________ There cannot exist any computer program that decides whether any two given C++ programs are equivalent.

(xlvii) ________ An undecidable language is necessarily $\mathcal{NP}$-complete.

(xlviii) Every context-free language is in the class $\mathcal{P}$-TIME.

(xlix) ________ Every function that can be mathematically defined is recursive.

(l) ________ Every bounded function from integers to integers is recursive. (We say that f is bounded if there is some B such that $|f(n)| \leq B$ for all n.)

(li) ________ Every function that can be mathematically defined is recursive.

(lii) ________ The language of all binary strings which are the binary numerals for multiples of 23 is regular.

(liii) ________ Let β be the busy beaver function. You know that β is not recursive, but there is some recursive function F such that $\beta = O(F)$.

5. Which of the following languages or problems are **known** to be $\mathcal{NP}$-complete? Write "T" if it is known to be $\mathcal{NP}$-complete, "F" otherwise. ("O" is not an option for this problem.) You may have to seach the internet.

   (i) ________ SAT

   (ii) ________ 2-SAT

   (iii) ________ 3-SAT

   (iv) ________ 4-SAT

   (v) ________ 5-SAT

   (vi) ________ Boolean Circuit.

   (vii) ________ Context-free membership.

   (viii) ________ The language of all strings generated by a given unrestricted grammar.

   (ix) ________ The set of all solvable configurations of RUSH HOUR.

   (x) ________ Given a big rectangle and a set of smaller rectangles, is it possible to place all the small rectangles into the big rectangle with no overlap?

   (xi) ________ The block sorting problem. Given a list of n items and a number K, a "block move" moves a contiguous subset of items into another location in the list. Can the list be sorted with no more than K block moves? For example, ABCLMNODEFGHIJK can be sorted with 1 block move.

   (xii) ________ Given a configuration in a game of generalized checkers (that means, any size board) can the black player force a win?

   (xiii) ________ The firehouse problem. Given a graph $G = (V, E)$ and numbers K and d, is there a set $F \subseteq V$ of size K such that every vertex is within at most d steps of some member of F?

   (xiv) ________ The traveling salesman problem.

   (xv) ________ Given a sequence σ of distinct integers, does σ have an increasing subsequence?

6. State the pumping lemma for regular languages.

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

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

9. This is not a question, but you must understand it!

   A deteriministic machine has at most one computation for a given input, but a non-deterministic machine could have many possible computations. We say that a non-deterministic machine M accepts a string w if, given w as input, M has at least one computation that ends in an accepting state. If L is a language, we say M accepts L if M accepts every $w \in L$ and accepts no other strings.

   If L is a language, we say that a non-deterministic machine M accepts L in polynomial time if M accepts L, and there is some constant k such that, for each $w \in L$, there is an accepting computation of M with input w consisting of $O(n^k)$ steps, where $n = |w|$.

   $\mathcal{NP}$–TIME (or simply $\mathcal{NP}$) is defined to be the class of all languages which are accepted by some machine in polynomial time.