CSI 4105 Midterm 2001 (Format was compressed to fit 2 pages)

1. Turing Machines (9 points)

Consider a Turing machine $M$ with input alphabet $\Sigma = \{0, 1\}$ and machine alphabet $\Gamma = \{0, 1, \cup\}$, and transition function depicted in the following diagram:

Part A (6 points) Give $L$, the language accepted by $M$.

2. Short answers (21 points) For each of the statements below choose TRUE or FALSE and briefly justify your answer. Note: You may use any result shown in class or in homework without proving it.

- $\text{NP} \cap \text{co-NP} = \emptyset$
- If $\text{HAM-CYCLE} \in \text{P}$ then $\text{VERTEX-COVER} \in \text{P}$
- If $\text{SAT} \notin \text{P}$ then $\text{CLIQUE} \notin \text{P}$
- $\text{SHORTEST-PATH} \in \text{P}$
- If $\text{SHORTEST-PATH} \in \text{NPC}$ then $\text{P} = \text{NP}$.
- It is possible that there exists a RAM program that decides a language $L_1$ in polynomial time, but every Turing machine that decides $L_1$ runs in exponential time.
- If there exists a polynomial-time algorithm that accepts a language $L_2$, then there exists a polynomial-time algorithm that decides $L_2$.

3. Polynomial-time reducibility (10 points)

Prove that $\leq_P$ relation is a transitive relation on languages. That is, prove that:
if $L_1 \leq_P L_2$ and $L_2 \leq_P L_3$ then $L_1 \leq_P L_3$.

Hint: Use the definition of $\leq_P$.

4. NP-completeness reductions (20 points)

In this question, you are asked to apply some reductions discussed in class to the examples given. For language $L_1$ and $L_2$ shown in each part below, we have seen a reduction algorithm that shows that $L_1 \leq_P L_2$; you should apply the algorithms discussed in class.

Part A (5 points) $\text{CIRCUIT-SAT} \leq_P \text{SAT}$

Consider the following instance for the circuit satisfiability problem (CIRCUIT-SAT):
circuit $C_1$:

- Give the corresponding instance for the formula satisfiability problem (SAT), in accordance with the reduction algorithm for $\text{CIRCUIT-SAT} \leq_P \text{SAT}$.
• Does \(< C_1 > \in \text{Circuit-Sat}\)? Justify.

Part B (5 points) \(\text{HAM-CYCLE} \leq_p \text{TSP}\)
Consider the following instance for the hamiltonian cycle problem (\(\text{HAM-CYCLE}\)):

\(G_1:\)

- Give the corresponding instance for the traveling salesman problem (\(\text{TSP}\)), in accordance with the reduction algorithm for \(\text{HAM-CYCLE} \leq_p \text{TSP}\).
- Does \(< G_1 > \in \text{HAM-CYCLE}\)? Justify.

Part C. (10 points) \(\text{VERTEX-COVER} \leq_p \text{HAM-CYCLE}\)
Consider the following instance for the vertex-cover problem (\(\text{VERTEX-COVER}\)):

\(k = 2, \quad G_2:\)

- Give the corresponding instance for the hamiltonian cycle problem (\(\text{HAM-CYCLE}\)), in accordance with the reduction algorithm for \(\text{VERTEX-COVER} \leq_p \text{HAM-CYCLE}\).
- Outline the hamiltonian cycle (on the graph you have drawn above) that corresponds to the vertex cover \(C = \{v_1, v_2\}\) of \(G_2\).

5. (40 points) **Longest simple cycle is in NPC**

The **longest-simple-cycle** problem is the problem of determining a simple cycle (no repeated vertices) of maximum length in a graph. For example:

<table>
<thead>
<tr>
<th>(G_1:)</th>
<th>(G_2:)</th>
</tr>
</thead>
<tbody>
<tr>
<td>longest simple cycle in (G_1): (1,2,3,6)</td>
<td>longest simple cycle in (G_2): (1,2,3,5,4)</td>
</tr>
<tr>
<td>length of the longest simple cycle: 4</td>
<td>length of the longest simple cycle: 5</td>
</tr>
</tbody>
</table>

Part A. (2) State the decision problem associated with the longest-simple-cycle problem:
Part B. (3) Give the language corresponding to the decision problem given in part A: \(\text{LSC} = \)
Part C. (35) Prove that \(\text{LSC}\) is NP-complete, by proving that:

• (10) \(\text{LSC} \in \text{NP}\).
• (25) \(\text{LSC}\) is NP-hard. (Hint: reduce from \(\text{HAM-CYCLE}\))