Family name: *Given names:* *Student ID:*

1	
2	
3	
T	

CSC 363 — Test #2 — 2010-03-17

No books, notes, or other information storage systems are allowed.
You may use results proved in the book (except in exercises or problems) without proving them here.

1) [30 marks] Recall that a clique in an undirected graph is a set of nodes in which every pair of nodes is connected by an edge. The textbook defined the language CLIQUE as follows:

$$\text{CLIQUE} \;=\; \{\, \langle G, k \rangle \mid G \text{ is an undirected graph that contains a clique with } k \text{ nodes} \,\}$$

The textbook proves that CLIQUE is NP-complete. Define the language TWO-CLIQUES as:

$$\text{TWO-CLIQUES} \;=\; \{\, \langle G, k \rangle \mid G \text{ is an undirected graph that contains two disjoint cliques of size } k \,\}$$

Prove that TWO-CLIQUES is NP-complete. Remember: You need to show *two* things to show that a language is NP-complete.

2) [45 marks total] Part of the proof in the textbook that SAT is NP-complete shows that for any language, A, in NP, which is decided by a nondeterministic Turing Machine, N, that runs in polynomial time, there is a function that maps a string w to a string $\langle \phi \rangle$ that is an encoding of a Boolean formula, ϕ, that is satisfiable iff N accepts w.

The proof shows that there is an algorithm to do this reduction in polynomial time, for some fixed nondeterministic Turing Machine, N, which runs in some polynomial time bound — say $n^k + 2$, for some k, where n is the length of the input. The algorithm takes the string w as input and outputs $\langle \phi \rangle$. The formula ϕ that it creates has variables that describe the "tableau" for a computation of N on input w that halts within $n^k + 2$ steps (we'll let this tableau be $n^k + 3$ by $n^k + 5$ in size). The rows of the tableau are successive configurations of N, bounded by "#" symbols. The variable $x_{i,j,s}$ is 1 iff cell (i, j) of the tableau contains symbol s, where $s \in Q \cup \Gamma \cup \{\#\}$.

Recall that the formula ϕ has the form

$$\phi \;=\; \phi_{\text{cell}} \;\wedge\; \phi_{\text{start}} \;\wedge\; \phi_{\text{move}} \;\wedge\; \phi_{\text{accept}}$$

where ϕ_{cell} enforces that the variables describe a tableau with exactly one symbol in each cell, ϕ_{start} enforces that the first configuration is the correct start configuration for input w, ϕ_{move} enforces that each configure is followed by a possible successor configuration (same as the previous one if the machine has halted), and ϕ_{accept} enforces that the tableau contains an accepting configuration.

Suppose that the input alphabet of machine N is $\Sigma = \{\, 0,\ 1 \,\}$, the tape alphabet is $\Gamma = \{\, 0,\ 1,\ \text{blank} \,\}$, the state space is $Q = \{\, q_0,\ q_1,\ q_{\text{accept}},\ q_{\text{reject}} \,\}$, the start state is q_0, and the transition function, $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$, is as follows:

$$\delta(q_0, 0) \;=\; \{\, (q_1, 1, L),\ (q_1, 0, R) \,\}, \quad \delta(q_0, 1) \;=\; \{\, (q_1, 1, L) \,\}, \quad \delta(q_0, \text{blank}) \;=\; \{\, (q_{\text{reject}}, \text{blank}, L) \,\}$$

$$\delta(q_1, 1) \;=\; \{\, (q_1, 1, R) \,\}, \quad \delta(q_1, 0) \;=\; \{\, (q_{\text{reject}}, 0, R) \,\}, \quad \delta(q_1, \text{blank}) \;=\; \{\, (q_{\text{accept}}, \text{blank}, L) \,\}$$

For all the questions below, suppose that the input is $w \;=\; 011$, so that $n = 3$, and that $k = 1$, so the tableau has 6 rows and 8 columns.

a) [12 marks] Fill in the two tableaus below to represent two different accepting computations on this input.

b) [5 marks] How many variables are there in the formula ϕ? Explain.

b) [9 marks] Write down the ϕ_{start} part of ϕ for this input.

c) [9 marks] The ϕ_{accept} part of ϕ is a disjunction (or) of literals. Write down three of these literals, and say (and explain) how many literals are in this disjunction.

d) [10 marks, +1 for each correct, −1 for each wrong, minimum 0] The ϕ_{move} part of ϕ ensures that every 2×3 "window" of the tableau is legal for the machine N. For each of the following windows, circle "Yes" or "No" to indicate whether it is legal or not (no explanation is required):

#	0	1
#	0	1

Legal? Yes No

1	1	q_1
1	1	1

Legal? Yes No

q_0	1	1
q_0	1	1

Legal? Yes No

0	0	1
0	1	1

Legal? Yes No

q_1	0	1
1	0	1

Legal? Yes No

#	q_0	1
#	q_0	0

Legal? Yes No

q_0	0	1
1	q_0	1

Legal? Yes No

q_1	1	1
1	q_1	1

Legal? Yes No

1	q_1	blank
q_{accept}	1	blank

Legal? Yes No

#	q_0	blank
#	q_0	blank

Legal? Yes No

3) [25 marks] The class coNP is defined to contain all languages whose complements are in NP — in other words, $L \in$ coNP iff $\overline{L} \in$ NP. A language L is defined to be coNP-complete if L is in coNP and any other language in coNP is polynomial time reducible to L — in other words, L is coNP-complete iff $L \in$ coNP and for all $L' \in$ coNP, $L' \leq_P L$.

Prove that $\overline{\text{SAT}}$ is coNP-complete. You may use any parts of the proof that SAT is NP-complete that are useful for proving this.