University of Nevada, Las Vegas Computer Science 456/656 Spring 2023

Answers to Assignment 5: Due Saturday April 1, 2023, 23:59

1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

   (i) **T** Every $\mathcal{NP}$ language is decidable.

   (ii) **F** Every sliding block problem is $\mathcal{NP}$-complete.

   (iii) **F** Every undecidable language is $\mathcal{NP}$-complete.

   (iv) **T** If the Boolean circuit problem is $\mathcal{NC}$, then $\mathcal{P} = \mathcal{NC}$.

   (v) **T** If L is any $\mathcal{NP}$ language, and if $w \in L$, there must be a polynomial time proof that $w \in L$.

   (vi) **T** Some sliding block problems are $\mathcal{P}$–SPACE complete.

2. For each of these problems, or languages, which ones are known to be $\mathcal{NP}$-complete? Write **Yes** if it is known to be $\mathcal{NP}$-complete, **No** otherwise. You may need to do some research. Note: **Open** is not an answer.

   (i) **Yes** The firehouse problem. Given a graph G and numbers k and d, does there exist a set F of k vertices of G such that every vertex of G can be can be reached from some member of F by a path of length at most d?

   (ii) **Yes** 3-SAT.

   (iii) **No** 2-SAT.

   It's polynomial

   (iv) **Yes** 4-SAT.

   (v) **Yes** Block sorting. Given a list, a *block move* consists of moving some contiguous sublist to another position. For example, the list AZCMXOPDYFQYS can be changed to AZCDYFMXOPQYS in one block move, by moving the sublist MXOP. The problem is, given a list ℓ and a number k, can ℓ be sorted in at most k block moves?

   (vi) **No** The minimum spanning tree problem for weighted graphs.

   It's polynomial, in fact can be solved by a simple greedy algorithm, such as Kruskal's.

   (vii) **Yes** The limited degree minimum spanning tree problem. The same as the previous problem, except that the degree of the spanning tree cannot exceed a given number.

   Amazing, isn't it! That simple restriction makes the problem much, much harder!

   (viii) **No** Determining whether there is a solution for a given configuration of **Rush Hour.**

   This problem is believed to be outside of $\mathcal{NP}$, although no proof is known.

   (ix) **Yes** Given a set of items of various sizes and a set of trucks of various capacities, can the items be loaded into the trucks?

   Proof that it's $\mathcal{NP}$: if there is a solution, that solution is a polynomial time certificate.

(x) **No** Given binary numerals for integers n and a, does there exist an integer d that divides n and is also between 2 and a?

That's really an open question of staggering importance. If it is polynomial, RSA coding is breakable in polynomial time. However, even if this problme were proved $\mathcal{P}$, it would still not prove $\mathcal{P} = \mathcal{NP}$, and other one-way functions could still exist.

3. Give a Chomsky Normal Form grammar equivalent to the grammar given below, where E is the start symbol.
$E \to E + E$
$E \to E - E$
$E \to -E$
$E \to x$
$E \to y$

Typically, when we convert to CNF, we have to increase the number of variables. Here we use $\{E, A, B, P, M\}$.
$E \to EA$
$E \to EB$
$A \to PE$
$P \to +$
$B \to ME$
$M \to -$
$E \to ME$
$E \to x$
$E \to y$

4. Consider the Chomsky Normal Form grammar G given below, where S is the start symbol.

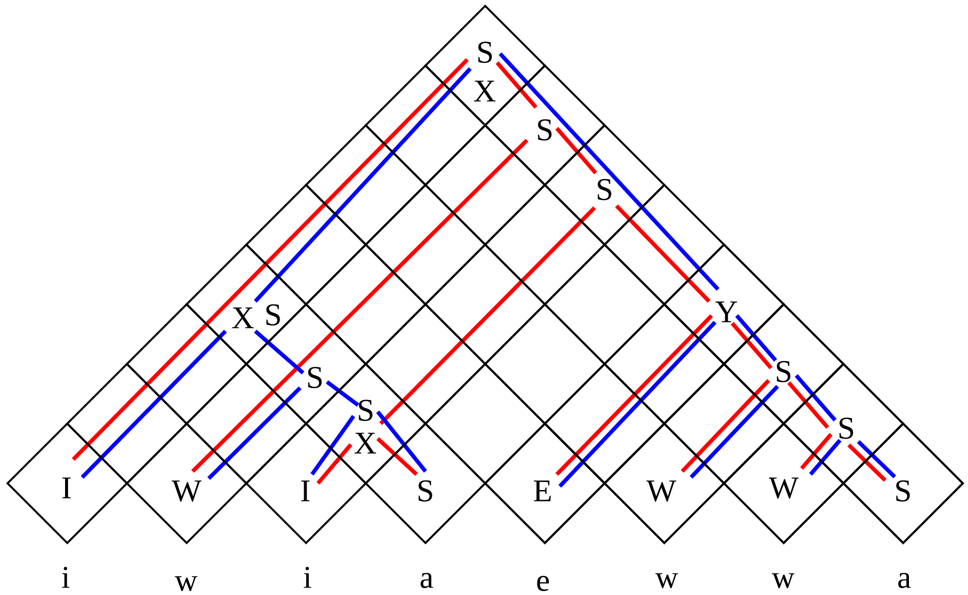
$S \to IS$
$S \to WS$
$S \to XY$
$X \to IS$
$Y \to ES$
$S \to a$
$I \to i$
$W \to w$
$E \to e$

(a) Show that G is ambiguous by giving two different **leftmost** derivations for the string $iiaea$.

