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

Assignment 5. Due Wednesday October 13, 2021

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet.

1. Indicate by "Yes" or "No" which of the following languages, or problems, are known to be $\mathcal{NP}$ complete. (You may have to do some research.)

   No 2-SAT
   Yes 3-SAT
   Yes 4-SAT
   Yes Partition
   Yes Subset Sum
   Yes Traveling Salesman
   No Regular Expression Equivalence
   Yes Vertex Cover
   No Rush Hour (the puzzle)
   No Linear Programming
   Yes Integer Programming
   No Membership Problem for a Context-Free Grammar
   No Membership Problem for a Context-Sensitive Grammar
   Yes Block Sorting

   Regular expression equivalence, context-sensitive membership, and sliding block puzzles (such as Rush Hour) are known to be $\mathcal{P}$-SPACE complete.

   Linear programming (using real variable) is only recently known to be $\mathcal{P}$-TIME, although the commonly used simplex algorithm takes exponential time in the worst case. Integer prograsmming, *i.e.* linear programming with all integer variables, is known to be $\mathcal{NP}$-complete.

   The membership problem for a context-free grammar can be solved in $\mathcal{P}$-TIME using the CYK algorithm. (It has recently been shown that CF grammar membership in in Nick's Class.)

   Block sorting is listed as an $\mathcal{NP}$-complete problem on a Wikipedia page. The problem is, given a list (say, a list of names in a text-file) we define *block move* to be a move which means a block of consecutive item to another place in the list. The question is whether a given list can be alphabetized within k block moves.

2. Let R be the reduction of 3-SAT to the independent set problem as I defined it in class. Let E be the 3-CNF expression given below.

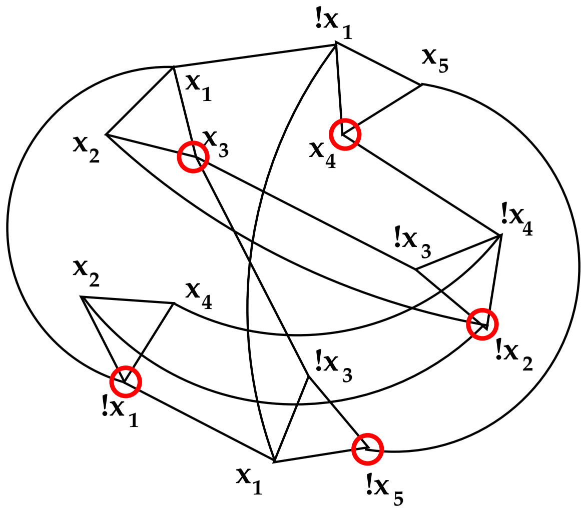
   $(x_1 + x_2 + x_3) \cdot (x_4 + !x_1 + x_5) \cdot (!x_2 + !x_4 + !x_3) \cdot (!x_3 + !x_5 + x_1) \cdot (x_2 + !x_1 + x_4)$

   Let $(G, k) = R(E)$.

   (a) What is k? 5.

(b) Sketch G.

(c) Circle an independent set of k vertices of G.



(d) Give a satisfying assignment of the variables of E corresponding to that independent set.

$x_1 = 0$
$x_2 = 0$
$x_3 = 1$
$x_4 = 1$
$x_5 = 0$

3. Prove that a language L is recursively enumerable if and only if there is a machine M which accepts L. (Recall that you may assume that M is a program.)

Theorem: If L is recursively enumerable, there is a machine that accepts L.

Proof: Let $w_1, w_2, \ldots$ be a recursive enumeration of L. That is, w_i can be computed for each i, and every member of L is w_i for some i. The following program accepts L.

Read a string w.
For each i from 1 to ∞:
If($w = w_i$) Accept and Halt.

If $w \in L$, the program halts. Otherwise, it runs forever.

Theorem: If there is a machine that accepts L, then L is recursively enumerable.

Proof: Let M be a machine that accepts L. Any computation of M consists of a sequence of steps. Let $w_1, w_2, \ldots$ be an enumeration of Σ^*. We know this enumeration exists, since Σ^* is recursively enumerable. The following program enumerates L.

For all t from t to ∞
For i from 1 to t.
If (M accepts w_i within t steps) write w_i.