University of Nevada, Las Vegas Computer Science 456/656 Spring 2026

Assignment 5: Due Saturday April 11, 2026, 11:59:59 PM

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. Turn in the assignment as instruced by the Graduate Assistant, Shubhashish Kar, `shubhashish.kar@unlv.nevada.edu`

1. True, False, or Open.
   - (i) **T** If L is a $\mathcal{P}$–TIME language, there is an $\mathcal{NC}$ reduction of L to CVP.
   - (ii) **O** $\mathcal{P} = \mathcal{NC}$.
   - (iii) **T** Every $\mathcal{NC}$ language is decided by some polynomial time program.
   - (iv) **T** In modern computers, addition is done in parallel using a built-in $\mathcal{NC}$ algorithm.
   - (v) **F** The set of all binary languages is countable.
   - (vi) **F** The limit of every convergent recursive sequence of rational numbers is a recursive real number. (A sequence of rational numbers is *recursive* if it can be printed by a program, which runs forever, of course.)

2. Explain why the $\mathcal{P} = \mathcal{NC}$ problem is of such practical importance today.

   Multiprocessor computers are getting common nowadays. It is important to know whether a given large sequential program can be efficiently parallelized.

3. State the pumping lemma for regular languages. For any regular language L
   There exists a positive integer p (called the pumping length of L) such that
   For any string $w \in L$ of length at least p
   There exist string x, y, z such that the following four statements hold
   1. $w = xyz$
   2. $|xy| \leq p$
   3. $|y| > 0$
   4. For any $i \geq 0$ $xy^iz \in L$

4. Use the pumping lemma for regular languages to prove that $L = \{a^n b^n : n \geq 0\}$ is not a regular language.

   Assume that L is regular. Then the pumping lemma holds for L. Let p be the pumping length of L.
   Pick $w = a^p b^p$ which is in L.
   Then there exist strings x, y, z such that the four statements of the pumping lemma hold.
   By statement 1, $xyz = a^p b^p$
   By statement 2, $|xy| \leq p$. It follows that xy is a substring of a^p, hence $y = a^k$ for some k.
   By statement 3, $k > 0$.
   By statement 4 for $i = 0$, $s = xz \in L$, meaning that $s = a^m b^m$ for some m
   However, $\#_b(s) = p$ and $\#_a(s) = p - k < p$, contradiction.

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

   For any context-free language L
   There exists a positive integer p (called the pumping length of L) such that
   For any string $w \in L$ of length at least p
   There exist string u, v, x, y, z such that the following four statements hold
   1. $w = uvxyz$
   2. $|vxy| \leq p$
   3. $|v| + |y| > 0$
   4. For any $i \geq 0$ $uv^ixy^iz \in L$

6. Use the pumping lemma for context-free languages to prove that $L = \{a^n b^n c^n \,:\, n \geq 0\}$ is not a context-free language.

   Assume that L is context-free. Then the pumping lemma holds for L. Let p be the pumping length of L.
   Pick $w = a^p b^p c^p$, which is in L.
   Then there exist strings u, v, x, y, z such that the four statements of the pumping lemma hold.
   By statement 1, $uvxyz = a^p b^p c^p$
   By statement 2, $|vxy| \leq p$. It follows that vxy is a substring of w which is not long enough to contain both a and c. That is, either $\#_a(vxy) = 0$ or $\#_c(vxy) = 0$. By statement 3, $|v| + |y| = k > 0$. By statement 4, letting $i = 0$, $s = uxz \in L$, Thus $|s| = 3p - k < 3p$. If $\#_a(vxy) = 0$, $\#_a(s) = p$ which implies that $|s| = 3p$, contradiction. On the other hand, If $\#_c(vxy) = 0$, $\#_c(s) = p$ which implies that $|s| = 3p$, In either case, we have a contradiction, hence L is not context-free.

7. (Hard.) Prove that the factoring problem for integers is $\mathcal{NP}$.

   We use the verification definition of $\mathcal{NP}$. Suppose $(\langle a\rangle\langle n\rangle)$ is an instance of the factoring program which has a solution, namely a numeral $\langle m\rangle$ such that $a \leq m \leq n$ and m is a divisor of n. Then $\langle m\rangle$ itself is a certificate for that instance. We conclude that the factoring problem is $\mathcal{NP}$.

8. (Harder.) Prove that primality is $co - \mathcal{NP}$. Attach extra sheets. I'm sure this is on the internet.[1]

   Let L be the complement of L_{prime}. We can assume that L is the set of numerals for all composite positive integers. We use the verification method to prove that L is $\mathcal{NP}$. If n is a composite number, a numeral $\langle m\rangle$ exists such that $1 < m < n$ and m is a divisor of n, and the numeral $\langle m\rangle$ is a certificate.

9. (Even harder.) Prove that the context-free grammar equivalence problem is co-$\mathcal{RE}$. Attach extra sheets. I am not sure this is on the internet.

   The context-free grammar equivalence problem is co-$\mathcal{RE}$ if and only if the language $L = \{(G_1, G_2) \,:\, L(G_1) \neq L(G_2)\}$ is $\mathcal{RE}$. (Recall that a language L is $\mathcal{RE}$ if and if, for every $w \in L$, there is a proof that $w \in L$.) If $(G_1, G_2) \in L$, there is some string w which is a member of $L(G_1)$ but not $L(G_2)$, or vice-versa. The membership problem of any context-free language is decidable, and thus it can be verified that $w \in L(G_1)$ but not in $L(G_2)$ (or vice-versa).

[1] The following proof is NOT acceptable: "Primality is known to be $\mathcal{P}$, and every $\mathcal{P}$ problem is co-$\mathcal{NP}$." Reason: you are not allowed to use the knowlege that primality is $\mathcal{P}$–TIME.

10. Prove that the halting problem is undecidable. (This will be on the next exam.)

There is a video at `https://www.youtube.com/watch?v=u_5erLilDXY` that contains a very nice proof of the undecidability of the halting problem, One thing you learn from that video is that humans can beat AI at some tasks. The video also contains an error, which anyone who's taken this course should be able to catch.

Here is my proof. Assume that the halting problem is decidable, that is, that there is a program H such that, $H(P, w)$ is **true** if the rogram P hallts with input w, and **false** otherwise. Let Q the following program which takes a program P as its argument:

$Q(P)$
If $H(P, P)$
    Loop forever
Else
    Halt

Does Q halt with input Q? If $H(Q, Q)$, then $Q(Q)$ runs forever, contradiction. Otherwise, if $H(Q, Q)$ is false, then $Q(Q)$ halts, contradiction.

We conclude that the H cannot exist, meaning that the halting problem is undecidable.