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

Practice Problems for the Examination on April 10, 2024

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

   (i) ______ Every subset of a regular language is regular.

   (ii) ______ Every language is enumerable.

   (iii) ______ If L is any $\mathcal{P}$–TIME language, there is a reduction of the Boolean circuit problem (CVP) to L which can be computed in polylogarithmic time using polynomially many processors.

   (iv) ______ The intersection of any two RE languages is RE.

   (v) ______ $\mathcal{P}$–TIME $= \mathcal{NC}$.

   (vi) ______ The context-free grammar equivalence problem is decidable.

   (vii) ______ The set of all positions of generalized checkers ($N \times N$ board for any N) from which black can win is decidable.

   (viii) ______ Every function that can be mathematically defined is bounded by some recursive function.

   (ix) ______ There are uncountably many languages over the binary alphabet.

   (x) ______ There are uncountably many RE languages over the binary alphabet.

   (xi) ______ If a language is both $\mathcal{NP}$ and co–$\mathcal{NP}$, it must be $\mathcal{P}$–TIME.

   (xii) ______ There is a $\mathcal{P}$–TIME algorithm which determines whether a given set of n positive integers has a subset whose total is n.

   (xiii) ______ If L_1 is $\mathcal{NP}$-complete and L_2 is $\mathcal{NP}$ and there is a $\mathcal{P}$–TIME reduction of L_1 to L_2, then L_2 must be $\mathcal{NP}$–COMPLETE.

   (xiv) ______ Nick's Class is closed under intersection.

   (xv) ______ Every subset of any enumerable set is enumerable.

   (xvi) ______ Every subset of any recursively enumerable language is recursively enumerable.

   (xvii) ______ The computer language C++ has Turing power.

   (xviii) ______ Binary numeral multiplication is $\mathcal{NC}$.

   (xix) ______ There is an $\mathcal{P}$–SPACE algorithm which decides SAT.

   (xx) ______ Every dynamic program problem can be worked by polynomially many processors in polylogarithmic time.

   (xxi) ______ If an abstract Pascal machine can perform a computation in polynomial time, there must be some Turing machine that can perform the same computation in polynomial time.

(xxii) ________ Let L be any undecidable $\mathcal{RE}$ language, and let M_L be a machine which accepts L. For any string $w \in M$, let $F_L(w)$ be the number of steps M_L executes, if its input is w. If $w \notin M$, let $F_L(w) = 0$. Now define $T_L(n) = \{\max F(w) \,:\, w \in \Sigma^* \text{ and } |w| = n\}$. Then T_L is recursive.

(xxiii) ________ There is a polynomial time reduction of the subset sum problem to 2SAT.

(xxiv) ________ Every $\mathcal{P}$–TIME dynamic programming problem has an $\mathcal{NC}$ reduction to CVP.

(xxv) ________ If a language L is accepted by a non-deterministic machine, then L must be accepted by some deterministice machine.

(xxvi) ________ The language $\{a^n b^n c^n \,:\, n \geq 0\}$ is $\mathcal{NC}$.

(xxvii) ________ The set of all languages over the binary alphabet is countable.

(xxviii) ________ No set is larger than $\mathbb{R}$, the set of real numbers.

(xxix) ________ The furniture mover's problem is known to be $\mathcal{NP}$-complete. (Given a room with a door and given a collection of furniture that must be put into the room, can all the furniture be moved into the room through the door?)

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

(xxxi) ________ Let $L = \{(G_1, G_2)\} \,:\, G_1$ and G_2 are not equivalent. Then L is recursively enumerable.

(xxxii) ________ The factoring problem for unary numerals is $\mathcal{P}$–TIME

(xxxiii) ________ The set of all binary numerals for prime numbers is in $\mathcal{P}$–TIME.

(xxxiv) ________ If L is a recursively enumerable language, there must be a machine which enumerates L in canonical order.

(xxxv) ________ The set of all positive real numbers is countable.

(xxxvi) ________ Let L be a recursive language over an alphabet Σ, and M a machine that decides L. For any n, let $F(n)$ be the maximum number of steps M needs to decide whether a given string in Σ^* of length n is in L. Then F must be recursive.

(xxxvii) ________ Let L be a recursively enumerable language over an alphabet Σ, and M a machine that accepts L. For any n, let $G(n)$ be the maximum number of steps M needs to accept any string in L of length n. Then G must be recursive.

