University of Ottawa

CSI 4105 – MIDTERM SOLUTION

Instructor: Lucia Moura

Feb 6, 2010
10:00 am
Duration: 1:50 hs

Closed book

Last name: ________________________

First name: ________________________

Student number: ________________________

There are 4 questions and 100 marks total.

This exam paper should have 8 pages, including this cover page.

1 – Short answers	/ 25
2 – Search versus Decision problem	/ 25
3 – NP-completeness reductions	/ 25
4 – Clique is NP-complete	/ 25
Total	/ 100

1 Short answers — 25 points

The questions below are of the type "true or false"; briefly justify your answer.
Note: You may use any result shown in class or in homework without proving it. You may use the fact that you know particular problems are polynomial-time solvable or NP-complete.

- For all problems $X \in \mathcal{NP}$, $X \leq_P$ 3D-MATCHING. **[TRUE]**
  Justify:
  3D-MATCHING is NP-complete, and the above statement says that 3D-MATCHING is NP-hard.

- If VERTEX-COVER $\in \mathcal{P}$ then SAT $\in \mathcal{P}$. **[TRUE]**
  Justify:
  Since VERTEX-COVER is NP-complete then any problem in $\mathcal{NP}$ polytime reduces to it, in particular SAT. From SAT $\leq_P$ VERTEX-COVER and VERTEX-COVER $\in \mathcal{P}$ it follows SAT $\in \mathcal{P}$.

- If $\mathcal{P} = \mathcal{NP}$ then SHORTEST-PATH is NP-complete. **[TRUE]**

  Any problem in $\mathcal{P}$ is polynomial-time reducible to any other problem in $\mathcal{P}$. If $\mathcal{P} = \mathcal{NP}$ then the same would hold for $\mathcal{NP}$. Since SHORTEST-PATH $\in \mathcal{NP}$, it follows that for all $X \in \mathcal{NP}$, $X \leq_P$ SHORTEST-PATH. Therefore, SHORTEST-PATH is NP-complete.

- It is possible that INDEPENDENT-SET $\in \mathcal{P}$ and HAM-CYCLE $\notin \mathcal{P}$ **[FALSE]**
  Justify:
  Similar to the second question. Since INDEPENDENT-SET is NP-complete then any problem in $\mathcal{NP}$ polytime reduces to it, in particular HAM-CYCLE. From HAM-CYCLE $\leq_P$ INDEPENDENT-SET and INDEPENDENT-SET $\in \mathcal{P}$ it follows HAM-CYCLE $\in \mathcal{P}$. Which contradicts HAM-CYCLE $\notin \mathcal{P}$.

- Let X_1 and X_2 be decision problems in $\mathcal{NP}$, and assume $\mathcal{P} \neq \mathcal{NP}$.
  If $X_1 \leq_P X_2$ and $X_2 \leq_P X_1$, then both X_1 and X_2 are NP-complete. **[FALSE]**

  Justify:
  **We give a counterexample. Take X_1 and X_2 in $\mathcal{P}$.
  This implies $X_1 \leq_P X_2$ and $X_2 \leq_P X_1$. In addition, since $\mathcal{P} \neq \mathcal{NP}$ by assumption, then X_1 and X_2 are not NP-complete.**

... continued

2 Search versus Decision problem — 25 points

Recall that 3-Sat is the following decision problem:
"Given a formula ϕ which is a conjunction of k clauses over a set of variables $\{x_1, x_2, \ldots, x_n\}$, does there exist a satisfying truth assignment for ϕ?"
Consider the analogous problem 3-SatSearch that **computes a satisfying assignment** for ϕ, if one exists, or outputs "ϕ is unsatisfiable", otherwise.

Show that if 3-Sat can be solved in polynomial time, then 3-SatSearch can be solved in polynomial time.
Hint 1: Do this by providing an algorithm that solves 3-SatSearch by doing calls to the polynomial-time algorithm that decides 3-Sat.
Justify that your algorithm runs in polynomial time. (Note, your formulas may grow in size, and you must be careful and justify that the calls for 3-Sat have time that remain polynomial on the size of ϕ, even when using for your transformed larger formulas).

