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

CSC 456/656 Fall 2024 Answers to Third Examination April 10, 2024

Name:____

The entire test is 365 points.

A binary function is defined to be a function F on binary strings such that, for each binary string w, $F(w)$ is a binary string. (Of course, the strings could be numerals.)

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.

   (i) **T** Every language is a subset of some regular language.

   (ii) **O** If L is any $\mathcal{P}$–TIME language, there is an $\mathcal{NC}$ reduction of the Boolean circuit problem (CVP) to L.

   (iii) **T** The concatenation of any two RE languages is RE.

   (iv) **F** Every function that can be mathematically defined is bounded by some recursive function.

   (v) **T** There are uncountably many languages over the binary alphabet.

   (vi) **F** There are uncountably many RE languages over the binary alphabet.

   (vii) **T** There is an algorithm which determines whether a given list of n positive integers has a sublist whose total is a given number S.

   (viii) **T** 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.

   (ix) **T** Nick's Class is closed under Kleene closure.

   (x) **T** Any computation that can be done by any machine can be done by some Pascal program.

   (xi) **T** Multiplication of integer matrices is $\mathcal{NC}$.

   (xii) **T** There is an $\mathcal{P}$–SPACE algorithm which decides SAT.

   (xiii) **O** Every dynamic program problem can be worked by polynomially many processors in polylogarithmic time.

   (xiv) **F** 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. Now define $T_L(n) = \{\max F_L(w) : w \in L \text{ and } |w| = n\}$. Then T_L is recursive.

   (xv) **O** There is a polynomial time reduction of the subset sum problem to 2-SAT.

   (xvi) **T** Every $\mathcal{P}$–TIME problem has an $\mathcal{NC}$ reduction to the Circuit Value Problem.

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

(xviii) **T** Every $\mathcal{NC}$ language is context-free.

(xix) **T** The language $\{a^nb^nc^nd^n : n \geq 0\}$ is $\mathcal{NC}$.

(xx) **F** The set of all languages over the binary alphabet is countable.

(xxi) **T** The context-free grammar equivalence problem is co-$\mathcal{RE}$.

(xxii) **T** The set of all binary numerals for prime numbers is $\mathcal{P}$–TIME.

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

(xxiv) **F** The union of any two undecidable languages is undecidable.

(xxv) **T** co-$\mathcal{P}$–TIME= $\mathcal{P}$–TIME.

(xxvi) **O** There exists a one-way function.

(xxvii) **T** The complement of any $\mathcal{P}$–SPACE language is $\mathcal{P}$–SPACE.

(xxviii) **T** The jigsaw puzzle problem is $\mathcal{NP}$ complete.

(xxix) **T** The furniture mover's problem is $\mathcal{P}$–SPACE complete.

(xxx) **T** The complement of any undecidable language is undecidable.

(xxxi) **T** If a Boolean expression is satisfiable, there is a polynomial time proof that it is satisfiable.

(xxxii) **T** If there is a recursive reduction from the halting problem to L, then L must be undecidable.

(xxxiii) **F** If L is undecidable, there must be a recursive reduction from the halting problem to L.

2. Fill in the blanks.

   (a) [5 points] 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 $\mathcal{NP}$–COMPLETE.

   (b) [5 points] A language is **decidable** or **recursive**. if and only if it is both RE and co-RE.

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. [5 points each]

   (a) **T** SAT

   (b) **F** 2-SAT

   (c) **T** 3-SAT

   (d) **F** Rush Hour

   (e) **F** The Boolean circuit problem.

   (f) **F** Integer factoring, using binary numerals.

   (g) **T** The tiling problem.

   (h) **F** The furniture mover's problem.

(i) **T** The bin packing problem. Given a set of bins, each with a given capacity, and given a set of items, can all the items be packed into the bins?

4. [20 points] Explain how to find the maximum of a list of n integers in logarithmic time using n processors.

partition the list into pairs. If there is one left over, call it a pair. Find the maximum of each pair in $O(1)$ time, using n processors in parallel, All these maxima form a new list half as long as the original. Repeat the process until there is only one number left. That will be the maximum of the original list.

