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

Answers to Assignment 6: Due Saturday November 9, 2024, 11:59 PM

1. True/False/Open

   (i) **T** Every regular language is $\mathcal{NC}$.

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

   (iii) **O** $\mathcal{NC} = \mathcal{P}$–TIME

   (iv) **T** Given polynomially many processors, the product of two n-bit binary numerals can be computed in polylogarithmic time.

   (v) **T** The asymptotic time complexity of a sequential algorithm cannot be less than its asymptotic space complexity.

   (vi) **F** The asymptotic space complexity of a sequential algorithm cannot be less than its asymptotic time complexity.

   (vii) **T** In fourth grade (or whenever) you learned how to multiply two n-digit decimal numerals in $O(n^2)$ time and space. But in the best computers nowadays, multiplication of integers is worked faster, thanks to parallel circuitry and microprogramming.

   (viii) **T** Matrix multiplication with integral entries is $\mathcal{NC}$.

   (ix) **T** Every finite language is decidable.

   (x) **F** If a theorem can be stated using n bits of text and is provably true, then it must have a proof which has length polynomial in n.

2. Give a $\mathcal{P}$–TIME reduction of the subset sum problem to the partition problem.

   Let $X = (x_1, x_2, \ldots x_n, K)$ be an instance of the subset sum problem. Let $S = x_1 + x_2 + \cdots + x_n$. We can assume $K \leq S$, since otherwise X cannot have a solution. Let $Y = (x_1, x_2, \ldots x_n, K{+}1, S{-}K{+}1)$, an instance of the partition problem. Then Y has a solution if and only if X has a solution. The mapping from X to Y is a polynomial time reduction.

3. Give a $\mathcal{P}$–TIME reduction of 3-SAT to the independent set problem.

   Let $E = C_1 * C_2 * \cdots * C_k$ be an instance of 3-SAT. That is, $C_i = t_{i,1} + t_{i,2} + t_{i,3}$ for each clause C_i, where each $t_{i,j}$ is either a Boolean variable or the negation of a Boolean variable.

   The reduction maps E to (G, k), an instance of the independent set problem, where G is a graph. G has $3k$ vertices, $V = \{v_{i,j}\}_{1 \leq i \leq k,\, 1 \leq j \leq 3}$. G has an edge connecting distinct vertices $v_{i,j}$ and $v_{i',j'}$ if and only if either $i = i'$ or $t_{i,j} * t_{i',j'}$ is a contradiction. Then E has a satisfying assignment if and only if G has an independent set of order k.

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

   Proof by contradiction. Assume L is regular. Let p be the pumping length of L. Let $w = a^p b^p$, which is in L and has length at least p. Then there must exist strings x, y, and z such that
   1. $w = xyz$
   2. $|xy| \leq p$

3. $|y| \geq 1$
4. For any $i \geq 0$ $xy^iz \in L$.

Let $i = 0$. By 1. and 2., xy must be a substring of a^p. Thus, y consists entirely of a's. By 3., $y = a^k$ for some integer $k \geq 1$. By 4., $xz \in L$. But xz is obtained from w by deleting a^k. Thus, xz has fewer a's than b's, and hence cannot be a member of L. Contradiction. We conclude that L is not regular.

5. State the Church Turing thesis.

   Any computation that can be done by any machine can be done by some Turing machine.

6. Prove that $\sqrt{3}$ is irrational.

   Proof by contradiction. Assume $\sqrt{3}$ is rational. Thus $\sqrt{3} = \dfrac{p}{q}$ for integers p and q which have no common divisor. Square both sides, and we get $3 = \dfrac{p^2}{q^2}$. Multiply both sides by q^2. We have $3q^2 = p^2$, hence p^2 is divisible by 3, hence p is divisible by 3, *i.e.,* $p = 3k$ for some integer k. We have $3q^2 = 9k^2$, thus $q^2 = 3k^2$, hence q^2 is divible by 3, hance q is divisibly by 3. But p and q have no common divisor, contradiction. We conclude that $\sqrt{3}$ is not rational.

7. Give a definition of an *instance* of the halting problem.

   A string of the form $\langle M \rangle w$, where $\langle M \rangle$ is a description of a machine M, and w is a string. That instance is a member of HALT if and only if M halts with input w.

8. Prove that the halting problem is undecidable.

   The proof is by contradiction. Assume that HALT is decidable. Let P be the following program (machine).

   Read a machine descritpion, $\langle M \rangle$.
   If (M halts with input $\langle M \rangle$) enter an infinite loop.
   Else halt.

   This program is valid because HALT is decidable, which implies that the condition can always be evaluated.

   We now ask whether P halts with input $\langle P \rangle$. One of the following two cases must be true.

   Case 1: P halts with input $\langle P \rangle$. When we run the program P with input $\langle P \rangle$, the value of the condition is True, after which P enters an infinite loop and does not halt. That is, P does not halt with input $\langle P \rangle$, contradiction.

   Case 2: P does not halt with input $\langle P \rangle$. When we run the program P with input $\langle P \rangle$, the value of the condition is False, after which P halts. That is, P halts with input $\langle P \rangle$, contradiction.

   We have a contradiction in either case, and thus we have a contradiction. It follows that the assmption that HALT is decidable is incorrect, that is, HALT is undecidable.

9. If you are parallelizing a sequential algorithm, it would be desirable to create a parallel algorithm with polylogarithmic time complexity, but without increasing the total work. There is an obvious sequential algorithm for the maxarray problem, *i.e.,* finding the maximum of an array of n numbers, which takes $O(n)$ time. In class I gave you a logarithmic time algorithm for the same problem which uses $O(n)$ processors. Thus, its work complexity is $O(n \log n)$, which is greater than the $O(n)$ work complexity of the sequential algorithm.

   Find a parallel algorithm for the maxarray problem whose time complexity is $O(\log n)$ and uses $\frac{n}{\log n}$ processors, and thus has work complexity $O(n)$.

   This problem is hard. It is not hard to describe, but it is hard to think of. I am not sure anyone will get it. (Hint: it's "out there.")

10. On the handout, I gave a proof that the halting problem is decidable. Of course, that proof is incorrect. The flaw in the proof is that I made a hidden (not explicitly stated) assumption that is false. What is that hidden assumption?

    I will not give the solution. Resolution of this "paradox" is a noteworthy achievement for a 456 or 656 student. Hint: If you write more than one line of text, your answer is not correct.