Solutions to Midterm 2

1. Prove or disprove:

   (a) Let M be a deterministic Turing machine that, on inputs of length n, uses space $O(n^2)$. Then for every input x, and every configuration C of the computation of M on input x, $K(C) \leq O(|x|)$ there K is Kolmogorov complexity and $|x|$ is the length of x.

   (b) Let M be a deterministic Turing machine that, on inputs of length n, uses space $O(n^2)$ and time $O(n^3)$. Then for every input x, and every configuration C of the computation of M on input x, $K(C) \leq O(|x|)$ where K is Kolmogorov complexity and $|x|$ is the length of x.

   [30 points]

   SOLUTION OUTLINE:

   (a) The statement is false. Consider a machine M which simply looks at the length n of the input and then enumerates all possible strings of length n^2 on its tape. Since there is a string C of length n^2 such that $K(C) \geq O(n)$, we get a contradiction.

   (b) This statement is true. Any configuration C of the machine can be describes by specifying the code of the machine, the input and the time at which the configuration occurs. Since the code of the machine is a constant, the time can be specified in $O(\log |x|^3) = O(\log x)$ bits and the input is $|x|$ bits, this gives $K(C) \leq O(|x|)$.

2. We define #SAT to be the language

   $$\#\text{SAT} = \{\langle \varphi, k \rangle \mid \varphi \text{ has exactly } k \text{ satisfying assignments}\}$$

   Show that:

   (a) #SAT is **coNP** hard (recall that $\mathbf{coNP} = \{L \mid \overline{L} \in \mathbf{NP}\}$).

   (b) #SAT $\in$ **PSPACE**.

   [40 points]

   SOLUTION OUTLINE:

   (a) To show #SAT is **coNP** hard, note that $\varphi \in \overline{SAT} \Leftrightarrow \langle \varphi, 0 \rangle \in \#\text{SAT}$ as $\overline{SAT}$ is the set of all the formulas which are unsatisfiable i.e. have exactly 0 satisfying assignments. This shows that $\overline{\text{SAT}} \leq_p \#\text{SAT}$ which proves the claim.

   (b) We enumerate all possible assignments to the variables and for each assignment, we can check if it satisfies the formula. We also keep a count of the satisfying assignments. Since it takes n bits to store an assignment, n bits to store the counter (as there can be at most 2^n satisfying assignments), polynomial space to check if a given assignment satisfies the formula, the algorithm uses polynomial space overall.

3. Let $Th(\mathbb{N}, +, \leq)$ denote theory of the model whose universe is the set of natural numbers (including 0) and the relations are the usual $+$ and $\leq$ relations as defined on natural numbers. Show that $Th(\mathbb{N}, +, \leq)$ is **NP**-hard.
   [30 points]

   SOLUTION OUTLINE: We reduce 3SAT to deciding statements in $Th(\mathbb{N}, +, \leq)$. Let φ be a 3SAT formula with variables $x_1, \ldots, x_n$. We create a logical sentence with quantifiers $\exists x_1, \ldots, x_n$. To simulate $\overline{x}_i$, for each variable x_i, we add the clause $\exists \overline{x}_i \ (x_i + \overline{x}_i = 1)$. Here, we think of 1 as `true` and 0 as `false`. Note that since x_i and $\overline{x}_i$ are both non-negative, they can be only 0 or 1 by the above clauses.

   Finally, we two clauses for checking every clause of φ is satisfied. Say $c_j = (x_{i_1} \vee \overline{x}_{i_2} \vee x_{i_3})$ is a clause in φ. Then we add $\exists y_j \ (y_j = x_{i_1} + \overline{x}_{i_2}) \wedge (1 \leq y_j + x_{i_3})$. Note that we need to add an extra variable y_j for each clause c_j since the '+' relation only allows us to add two numbers at a time. It is easy to see that the reduction is polynomial time and the new sentence is true if and only if φ is satisfiable.