CSC 363H1 Y **Assignment # 3** **Summer 2006**

Due: By 7:00pm on Thursday, July 20. **Worth**: 10%

1. [15 marks]

   Prove that the following languages are in the class P.

   (a) $\{\langle C \rangle : C$ is a winning configuration in an 8x8 chess game$\}$.
   A *configuration* in the 8x8 chess game consists of a chessboard, the positions of the pieces, and which player is to act next. Some of the pieces may be captured by the opponent. A configuration is said to be *winning* if the white player can play in such a way that, no matter how the black player chooses his moves, the white player will eventually win the game. That is, assuming that both players play optimally, white player will eventually win. Note that if the game ends in a tie, the white player has not won.

   (b) $\{\langle G, s_1, s_2, t \rangle : G$ is an undirected graph that contains two simple paths P_1, from s_1 to t, and P_2, from s_2 to t, such that P_1 and P_2 share at least two different vertices$\}$.
   Note that the two different vertices that P_1 and P_2 share may include the origin and the end vertices.

2. [15 marks]

   Suppose L is a language over the alphabet Σ^n, where $|\Sigma| > 1$ and $n > 1$. The *projection* of a string $w \in (\Sigma^n)^*$ is the string $\pi(w) \in (\Sigma^{n-1})^*$ that you obtain by forgetting the last entry of each character in w. For example, if $w = (0,1,1)(1,1,0)(0,0,1)$, then $\pi(w) = (0,1)(1,1)(0,0)$. The *projection* of a language L is the language $\pi(L)$ over the alphabet Σ^{n-1} consisting of strings $\pi(w)$ where $w \in L$. A complexity class $\mathcal{C}$ is said to be *closed under projection* if whenever a language L over the alphabet Σ^n is a member of $\mathcal{C}$, where $|\Sigma| > 1$ and $n > 1$, its projection also belongs to $\mathcal{C}$.

   (a) Show that NP is closed under projection.

   (b) Show that P is closed under projection iff $P = NP$.

3. [15 marks]

   Prove that, if $P = NP$, then there exists a polynomial time deterministic algorithm that produces a satisfying assignment whenever a satisfiable boolean formula is given. Whenever the input formula is unsatisfiable, the algorithm simply outputs "no".

   Note: If $P = NP$, there exists a polynomial time Turing machine that *decides* SAT. Even so, it is not immediate that the satisfying assignment to a satisfiable boolean formula can be produced in polynomial time.
   Hint: Use the satisfiability tester to find the satisfying assignment bit-by-bit.

4. [15 marks]

   Define a variant SAT' of SAT as follows:

   $$SAT' = \{\langle F \rangle : F \text{ is a Boolean formula with at least 4 satisfying truth assignments}\}.$$

   Show that SAT' is NP-complete.