University of Nevada, Las Vegas Computer Science 456/656 Spring 2021

Assignment 6: Due Saturday November 12 2022, 11:59 PM

This is the final version of the assignment.

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. Turn in the assignment in the manner given to you by our grader, Janeen Sudiacal.

1. True, false, or open:

   (a) ______ If L_1 and L_2 are languages and there is a $\mathcal{P}$–TIME reduction of L_1 to L_2, and if there is a machine that accepts L_2 in polynomial time, then there must be a machine that accepts L_1 in polynomial time.

   (b) ______ If L_1 and L_2 are languages and there is a $\mathcal{P}$–TIME reduction of L_1 to L_2, and if there is a machine that accepts L_1 in polynomial time, then there must be a machine that accepts L_2 in polynomial time.

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

   (d) ______ If L_1 and L_2 are languages and there is a recursive reduction of L_1 to L_2, and if L_1 is undecidable, then L_2 is undecidable.

   (e) ______ Context-free grammar equivalence is co-RE.

   (f) ______ The set of binary numerals for prime numbers is $\mathcal{P}$–TIME.

   (g) ______ The factoring problem for binary numerals is $\mathcal{P}$–TIME.

   (h) ______ The class of regular languages is closed under intersection.

   (i) ______ co-$\mathcal{P}$–TIME = $\mathcal{P}$–TIME.

   (j) ______ The class of context-free languages is closed under union.

   (k) ______ The class of context-free languages is closed under intersection.

   (l) ______ The complement of any undecidable language must be undecidable.

   (m) ______ Every context-free language can be parsed by an LALR parser.

   (n) ______ If a function $f : \mathcal{N} \to \mathcal{N}$, where $\mathcal{N}$ is the set of natural numbers, has a mathematical definition, then f must be recursive.

   (o) ______ If Σ is any alphabet, the set of all languages over Σ is countable.

   (p) ______ $\mathcal{P} = \mathcal{NP}$.

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

(r) ______ Every language has a canonical order enumeration.

(s) ______ $\sqrt{2}$ is a recursive real number.

(t) ______ There is a $\mathcal{P}$–TIME algorithm which determines whether a given weighted directed graph has a Hamiltonian cycle whose total weight is no greater than a given number. (Given a directed graph G, a Hamiltonian cycle of G is a directed cycle in G that includes every vertex of G exactly once.)

(u) ______ It is known that 2-SAT is $\mathcal{NP}$-complete.

(v) ______ Every context-free language is in Nick's Class.

(w) ______ Context-free grammar equivalence is co-RE.

(x) ______ Every context-free language is accepted by some LALR parser.

(y) ______ The Circuit Value Problem (CVP) is $\mathcal{NC}$.

(z) ______ If L is any $\mathcal{P}$–TIME language, there is an $\mathcal{NC}$ reduction of CVP to L.

2. Correctly state (do not prove) the pumping lemma for context-free languages.

3. Describe a Nick's Class algorithm which finds the maximum of n integers in $O(\log n)$ time, using $n/\log n$ processors.

We have proved, in class (or we will have by the end of class on November 7) that a dynamic programming problem is $\mathcal{NC}$ if it has logarithmic reachback and every subprogram is $\mathcal{P}$–TIME. Note that the circuit value problem does not have logarithmic reachback, since an arc can stetch from a starting gate all the way to the final gate. You can use this fact to work problems 4 and 5 below. (Read the handout NC.pdf.)

4. Prove that addition of binary numerals is $\mathcal{NC}$. (Hint: This is important for computer architecture.)

5. Prove that every regular language is $\mathcal{NC}$.

6. Prove that every decidable language can be enumerated in canonical order by some machine.

7. (a) State the Church-Turing thesis.

(b) Why is the Church-Turing thesis important?

8. The CF grammar given below generates $L = \{a^n b^m : n, m \geq 0\}$. S is the start symbol, and there are two other variables, A and B. An LALR parser for that grammar is also given. Walk through the computation of the parser for the input string aab.

1. $S \to A_2 B_4$
2. $A \to A_2 a_3$
3. $A \to \lambda$
4. $B \to B_4 b_5$
5. $B \to \lambda$

	ACTION			GOTO		
	a	b	$\$$	S	A	B
0	$r3$	$r3$	$r3$	1	2	
1			HALT			
2	$s3$	$r5$	$r5$			4
3	$r2$	$r2$	$r2$			
4		$s5$	$r1$			
5		$r4$	$r4$			