6.045J/18.400J: Automata, Computability and Complexity Prof. Nancy Lynch

Practice Quiz 3

April 25, 2007 *Elena Grigorescu*

Please write your name in the upper corner of each page.

Problem	Points	Grade
1	20	
2	10	
3	10	
4	20	
5	20	
6	20	
Total	100	

Problem 1: **True, False, or Unknown (20 points).** In each case, say whether the given statement is known to be TRUE, known to be FALSE, or currently not known either way. Full credit will be given for correct answers. If you include justification for your answers, you may obtain partial credit for incorrect answers.

1. True, False, or Unknown: 8^n is $O(2^n)$.

2. True, False, or Unknown: There exists a language that is not decidable in time n^8 but is decidable in time 8^n (where n is the length of the input).

3. True, False, or Unknown: $\mathsf{SAT} \in \mathsf{coNP}$.

4. True, False, or Unknown: The $\mathsf{2COLOR}$ problem is NP-complete.

5. True, False, or Unknown: The 4-CLIQUE problem, defined as the set of undirected graphs containing a 4-CLIQUE, is in P.

6. True, False, or Unknown: 2SAT (the set of CNF Boolean formulas with at most two literals per clause) $\leq_p$ PALINDROMES.

7. True, False, or Unknown: There is a nontrivial language A such that $P^A \neq NP^A$.

8. Suppose there is a 2^n time-bounded reduction from language L_1 to language L_2, and L_2 is in TIME(2^n). Then, L_1 is in TIME($2^{O(n)}$).

Problem 2: **(10 points)** Suppose that A, B, and C are nontrivial languages over $\Sigma = \{0, 1\}$. Assume that:

1. $A \in NP$,
2. $C \in P$,
3. $A \leq_p B \leq_p A \cap C$, and
4. $A \cap C$ is NP-hard.

Prove that B is NP-complete. You may invoke theorems proved in class and in the book, but if you do this, cite them explicitly.

Problem 3: **(10 points)** The proof that SAT is NP-complete appears in Section 7.4 of Sipser's book.

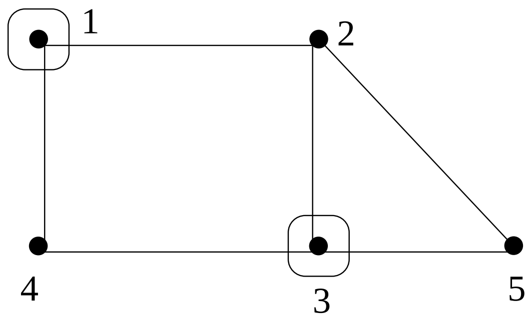
1. Suppose we try to use exactly the same construction to show that every problem A in the class NEXPTIME, representing nondeterministic exponential time (time $O(2^{n^k})$ for some k), is polynomial time reducible to SAT. Exactly where would the proof fail?

2. The overall formula ϕ actually depends on the input x for the given nondeterministic Turing machine. Explain in words where the input appears in the formula.

Problem 4: **(20 points)** The following is a variation on the Vertex Cover problem, called Dominating Set:

$$\mathsf{DS} = \{\langle G, k\rangle : G \text{ is an undirected graph } (V, E), \text{ and there exists a subset } C \subseteq V, \text{ with } |V| = k,$$
$$\text{and such that every vertex in } V \setminus C \text{ has an edge in } E \text{ to some vertex in } C\ \}.$$

For example, in the following graph G, $C = \{1, 3\}$ is a dominating set of size 2:



1. Prove that DS is in NP, by describing a polynomial-time nondeterministic Turing machine that decides DS.

2. Prove that DS is NP-hard, using a polynomial time reduction from the Vertex Cover problem.

   (**Hint:** From the given graph $G = (V, E)$, construct a graph $G' = (V', E')$ such that $V' = V \cup E$.)

Problem 5: **(20 points)** Suppose we are given a finite set S of elements, together with a finite collection of subsets $S_1, S_2, S_3, ...S_k$ of S. Also, for each set S_i, we are given two numbers low_i and high_i. The problem is to figure out whether it is possible to select a subset $T \subseteq S$ such that, for every i, the number of elements of S_i in T is in the interval [low_i,high_i]. i.e, for all i, $\mathsf{low}_i \leq |T \cap S_i| \leq \mathsf{high}_i$. For example, consider

$$S = \{a, b, c, d, e\}$$
$$S_1 = \{a, b, e\}, \mathsf{low}_1 = 1, \mathsf{high}_1 = 1$$
$$S_2 = \{a, c, d\}, \mathsf{low}_2 = 1, \mathsf{high}_2 = 2$$
$$S_3 = \{b, c, d, e\}, \mathsf{low}_3 = \mathsf{high}_3 = 3$$

This instance is in the language. To see this, set $T = \{b, c, d\}$. T intersects S_1 in one element b, S_2 in two elements c and d, and S_3 in three elements b, c and d. However, the same example, with $\mathsf{low}_2 = \mathsf{high}_2 = 1$, has no choice for T that works.

1. Define this problem formally as a language called SUBSETS.

2. Prove that SUBSETS is in NP, using the certificate/verifier formulation.

3. Prove that SUBSETS is NP-hard, using a direct reduction from 3SAT.

Problem 6: **(20 points)** Consider the language 3DM (Three-Dimensional Matching) defined as:

$$\mathsf{3DM} = \{\langle A, B, C, M\rangle \mid A, B, \text{ and } C \text{ are disjoint sets of the same size, } M \subseteq A \times B \times C, \text{ and there exists } M' \subseteq M \text{ such that every element of } A, B, \text{ and } C \text{ appears in exactly one triple in } M'\}.$$

Suppose that we are given a polynomial-time decider algorithm D that decides membership in 3DM.

1. Describe a polynomial time algorithm that, given the representation $\langle A, B, C, M\rangle$, finds a set $M' \subseteq M$ such that every element of A, B, and C appears in exactly one triple in M', if one exists, and otherwise outputs "none exists". Your algorithm may use the decider D as a "subroutine".

2. Show that your algorithm in part 1 is actually polynomial time-bounded. Specifically, assuming that the decider D runs in time $p(n)$ for polynomial p (of degree at least 2), give a specific polynomial time upper bound for your algorithm (including the time for the "subroutine" D).