CS3 Computability and Intractability (2012-2013)
Exercise Sheet Friday 23 November: Sample Solutions

1. (a) Exact-3-Sat is in NP since it is a special case of Sat. We reduce 3-Sat to Exact-3-Sat as follows. Given an instance ϕ of 3-Sat note first of all that if ϕ has an empty clause then it is unsatisfiable. In this case we can send ϕ to a formula such as $(x \vee x \vee x) \wedge (\neg x \vee \neg x \vee \neg x)$. Otherwise if any clause has less than three literals just repeat one that occurs in it so that it now has exactly three to produce an instance ϕ' of Exact-3-Sat. Clearly an assignment satisfies ϕ if and only if it satisfies ϕ' and the reduction is polynomial time (in fact linear time).

   Guidelines: 1 mark for observing that the problem is in NP, two marks for the rest. [*3 marks*]

   (b) The reduction of the previous part does not do the job. We can modify it by introducing new variables rather than repeating literals. However this is not enough since now we might turn an unsatisfiable formula into a satisfiable one (e.g., consider $x \wedge \neg x$). We must also add new clauses to ensure that the formula ϕ' can only be satisfied by having at least one of the literals of ϕ set to true in each of its clauses. We can employ a simple observation: if ψ is any boolean formula and z a new variable then each satisfying assignment of ψ (if any) corresponds to a satisfying assignment of $(\psi \vee z) \wedge (\psi \vee \neg z)$. We can use this to replace clauses with two literals by clauses with exactly three. Likewise we can replace clauses with one literal by clauses with two and then with three literals. Note that this general observation also deals with the situation when ϕ has an empty clause; we iterate the 'trick' three times. It is clear that this reduction is polynomial time.

   Guidelines: This part is here to help students avoid an obvious ptifall in the first part and get to them to think about the simple trick that is possible there. No marks here if they do not see the pitfall. Otherwise interpolate if they see it but don't give any further details. Don't insist on absolutely complete details. [*4 marks*]

   (c) Let ϕ be an instance of Exact-3-Sat with variables $X_1, X_2, \ldots, X_n$. We build an instance of 3-ProdEqns as follows: take integer variables $x_1, x_2, \ldots, x_n$. Send the literal X_i to $1 - x_i$ and $\neg X_i$ to x_i. A clause $L_1 \vee L_2 \vee L_3$ is sent to $l_1 l_2 l_3$ where l_i is the image of the literal L_i as described before. Note that each clause is sent to a 3-product. Observe that an instance of 3-ProdEqns has a solution if and only if it has one with the variables taking values from $\{0, 1\}$. It is now straightforward to see that ϕ is satisfiable if and only if the derived instance of 3-ProdEqns has a solution. The reduction is clearly polynomial time. It follows that 3-ProdEqns is NP-hard. To deduce that it is NP-complete we must show that it is in NP. This is easily seen since we know that if there is any solution at all then there is one with the variables taking values from $\{0, 1\}$. We thus guess an assignment and check if in each product there is a factor of 0 (this takes linear time).

Guidelines: 4 marks for the reduction. 1 mark for deducing that it is NP-complete. [*5 marks*]

2. (a) Let G, k be an instance of VC. Let $V(G)$ and $E(G)$ be the vertex set and edge set of G respectively. We define a new graph G' as follows: augment $V(G)$ by adding, for each edge $e = (u, v)$ in $E(G)$, a new vertex v_e in G. Augment $E(G)$ by including, for each such edge e, (u, v_e) and (v, v_e) as edges. Clearly, the new graph G' can be constructed in polynomial time from G. The output of the reduction is G', k.

   It is not hard to see that this is indeed a reduction from VC to DS. If a set S of vertices is a vertex cover in G, it is also a dominating set in G'. Conversely, we can assume without loss of generality that the smallest dominating set S' in G' does not contain any vertices of the form v_e for $e \in E(G)$, and then the same set constitutes a vertex cover in G.

   Guidelines: Award 3 marks for the reduction and 2 marks for the rest. [*5 marks*]

   (b) We need to show that DS is in NP, which is clearly the case. (Given a proposed dominating set all we have to do is check for each vertex of the graph that it is adjacent to a vertex in the dominating set, taking no more than $\mathrm{O}(n^2)$ time.)

   Guidelines: Award 1 mark for the additional fact and a further mark for its justification. [*2 marks*]

   (c) The problem is not NP-complete unless P = NP. There are $\binom{n}{10} = \mathrm{O}(n^{10})$ subsets of 10 vertices so that we can check for the existence of a dominating set of size 10 in polynomial time.

   Guidelines: Award 2 marks for the observation and a further 2 marks for the justification. [*4 marks*]

3. Let $G = (V, E)$ and k be an instance of COLOURABILITY. We create an instance of TIMETABLE as follows. The set of papers P has one paper corresponding to each vertex in V. Corresponding to each edge $e_i \in E$ there is a candidate c_i. If $e_i = (u, v)$ then the set of papers P_i that c_i is expecting to sit consists of the papers corresponding to the vertices u and v. Let $S = \{1, 2, \ldots, k\}$ denote the set of timetable slots. It is now easy to prove that G is k-colourable iff there is a timetable with no clashes.

   Guidelines: Award 6 marks for a clear description of the reduction, 4 marks for arguing the if and only if bit, and 1 mark for saying that the reduction is polytime. [*11 marks*]

Rahul Santhanam, December 2012