CS3 Computability and Intractability (2012-2013)
Exercise Sheet 3

The deadline for this coursework is 4pm on Friday 23 November; submit your solutions to the ITO. Please note that multiple submissions are not allowed. Your work will be marked and returned, but the mark will not contribute to the overall mark for the course. Note that the marks for questions are not always related to their length or difficulty.

In your answers you should aim for clarity, conciseness and correctness; look at your answers with an objective eye, e.g., imagine that somebody else gave them to you to check. You should therefore make an early start to give yourself time to consider your answers before submitting the final version. This coursework should take about 10 hours of work to complete.

1. The Exact-3-Sat problem is the special version of 3-Sat in which each clause must have exactly three literals.

   (a) Prove that Exact-3-Sat is NP-complete. [*3 marks*]

   (b) Does your proof of the preceeding part still hold if we further insist that within each clause a literal must not be repeated? If not explain briefly how you would modify your proof so that it holds. (*Hint:* Consider formulae such as $x \vee \neg x$ as test cases for your argument; but it is *not* acceptable just to discuss test cases, you must give a general proof.) [*4 marks*]

   (c) Let $x_1, x_2, \ldots, x_n$ be variables over the integers. By a *3-product* we mean an expression of the form $\pm(a_1 - x_i)(a_2 - x_j)(a_3 - x_k)$ where $1 \leq i \leq j \leq k \leq n$ and $a_1, a_2, a_3 \in \{0, 1\}$.

   The *3-Product Equations* problem, 3-ProdEqns, is the following.

   Instance: Variables $x_1, x_2, \ldots, x_n$ and a finite number of 3-products using these variables.

   Question: Does the system have a solution, i.e., is there an assignment of integer values to the variables such that all the products evaluate to zero?

   Show that Exact-3-Sat is reducible in polynomial time to 3-ProdEqns. Deduce that 3-ProdEqns is NP-complete.

   Note: You may use without proof the fact that an instance of 3-ProdEqns has a solution if and only if it has one with the variables taking values from $\{0, 1\}$. [*5 marks*]

2. The *Dominating Set* problem, DS, is the following.

   Instance: A graph $G = (V, E)$ and integer k.

QUESTION: Does G contain a dominating set of size k? (A *dominating set* is a set $U \subseteq V$ such that every vertex of $V \setminus U$ is adjacent to at least one vertex in U.)

(a) Demonstrate that VC $\leq_P$ DS, where VC is the Vertex cover problem. *[5 marks]*

(b) It is known that VC is NP-complete. Which additional fact would enable you to deduce that DS is also NP-complete? Provide a brief justification for this additional fact. *[2 marks]*

(c) Is the following problem likely to be NP-complete? Justify your answer.

INSTANCE: A graph $G = (V, E)$.
QUESTION: Does G contain a dominating set of size 10? *[4 marks]*

3. The following problem, TIMETABLE, arises in creating a timetable for examination papers.

INSTANCE: A set P of papers, a set S of timetable "slots", a set $C = \{c_1, \ldots, c_k\}$ of candidates, and, for each candidate c_i, a set $P_i \subseteq P$ of papers that the candidate is expecting to sit.

QUESTION: Is there an assignment of the examination papers to timetable slots that avoids clashes? (A clash occurs if some candidate is required to sit two papers simultaneously.)

By presenting a reduction from COLOURABILITY (or otherwise) show that TIMETABLE is NP-hard. *[11 marks]*

Rahul Santhanam, Monday 12 November