$S \Rightarrow IS \Rightarrow iS \Rightarrow iXY \Rightarrow iISY \Rightarrow iiSY \Rightarrow iiaY \Rightarrow iiaES \Rightarrow iiaeS \Rightarrow iiaea$

$S \Rightarrow XY \Rightarrow ISY \Rightarrow iSY \Rightarrow iISY \Rightarrow iiSY \Rightarrow iiaY \Rightarrow iiaES \Rightarrow iiaeS \Rightarrow iiaea$

(b) Use the CYK algorithm to prove that $iwiaewwa \in L(G)$.



5. Give a polynomial time reduction of 3SAT to the independent set problem.

We give a polynomial time function $R : 3CNF \to \{\langle G\rangle\langle k\rangle\}$ such that for any $E \in 3CNF$ and $R(E) = \langle G\rangle\langle k\rangle$, G has an independent set of order k if and only if E is satisfiable.

Let $e = C_1 * C_2 * \cdots * C_k$, where each clause C_i is the disjunction of three terms, $t_{i,1} + t_{i,2} + t_{i,3}$, and each term $t_{i,j}$ is either a variable or the negation of a variable.

Let $G = (V, E)$ where $V = \{v_{i,j} : 1 \le i \le k, 1 \le j \le 3\}$ and E consists of the following edges:

- $\{v_{i,1}, v_{i,2}\}$, $\{v_{i,1}, v_{i,3}\}$, and $\{v_{i,2}, v_{i,3}\}$ for all $1 \le i \le k$. Call these clique edges.
- $\{v_{i,j}, v_{i',j'}\}$ for all $1 \le i < i' \le k$, $1 \le j, j' \le 3$ such that $t_{i,j} * t_{i',j'}$ is a contradiction. Call these long edges.

Suppose e is satisfiable. Pick a satisfying assignment of e. Each clause must have at least one term assigned true. For each $1 \le i \le k$, let $t_{i,j[i]}$ be such a term assigned true in clause C_i. Let $K = \{v_{i,j[i]}\} \subseteq V$. We need to prove that the members of K are independent. Pick $v_{i,j[i]}, v_{i',j[i']} \in K$. Suppose they are connected by a long edge. Then $t_{i,j[i]}$ are assigned contradictory values, which is impossible since the assignment is satisfying.

Conversely, suppose $\{G\}$ has an independent set I of k vertices. Since G is the union of k cliques, each clique must contain one member of I. Choose an assignment so that each member of I is true. This is possible because I cannot contain vertices which contradict each other. Since one term of each clause is true, the assignment is satisfying, hence e is satisfiable.

6. Give a polynomial time reduction of the subset sum problem to partition.

   An instance of the subset sum problem is a finite sequence of positive numbers, say $x_1, \ldots x_n$, together with a single number K, and the instance has a solution if there exists a subsequence of $\{x_i\}$ whose sum is K. An instance of the partition problem is a finite sequence of positive numbers, say $y_1, \ldots y_m$, and that instance has a solution if it has a subsequence whose total is half the sum of the sequence.

   We define a polynomial function R such that R maps each instance of the subset sum problem to an instance of the partition problem, such that either each has a solution or neither has a solution.

   Let $S = \sum_{i=1}^{n} x_i$. We can assume $S \leq K$, since otherwise it is trivial that the instance of the subset sum problem has no solution. Choose an instance of the partition problem $y_1, \ldots y_m$ as follows:

   (a) $m = n + 2$

   (b) $y_i = x_i$ if $i \leq n$

   (c) $y_{n+1} = K + 1$

   (d) $y_{n+2} = S - K + 1$

   Suppose that there is a solution to the instance of the subset sum problem. We can rearrange the terms of the sequence such that the sum of the first k terms equals K. The sum of all $\{y_j\}$ is $\sum_{i=1}^{n} x_i + K + 1 + S - K + 1 = 2S + 2$ Then $y_1 + \cdots + y_k + y_{n+2} = S + 1$ which is half the total, giving a solution to the instance of the partition problem.

   Conversely, suppose there is a solution to the instance of the partition problem. Then there must be two subsequences, each of whose total is $S + 1$. It is impossible for one of them to contain both y_{n+1} and y_{n+2}, since the sum of those two terms is greather than $S + 1$. Thus one of the subsequences contains y_{n+2} together with a subsequence of other terms. We can rearrange the sequence so that those other terms are $y_1, \ldots y_k$. Then $\sum_{i=1}^{n} x_i + S - K + 1 = S + 1$, hence $\sum_{i=1}^{n} x_i = K$, which gives us a solution to the subset sum instance.

7. State the Church-Turing thesis. Why is it important?

   The Church-Turing thesis is that any computation performed by any machine can be performed by some Turing machine. This is important because To prove that no machine can perform a given computation, it is sufficient to prove that no Turing machine can perform that computation, which is made easier by the fact that Turing machines are very simple.