(xxxviii) ________ For any alphabet Σ, the set of all recursively enumerable languages over Σ is countable.

(xxxix) ________ If L is a context-free language over the unary alphabet, then L must be regular.

(xl) ________ The union of any two undecidable languages is undecidable.

(xli) ________ co-$\mathcal{P}$–TIME= $\mathcal{P}$–TIME.

(xlii) ________ Every finite language is decidable.

(xliii) ________ Every context-free language is in Nick's class.

(xliv) ______ 2SAT is known to be $\mathcal{NP}$-complete.

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

(xlvi) ______ The complement of any $\mathcal{P}$–SPACE language is $\mathcal{P}$–SPACE.

The jigsaw puzzle problem is, given a set of various polygons, and given a rectangular table, is it possibe to assemble those polygons to exactly cover the table?

(xlvii) ______ The jigsaw puzzle problem is known to be $\mathcal{NP}$ complete.

(xlviii) ______ The jigsaw puzzle problem is known to be $\mathcal{P}$–SPACE complete.

(xlix) ______ The furniture mover's problem is known to be $\mathcal{P}$–SPACE complete.

(l) ______ The complement of any recursive language is recursive.

(li) ______ The complement of any undecidable language is undecidable.

(lii) ______ Every undecidable language is either $\mathcal{RE}$ or co-$\mathcal{RE}$.

(liii) ______ For any infinite countable sets A and B, there is a 1-1 correspondence between A and B.

(liv) ______ A language L is recursively enumerable if and only if there is a machine which accepts L.

(lv) ______ Every $\mathcal{NP}$ language is reducible to the independent set problem in polynomial time.

(lvi) ______ If a Boolean expression is satisfiable, there is a polynomial time proof that it is satisfiable.

(lvii) ______ If a language L is recursively enumerable, there is a proof that L is recursively enumerable.

(lviii) ______ If a language L is co-$\mathcal{RE}$, there is a proof that L is co-$\mathcal{RE}$

(lix) ______ The Post correspondence problem is undecidable.

2. Fill in the blanks.

   (a) If L_1 is $\mathcal{NP}$–complete and L_2 is $\mathcal{NP}$, and there is a polynomial time reduction of L_1 to L_2, then L_2 must be ______________________.

   (b) The class of RE languages is generated by the class of __________________ grammars.

   (c) A language is __________________ if and only if it is both RE and co-RE.

   (d) The class of push-down-automata accepts the class of __________________ languages.

   (e) The class of Turing machines accepts the class of ______________________ languages.

   (f) A language L is ______________________ if and only if there is a machine which enumerates L in canonical order.

3. Here is a list of problems or languages. For each problem, enter **T** if it is known to be $\mathcal{NP}$–complete, **F** if it is not known to be $\mathcal{NP}$–complete.

   (a) ______ SAT

   (b) ______ 2-SAT

   (c) ______ 3-SAT

(d) ________ 4-SAT

(e) ________ Rush Hour

(f) ________ The Boolean circuit problem.

(g) ________ Integer factoring, using binary numerals.

(h) ________ Tiling, i.e., covering a big polygon exactly with the members of a set of smaller polygons.

(i) ________ Given a room with a door and some pieces of furniture, move them all into the room through the door.

(j) ________ Given a set of trucks, each with a given capacity, and given a set of items, can all the items fit into the trucks?

4. Give a definition of "unrestricted grammar."

5. Give a definition of the class $\mathcal{P}$–SPACE.

6. Give a context-sensitive grammar for $L = \{a^n b^n c^n \ : \ n \geq 1\}$.

7. Consider the following CF grammar G with start symbol E, and an LALR parser for G. The grammar generates algebraic expressions, where the only operations are subtraction and multiplication. The precedence of operators is as used in algebra and in programming languages.

1. $E \to E -_2 E_3$
2. $E \to E *_4 E_5$
3. $E \to \mathbf{id}_6$

	id	$-$	$*$	\$	E
0	$s6$				1
1		$s2$	$s4$	HALT	
2					
3					
4	$s6$				5
5		$r2$	$r2$	$r2$	
6					

(a) The entries in row 2 are left blank. Fill them in.

(b) The entries in row 3 are left blank. Fill them in.

