CSE200: Computability and Complexity
Homework 3
Instructor: Daniele Micciancio
Fall 2007. Due on Thu. Nov. 15

In this homework we consider the following four problems:

• Given two undirected graphs $G_1 = (V_1, E_1)$ and $(G_2 = (V_2, E_2)$, we say that $G_1$ is a subgraph of $G_2$ if there is an injective function $\phi : V_1 \rightarrow V_2$ such that for any two nodes $a, b \in V_1$, we have $(a, b) \in E_1$ if and only if $(\phi(a), \phi(b)) \in E_2$. The language SUBGRAPH-ISOMORPHISM is the set of all pairs of graphs $(G_1, G_2)$ such that $G_1$ is a subset of $G_2$.

• A boolean formula is an expression built up from a set of variable symbols $V$ using the operations $\neg, \land$ and $\lor$. E.g., $\neg(x_1 \land (x_2 \lor x_1)) \land x_2$ is a boolean formulat. A boolean formula $F$ is satisfiable if there is a truth assignment $\sigma : V \rightarrow \{0, 1\}$ such that $\sigma(F) = 1$, where the value of $F$ under the assignment $\sigma$ is defined by interpreting the logical operations in $F$ in the natural way. The language SATISFIABILITY is the set of all boolean formulas $F$ such that there exist an assignment $\sigma$ for which $\sigma(F) = 1$.

• Let $A \in \mathbb{Z}^{m \times n}$ be an integer matrix, and $\vec{b} \in \mathbb{Z}^{m}$ an integer vector. A solution to the integer programming problem $(A, b)$ is an integer vector $\vec{x} \in \mathbb{Z}^{n}$ such that $Ax \leq b$, where $Ax$ is the usual matrix vector multiplication, and $\leq$ holds componentwise for every row. INTEGER-PROGRAMMING is the set of all integer programs $(A, b)$ that have a solution.

**Problem 1: From graphs to formulas**

Give a polynomial time map reduction from SUBGRAPH-ISOMORPHISM to SATISFIABILITY. No implementation is required. However, as part of your solution, you should

• Give a mathematical definition of the mapping function, i.e., specify the input/output relation of a valid mapping function from pairs $(G_1, G_2)$ to formulas $F$.

• Prove that the mapping function $f$ you specified is correct, i.e., $(G_1, G_2) \in$ SUBGRAPH-ISOMORPHISM if and only if $f(G_1, G_2) \in$ SATISFIABILITY.
• Specify an encoding of SUBGRAPH-ISOMORPHISM and SATISFIABILITY instances as lists. (Many different encodings are possible, and they are all equally good. You can pick the encoding you like best.)

• Give a high level description of how the mapping function $f$ could be implemented as a WHILE program. “High level” means that you can use pseudocode and expressions like “scan the input”, or “for every pair (a,b) in the input list”, etc. Your description should have just enough detail so that the running time of the corresponding program can be analyzed.

• Analyze the running time of your program and show that it runs in polynomial time. Your analysis is not required to be tight. Any polynomial upper bound on the running time is fine.

**Problem 2: From formulas to linear systems**

Give a polynomial time map reduction from SATISFIABILITY to INTEGER-PROGRAMMING. See problem 1 for details about how to write your solution. You should use the same encoding for SATISFIABILITY as used in problem 1.

**Optional problem: Closing the loop**

This problem is provided just in case you find the first two not challenging enough, and need something more to think about. I expect everybody to read the question, most of you to think about it, and only very few (quite possibly none) to actually solve it. There is no formal credit associated to this problem, so make sure your solution to problems 1 and 2 is correct, and properly written before attempting to solve this one.

In the first two problems you proved that INTEGER-PROGRAMMING is at least as hard as SATISFIABILITY and the latter is at least as hard as SUBGRAPH-ISOMORPHISM. But are the three problems equivalent? Or some of the problems are strictly harder then the others?

Try to give a polynomial time map reduction from INTEGER-PROGRAMMING to SUBGRAPH-ISOMORPHISM. Such reduction, if it exist, would establish that all three problems are equivalent, and either none or all of them are solvable in polynomial time.