Hint 2 : One can force the value of a variable x_i in formula ϕ, by transforming ϕ into $\phi \wedge \phi'_{i,v}$, where $\phi'_{i,v}$ is given below, and uses two extra variables y and z:

- Forcing $x_i = 1$: $\phi'_{i,1} = (x_i \vee y \vee z) \wedge (x_i \vee y \vee \overline{z}) \wedge (x_i \vee \overline{y} \vee z) \wedge (x_i \vee \overline{y} \vee \overline{z})$
- Forcing $x_i = 0$: $\phi'_{i,0} = (\overline{x_i} \vee y \vee z) \wedge (\overline{x_i} \vee y \vee \overline{z}) \wedge (\overline{x_i} \vee \overline{y} \vee z) \wedge (\overline{x_i} \vee \overline{y} \vee \overline{z})$

```
Algorithm 3-SatSearchALG(φ)
    Input:    a fomula φ in conjunctive normal form with tree literals per clause,
              having n-variables and k-clauses.
    Output:   a truth assignment to x1, x2 ..., xn that satisfies φ,
              or ''φ is not satisfiable''

    If (3-SAT-ALG(φ)='no' ) then return ''φ is not satisfiable''
    else
      φ' ← φ
      for i = 1 to n do
          if (3-SAT-ALG(φ' ∧ φ'_{i,1})='yes') then
             φ' ← φ' ∧ φ'_{i,1};  x_i ← 1;
          else
             φ' ← φ' ∧ φ'_{i,0};  x_i ← 0;
    return (x1, x2, ..., xn)
```

(continues in the next page...)

Justification of polynomial running time: Assume 3-SAT-ALG is a polynomial-time algorithm that solves 3-SAT. The above algorithm repeats n times the loop. The loop consists of a constant number of operations and a call to 3-SAT-ALG which runs in time polynomial with its input. Over all the iterations, the input formula grows from ϕ (having n variables and k clauses) to a formula with $n+2$ variables and $k+4n$ clauses. So each formula for which 3-SAT-ALG is called has size polynomial on the size of ϕ, and this algorithm runs in time polynomial on the size of this formula. So 3-SAT-ALG runs in time polynomial on the size of ϕ. Therefore, the above algorithm runs in polynomial time.

... continued

3 NP-completeness reductions — 25 points

In this question, you are asked to apply some reductions discussed in class to the examples given. For problems X_2 and X_2 shown in each part below, we have studied a reduction algorithm that shows that $X_1 \leq_P X_2$; you should apply the algorithms discussed in class.

Part A — 10 points 3-SAT $\leq_P$ INDEPENDENTSET

Consider the following instance for 3-SAT: $\phi = (\overline{x_1} \vee x_3 \vee \overline{x_4}) \wedge (x_1 \vee x_2 \vee x_4)$.

- Give the corresponding instance for the independent set problem INDEPENDENT SET, according to the reduction algorithm for 3-SAT $\leq_P$ INDEPENDENTSET. You need to provide (G, k), where G is a graph and k is a target size for the independent set.

  $k = 2$

  Will avoid having to draw...
  graph description: 6 vertices, 8 edges
  triangle with vertices 1, 2, 3 with labels $\overline{x}_1$, x_3, $\overline{x_4}$
  triangle with vertices 4, 5, 6 vertices with labels x_1, x_2, x_4.
  extra edges connecting $\{x_1, \overline{x_1}\}$ and $\{x_4, \overline{x_4}\}$

  One possible independent set coming from the truth assignment below is vertices 1 and 6, corresponding to $\overline{x_1}$ in the irst triangle and x_4 in the second triangle.

- Give a satisfying assignment for ϕ and show how it translate to an independent set of the right size, by marking the independent set in the picture above.

  $x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1$; **independent set of size 2 above.**

... continued

Part B — 15 points 3-SAT $\leq_P$ DIRECTEDHAMCYCLE

Use the same instance of 3-SAT as shown in the previous part.

- Give the corresponding instance for the directed Hamiltonian cycle problem, DIRECTED-HAMCYCLE, according to the reduction algorithm for 3-SAT $\leq_P$ DIRECTEDHAMCYCLE. You need to provide a graph.

  This is a bit tedious to draw in the computer...

  In summary, we need 4 paths with 9 vertices each.
  In each path, vertices 3,4 are reserved for clause 1; and 6,7 is reserved for clause 2.

