The Australian National University
Research School of Computer Science
Dirk Pattinson
Jinbo Huang

Semester 1, 2013
Assignment 3

Theory of Computation

Release Date: Monday 6 May 2013

Due Date: Friday 17 May 2013

Submission: Hand in to Jinbo Huang in class.

Note: Hand written answers are acceptable if written neatly. Correct answers may be given less than full credit if unnecessarily complicated.

Exercise 1 **Turing Machine Design** (A)

Design a Turing machine that shifts the entire input string to the right by one place, under the following conditions: The input alphabet is $\{0, 1\}$ and the Turing machine has a single tape with a single track. Describe the Turing machine in words, and draw its transition diagram.

Exercise 2 **Enumerating a Recursively Enumerable Language** (A)

Given a Turing machine M that accepts a language L, informally but clearly show that a Turing machine M' can be constructed to enumerate all members of L in the following sense: (i) Whenever M' enters a special state p, the string to the left of the tape head is a member of L, and (ii) every member of L appears on the tape at some point in the aforementioned way. Keep in mind that the Turing machine M may not halt on all inputs.

Exercise 3 **Proving Undecidability** (A)

Consider the following theorem and attempted proof thereof. Identify the flaw in the attempted proof, and give a correct proof.

Theorem: The set of all (encodings of) Turing machines that have a *useless* state is undecidable, where a useless state is defined as a state that is never visited on any input string.

Attempted proof (sketch): Reduce the universal language to this problem. Given (M, w), construct a Turing machine Q that replaces its input string with (M, w), simulates the universal Turing machine, and enters the accepting state q_{accept} if and only if the (simulated) universal Turing machine accepts (M, w). The reduction works as the Turing machine Q will have a useless state (q_{accept}) if and only if the Turing machine M does not accept the string w.

Exercise 4 **Proving Non-Recursive Enumerability** (A)

(Exercise 9.3.7b) Show that the following is not recursively enumerable: $\{(M_1, M_2) \mid L(M_1) \cap L(M_2) = \emptyset\}$, i.e., the set of pairs of Turing machines the intersection of whose languages is empty.

Exercise 5 **Boolean Encodings of Graph Properties** (A)

(Exercise 10.2.2b–d) Suppose G is an undirected graph of four nodes: 1, 2, 3, and 4. Let x_{ij}, for $1 \leq i < j \leq 4$, be a Boolean variable that we interpret as saying "there is an edge between nodes i and j." The expression $x_{12}x_{23}x_{34}x_{14} + x_{13}x_{23}x_{24}x_{14}+x_{13}x_{34}x_{24}x_{12}$, for example, says that the graph G has a Hamilton circuit. In general, a Boolean expression over the x_{ij} variables describes a property of the graph in the sense that a truth assignment to the variables satisfies the expression if and only if it describes a graph having that property. Write expressions for the following properties:

1. G contains a clique of size 3 (i.e., a triangle). 2 + 8 + 4 = 14 0 1 2 nonexisting edge in squre
2. G contains at least one node with no edges. 4*(1+3) + (2 + 4) + 0 + 1 = 23 1 2 3 4 nodes with no edges
3. G is connected. 64 - 23 = 41

Exercise 6 **Proving $\mathcal{NP}$-Completeness** (A)

(Exercise 10.4.4d) We know that the Node Cover problem is $\mathcal{NP}$-complete. Show that the following Dominating Set problem is $\mathcal{NP}$-complete: Given a graph G and an integer k, does there exist a subset S of at most k nodes of G such that each node is either in S or adjacent to a node of S?