6. [20 points] Give a definition of a recursive real number. (There is more than one correct definition.)

Each of these is a correct answer. A real number s is recursive if:

- There is a program that runs forever, writing the decimal expansion of x.
- There is a program that, given an integer n, writes the n^{th} digit of the decimal expansion of x.
- There is a program that, given a rational number q, decides whether $q < x$.
- There is a program that, given a positive integer q, writes the integer p such that $\frac{p}{q} \leq x < \frac{p+1}{q}$

Those are not the only ones.

7. [20 points] State the pumping lemma for context-free languages.

8. [20 points] Let G be the following context-free gammar with start symbol E. Stack states are indicated. An LALR parser for G is given below. Give a complete computation of the parser if the input string is $x - x * -(-x + x)$.

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. $E \to x_{13}$

	x	$+$	$-$	$*$	(	)	\$	E
0	s13		s8		s10			1
1		s2	s4	s6			**halt**	
2	s13		s8		s10			3
3		r1	r1	s6		r1	r1	
4	s13		s8		s10			5
5		r2	r2	s6		r2	r2	
6	s13		s8		s10			7
7		r3	r3	r3		r3	r3	
8	s13		s8		s10			9
9		r4	r4	r4		r4	r4	
10	s13		s8		s10			11
11		s2	s4	s6		s12		
12		r5	r5	r5		r5	r5	
13		r6	r6	r6		r6	r6	

$\$_0$	$x - x * -(-x + x)$		
$\$_0 x_{13}$	$-x * -(-x + x)$		$s13$
$\$_0 E_1$	$-x * -(-x + x)$	6	$r6$
$\$_0 E_1 -_4$	$x * -(-x + x)$	6	$s4$
$\$_0 E_1 -_4 x_{13}$	$* - (-x + x)$	6	$s13$
$\$_0 E_1 -_4 E_5$	$* - (-x + x)$	66	$r6$
$\$_0 E_1 -_4 E_5 *_6 -_8$	$(-x + x)$	66	$s8$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10}$	$-x + x)$	66	$s10$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} -_8$	$x + x)$	66	$s8$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} -_8 x_{13}$	$+x)$	66	$s13$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} -_8 E_9$	$+x)$	666	$r6$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11}$	$+x)$	6664	$r4$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11} +_2$	$x)$	6664	$s2$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11} +_2 x_{13}$	$)$	6664	$s12$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11} +_2 E_3$	$)$	66646	$r6$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11}$	$)$	666461	$r1$
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11})_{12}$		666461	$s12$
$\$_0 E_1 -_4 E_5 *_6 -_8 E_9$		6664615	$r5$
$\$_0 E_1 -_4 E_5 *_6 E_7$		66646154	$r4$
$\$_0 E_1 -_4 E_5$		666461543	$r3$
$\$_0 E_1$		6664615432	$r2$

halt

9. 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) [5 points] Which of the above complexity classes is the smallest class which is known to contain SAT, the Boolean satisfiability problem? $\mathcal{NP}$

   (b) [5 points] Which of the above complexity classes is the smallest class which is known to contain the connectivity problem for graphs? $\mathcal{P}$–TIME

   (c) [5 points] Which of the above complexity classes is the smallest class which is known to contain the context-free language membership problem? $\mathcal{NC}$

   (d) [5 points] Which of the above complexity classes is the smallest class which is known to contain every sliding block problem? $\mathcal{P}$–SPACE

   (e) [5 points] Which of the above complexity classes is the smallest class which is known to contain integer matrix multiplication? $\mathcal{NC}$

10. [20 points] Give the verifier definition of the class $\mathcal{NP}$. A language L is $\mathcal{NP}$ if and only if there is some machine M and some integer k such that:

   (a) For any string $w \in L$ there is a string c such that M accepts (w, c) within n^k steps, where $n = |w|$

   (b) For any string $w \notin L$ M does not accept (w, c) for any string c.

11. [20 points] Prove that the halting problem is undecidable.

    The proof is in Tests/stdy3ans.pdf.