CS 525: Theory of Computation
Midterm 2

Dustin Ingram

February 28, 2012

1. **Solution:**

   (a) To show that the set of nodes in T is countable, we must show that there is a one-to-one, onto function f for which $f(A) = B$ if:

   $$A = \text{the set of natural numbers } \{1, 2, 3, \dots\}$$
   $$B = \text{the set of nodes} \in T$$

   For the nodes in T, this function is simply $f(n) = n$.

   (b) Here, we let S represent the infinite list of all paths from the root, and thus let each s_1, s_2, s_3, etc. represent each unique path. Because any path can be represented as a unique sequence of 'directions' at each node, each path is equivalent to a series of 'lefts' and 'rights', since every node in T has exactly two children. Therefore, if we let 'left'$= 0$ and 'right'$= 1$, each s_n can be represented as a series of 0's and 1's.
   Based on the diagonalization argument, this also means that, for any given set S', it is possible to construct a sequence s_{OPP} from the 'diagonalization' of S such that $s_{0,n} = \mathit{opposite}(s_{n,n})$. This sequence s_{OPP} is therefore not contained in S, but a valid sub-seqence of S. Therefore, S is uncountable.

2. **Solution:** To test if a DFA A has no useless states, for every state $q_n \in A$, we create a new DFA A_n which has the same states and transitions as A, except q_n is the only accepting state. Then, if for any A_n, $L(A_n) = \emptyset$, reject (because q_n is a useless state); otherwise, accept.

3. **Solution:** If we assume that TM M decides L_3, then we can construct TM N such that it decides the (undecidable) reduction of L_3 as follows:

   N: on input (O, w)

   (a) Design TM P such that it only accepts the input word w;

   (b) Run $M(O, P)$;

(c) Accept if $M(O, P)$ accepts, reject otherwise.

Thus, $M(O, P)$ will only accept if $w \in L(O)$; however, this would also decide the reduction of M, and thus is a contradiction.

4. **Solution:** Since the complement of M must also be non-Turing recognizable, this would mean that M accepts all input except some single input w, which might never be found by M, and thus M is not Turing recognizable.

5. **Solution:** To prove that any infinite subset of MIN_{TM} is not Turing recognizable, let us first assume that there is some Turing recognizable subset $S \in \text{MIN}_{TM}$; each element $n \in S$ is therefore enumerable by some TM N. We then design a Turing machine as follows:

   M: on input (N, w)

   (a) Obtain the description of M[1]
   (b) Determine the length m of M
   (c) Use N to list all $n \in S$
   (d) For each N determine the length n of N
   (e) If $n > m$, simulate $N(w)$

   The result is a TM N which recognizes the same langague as M, but has a longer length, thus resulting in a contradiction.

6. **Solution:**

   (a) $\forall x \exists y [x \cdot y = 1] \notin \text{Th}(\mathbb{N}, \cdot)$
   Reason: This can be proven by letting $x > 1$; there is no natural number by which we can multiply x to produce 1.

   (b) $\forall x \exists y [x \cdot y = 1] \in \text{Th}(\mathbb{Q}, \cdot)$
   Reason: This proven based on the definition of rational numbers; if x is a rational number, it can be represented by the quotient of two integers.

   (c) $\forall x, y \exists z [z \cdot z + x = y] \in \text{Th}(\mathbb{R}, +, \cdot)$
   Reason: This implements the operator 'less than or equal to' such that $x \leq y$; no matter what z is, its square can represent the positive difference between two rational numbers.

[1]Theorem 6.3, pg. 220