CpSc 421 Homework 11 **Due: Dec. 1, 4pm**
No late homework accepted.

Attempt any **three** of the **six** problems below. The homework is graded on a scale of 100 points, even though you can attempt fewer or more points than that. Your recorded grade will be the total score on the problems that you attempt.

1. (**30 points**, Sipser problem 7.20) Let G represent an undirected graph, and let

$$\begin{array}{rcl} \mathit{ShortPath} & = & \{(G,a,b,k) \mid G \text{ contains a simple path of length at most } k \text{ from } a \text{ to } b.\} \\ \mathit{LongPath} & = & \{(G,a,b,k) \mid G \text{ contains a simple path of length at least } k \text{ from } a \text{ to } b.\} \end{array}$$

   (a) (**15 points**) Show that $\mathit{ShortPath} \in P$.

   (b) (**15 points**) Show that $\mathit{LongPath}$ is NP-complete. You may assume the NP-completeness of any problem shown to be NP-complete in class or in assigned reading of Sipser (e.g. you may assume the NP completeness of $\mathit{UHAMPATH}$, Sipser theorem 7.55).

2. (**35 points**, Sipser problem 5.24) Let ϕ be a 3cnf formula. An $\neq$-*assignment* to the variables of ϕ is one where each clause contains two literals with unequal truth values. In other words, an $\neq$-assignment satisfies ϕ without assigning three true literals in any clause.

   (a) (**10 points**) Show that the negation of any $\neq$-assignment to ϕ is also a $\neq$-assignment.

   (b) (**20 points**) Let $\neq\!\mathit{SAT}$ be the collection of 3cnf-formulas that have an $\neq$-assignment. Show that we obtain a polynomial time reduction from $\mathit{3SAT}$ to $\neq\!\mathit{SAT}$ by replacing each clause, c_i,

$$(y_1 \vee y_2 \vee y_3)$$

   with

$$(y_1 \vee y_2 \vee z_i) \quad \mathit{and} \quad (\overline{z_i} \vee y_3 \vee b),$$

   where z_i is a new variable for each clause and b is a single additional new variable.

   (c) (**5 points**) Conclude that $\neq\!\mathit{SAT}$ is NP-complete. In other words, part (b) shows either that $\neq\!\mathit{SAT}$ is in NP or that it is NP hard. Identify which it does, and take care of the other piece needed to make the requested conclusion.

3. (**35 points**, Sipser problem 7.26) You are given a box and a collection of cards as indicated in Figure 1. Because of the pegs in the box and the notches in the cards, each care will fit in the box either of two ways. Each card consists of two columns of holes, some of which may not be punched out. The puzzle is solved by placing all the cards in the box so as to completely cover the bottom of the box (i.e. every hole position is blocked by at least one card that has no hole there.). Let

$$\mathit{PUZZLE} \;=\; \{\langle c_1, \ldots, c_k \rangle \mid \text{each } c_i \text{ represents a card and this collection of cards has a solution}\}.$$

   (a) (**30 points**) Show that PUZZLE is NP-complete.

   (b) (**5 points**) Let

$$\begin{array}{rcl} \mathit{UNPUZZLE} & = & \{\langle c_1, \ldots, c_k \rangle \mid \text{each } c_i \text{ represents a card and there is a way to place the} \\ & & \qquad \text{cards that is } \textit{not} \text{ a solution to the puzzle problem } \}. \end{array}$$

   Show that the $\mathit{UNPUZZLE} \in P$.

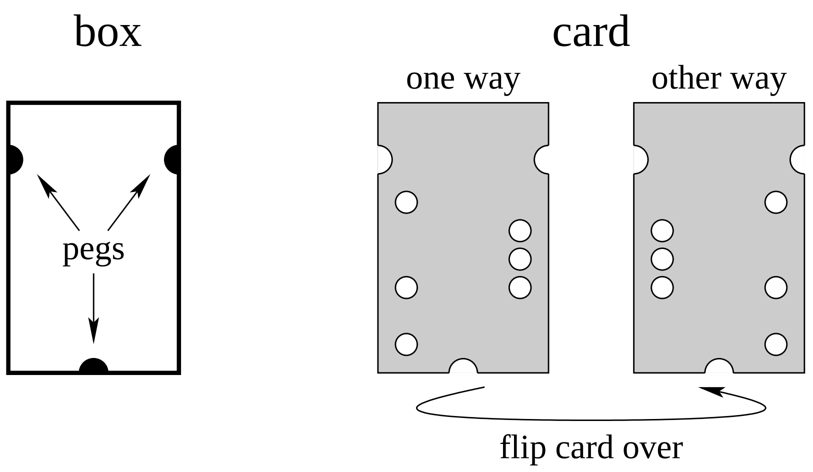

Figure 1: Card for question 3

4. (**45 points**, Sipser problem 7.35) Let $Q = \{q_0, q_1, \dots q_m\}$ be a finite set of states and $P = \{(s_1, p_1), (s_2, p_2), \dots, (s_k, p_k)\}$ be a finite set of pairs where the s_i are distinct strings over $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, and the r_i are (not necessarily distinct) members of Q. Let $\langle Q, P\rangle$ be a string describing Q and P. The language ∃DFA language is

   $$\exists\text{DFA} \quad = \quad \{\langle Q, P\rangle \mid \text{there exists a DFA } M = (Q, \Sigma, \delta, q_0, F) \text{ such that } \forall (s, p) \in P.\ \delta(q_0, s) = p\}\delta$$

   Show that the language ∃DFA is NP-complete.

5. (**40 points**, Sipser problem 7.35) Show that if $P = NP$, then an polynomial time algorithm exists that produces a satisfying assignment when given a satisfiable Boolean formula.

   Note: The algorithm you are asked to provide computes a function whereas NP contains languages, not functions. The $P = NP$ assumption implies that SAT is in P; so, testing satisfiability is solvable in polynomial time. However, this assumption *does not* say how this test is done, and the test may not reveal satisfying assignments. You must show that you can find them anyway.

   Hint: Use the satisfiability solver repeatedly to find the assignment, bit-by-bit.

6. (**40 points**, Sipser problem 7.42) A *2-cnf* formula is an AND of clauses, where each clause is an OR of at most two literals. Let $2SAT = \{\langle\phi\rangle \mid \phi \text{ is a satisfiable 2cnf-formula}\}$. Show that $2SAT \in P$.