(c) The entries in row 6 are left blank. Fill them in.

8. Explain how it is possible to find the maximum of an array of n integers in $O(\log n)$ time using $O(n/logn)$ processors. You don't need to draw a diagram, but it might help.

9. Prove that the halting problem is undecidable.

10. Which of the following conditions is true if and only if a real number x is recursive? Yes, No, or Open for each.

(a) ________ There is a program which, given n, finds the n^{th} digit after the decimal point of the decimal expansion of x.

(b) ________ There is a machine which, given any positive integer q, computes an integer p such that $\frac{p}{q} \leq x < \frac{p+1}{q}$.

(c) ________ There is a polynomial P with integral coefficients such that $P(x) = 0$. (For example: $5x^3 - 2x^2 + 9x - 4$ is a polynomial with integral coefficients.)

(d) ________ There is a mathematical definition of x.

11. Prove that every recursively enumerable language is accepted by some deterministic machine.

12. (a) What is a one-way function?

    (b) Does any one-way function exist?

13. Prove that the halting problem is undecidable.

14. Determine whether the following Boolean expression is satisfiable. If so, give a satisfying assignment.

    $(a+b)*(a+c)*(!a+e)*(!b+d)*(!c+!d)*(!d+!e)$

15. Using the fact that 3SAT is $\mathcal{NP}$-complete, prove that the independent set problem is $\mathcal{NP}$-complete.

16. State the Church Turing thesis. Why is it important?

17. Prove that every language which can be enumerated in canonical order by some machine is recursive.

18. Prove that every recursive language can be enumerated in canonical order by some machine.

19. Prove that every recursively enumerable language is accepted by some machine.

20. Prove that every language accepted by a machine is recursively enumerable.

21. Why is the question of whether $\mathcal{NC} = \mathcal{P}$–TIME so important nowadays?

22. Prove that every regular languge is $\mathcal{NC}$.

23. Prove that the halting problem is undecidable. Do not quote any theorem or lemma from the handouts.

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

25. 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.
    ○ ○ Subset sum problem.
    ○ ○ Generalized checkers, i.e. on a board of arbitrary size.
    ○ ○ Traveling salesman problem.
    ○ ○ Rush Hour: https://www.youtube.com/watch?v=HI0rlp7tiZ0
    ○ ○ Dominating set problem.
    ○ ○ Strong connectivity of directed graphs.
    ○ ○ Circuit value problem, CVP.
    ○ ○ C++ program equivalence.
    ○ ○ Partition.
    ○ ○ Regular language membership problem.
    ○ ○ Block sorting.

26. State the pumping lemma for regular languages.

27. State the pumping lemma for context-free languages.

28. Give a polynomial time reduction of 3SAT to the independent set problem. (Pictures help.)

29. Prove that any recursively enumerable language is accepted by some machine.

30. Prove that any recursive language can be enumerated in canonical order by some machine.

31. 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. $E \to x_{13}$

   (a) Below are the tables of an LALR parser for G. Fill in the missing colums.

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

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

32. 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
$\mathcal{NC}$					
context-free					
$\mathcal{NP}$					
recursive					
co-$\mathcal{RE}$					
undecidable					

33. Consider the following well-known complexity classes:

$\mathcal{NC} \subseteq \mathcal{P}$–TIME$\subseteq \mathcal{NP} \subseteq \mathcal{P}$–SPACE $\subseteq$**EXP**–TIME $\subseteq$**EXP**–SPACE

   (a) Which of the above complexity classes is the smallest class which is known to contain SAT, the Boolean satisfiability problem?

   (b) Which of the above complexity classes is the smallest class which is known to contain the connectivity problem for graphs?

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

   (d) Which of the above complexity classes is the smallest class which is known to contain every sliding block problem?

   (e) Which of the above complexity classes is the smallest class which is known to contain integer matrix multiplication?

   (f) We say that a computer program is straight-line if no portion of the code can be executed more than once. That implies that the code contains no loops or recursion, and no GOTO from one line of the code to an earlier line. Which of the above complexity classes is the smallest class which contains the problem of determining whether the output of a straight-line program is zero?

34. Let L be the simple algebraic language with three operators, subtraction, multiplication, and negation, and with only one variable. Write an annotated context-free grammar for L, annotated with stack states as in the handout, and write the ACTION and GOTO tables for an LALR parser.