  Follow construction in the textbook to connect 2 clause nodes to these paths, and add edges that connect the paths and add vertices s and t accordingly.

  My hamiltonian cycle, corresponding to the truth assignment in the previous page, in summary follows the order: s, P_1 from right to left picking up clause C_1, P_2 from left to right, P_3 from left to write, P_4 from left to right picking up clause 2, t, s.

- Show a hamiltonian cycle in the graph above, corresponding to a satisfying assignment for ϕ. You may use the same satisfying assignment as in the previous question.

... continued

4 Clique is NP-complete — 25 points

Recall that an **independent set** in a graph $G = (V, E)$ is a subset $I \subseteq V$ such that for all $x, y \in I$, $\{x, y\} \notin E$.

A **clique** in a graph $G' = (V', E')$ is a subset $C \subseteq V'$ such that for all $x, y \in C$ with $x \neq y$, $\{x, y\} \in E'$.

Consider the following decision problems:

IndSet: Given (G, k), does G have an independent set of size at least k ?

Clique: Given (G, k), does G have a clique of size at least k ?

Recall the definition of the complement of a graph below, which in plain words say that you keep the same vertex set, but switch edges with non-edges.
The **complement** of a graph $G = (V, E)$ is a graph $\overline{G} = (V, E')$ such that for every $x, y \in V$, $x \neq y$, we have $\{x, y\} \in E'$ if and only if $\{x, y\} \notin E$.

Part A — 10 points Let $G = (V, E)$ be a graph, let $\overline{G}$ be its complement graph and let $S \subseteq V$. Prove that S is an independent set of G if and only if S is a clique of $\overline{G}$.

Example:

$\{1, 2, 3\}$ is an independent set of G	$\{1, 2, 3\}$ is a clique of $\overline{G}$

Proof:

($\Rightarrow$) Let S be an independent set of G. Then, for all $x, y \in S$ we have $\{x, y\} \notin E$. Then, for all $x, y \in S$ we have $\{x, y\} \in E'$. Therefore, S is a clique in $\overline{G}$.

($\Leftarrow$) Let S be a clique of $\overline{G}$. Then, for all $x, y \in S$ we have $\{x, y\} \in E'$. Then, for all $x, y \in S$ we have $\{x, y\} \notin E$. Therefore, S is an independent set in G.

... continued

Part B — 15 points Prove that CLIQUE is NP-complete, by proving that:

- (5 points) CLIQUE $\in \mathcal{NP}$.

  We can build an efficient certifier for CLIQUE**, whose inputs are** (G, k) **and a set of vertices** S**. The certifier checks that** $|S| \geq k$ **and that for all** $x, y \in S, x \neq y$**, we have** $\{x, y\} \in E(G)$**. The certifier contains at most** n **integers smaller than or equal to** n**, so its size is polynomial on the size of** (G, k)**. This algorithm runs in time** $O(n^2)$**.**

- (10 points) CLIQUE is NP-hard.
  Hint: Reduce from INDEPENDENTSET. You may use the result in Part A, even if you did not prove it.

  - **The reduction algorithm description for** INDEPENDENT-SET $\leq_P$ **Clique:**
    The reduction algorithm takes as input (G, k)**, calculates** $\overline{G}$ **and returns** $(\overline{G}, k)$**. To calculate** $\overline{G}$**, we go over every pair of distinct vertices** x, y**: if** x **and** y **are not connected by an edge in** G**, we make them connected by an edge in** $\overline{G}$**.**
  - **The polynomial time of the reduction algorithm:**
    This runs in polynomial time, since we can calculate the complement of G **in** $O(n^2)$**.**
  - **The fact that yes-instances map to yes-instances and no-instances map to no-instances:**
    If there exists S **an independent set of** G **of size** k **then by part A,** S **is a clique of** $\overline{G}$ **of size** k**.**

    If there exists no independent set of size k **in** G**, then for any set of vertices** S **with** $|S| = k$**, there exist a pair of vertices** x, y **with** $x \neq y$ **such that** $\{x, y\}$ **is an edge, which implies** $\{x, y\}$ **is not an edge in** $\overline{G}$ **and** S **is not a clique on** $\overline{G}$**. Thus, there exists no clique of size** k **in** $\overline{G}$**.**