CS 525: Theory of Computation
Final Exam

Dustin Ingram

March 20, 2012

1. **Solution:**

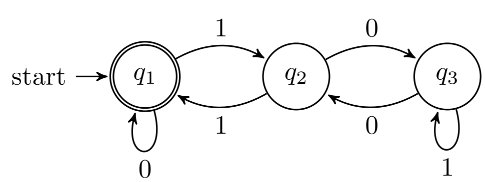
    (a) The variables in G are $\{R, S, T, X\}$. The terminals in G are $\{a, b, \epsilon\}$. The start variable is R.

    (b) Two strings in G are 'ab' and 'ba'. A string not in G is 'a'.

    (c) True.

    (d) This language ensures that every string contains at least one 'a' and one 'b', and that for at least one character in the string $\{c_1, c_2, \ldots, c_n\}$ there is a character at some c_i such that the character at c_{n-i} is the opposite.

2. **Solution:**



3. **Solution:**

    (a) **intersection**: Let M_A and M_B be turing machines that accept languages A and B, respectively. Construct a third machine, M_C, as follows:

    $M_C =$ On input $\langle w \rangle$:

    i. Run M_A on w. If M_A halts and rejects, *reject*, otherwise continue;

    ii. Run M_B on w. If M_B halts and rejects, *reject*. If it accepts, *accept*.

(b) **concatenation**: Let M_A and M_B be turing machines that accept languages A and B, respectively. Construct a third machine, M_C, as follows:

   $M_C =$ On input $\langle w \rangle$:

   i. Non-deterministically partition w into every possible pair of substrings $\{w_1, w_2\}$;
   ii. For every pair $\{w_1, w_2\}$, run M_A on w_1 and M_B on w_2;
   iii. If, for any pair, M_A accepts w_1 and M_B accepts w_2, *accept*;
   iv. Otherwise, *reject*.

(c) **star**: Let M_A be a turing machine that accepts $L(A)$. Construct a second machine, M_{A^*} as follows: $M_{A^*} =$ On input $\langle w \rangle$:

   i. Non-deterministically partition w into every possible set of non-empty substrings $\{w_1, w_2 \ldots, w_n\}$;
   ii. Run M_A on all $w_i \in$ all possible partitions;
   iii. If, for any partition M_A accepts all w_i in the partition, *accept*, otherwise *reject*.

4. **Solution:** To show S_{DFA} is decidable, construct a new DFA $M^{\mathcal{R}}$ that accepts the reverse language $L^{\mathcal{R}}$ as follows:

   (a) Reverse the directions of all transitions in M;
   (b) Create a new start state, with ϵ-transitions from this state to all accepting states in M;
   (c) Make the start state of M the accepting state of $M^{\mathcal{R}}$
   (d) Transform the resulting NFA into the DFA S_{DFA}.[1]

5. **Solution:** $L = \{\langle M, N\rangle | M, N$ are turing machines, M uses an oracle to determine if N is empty$\}$

6. **Solution:**

   (a) Construct a machine M as follows:

   $M =$ On input $\langle G, a, b, k \rangle$:

   i. If $k = 0$, *reject*;
   ii. For each node c adjacent to a that has not already been considered:
      A. If b is c, *accept*;
      B. Otherwise run M on $\langle G, c, b, k-1 \rangle$, if it accepts, *accept*.
   iii. Otherwise, *reject*.

[1] Theorem 1.39, pg. 55

(b) Assume M decides LPATH in polynomial time. Then we can construct a DFA M' which can decide HAMPATH in polynomial time as follows:

$M' = \langle G, a, b \rangle$:

i. Run $M\langle G, a, b, n \rangle$, where n is the number of nodes in G;

ii. If M accepts, *accept*, otherwise *reject*.

This is contradictory, as we know HAMPATH is NP-complete.[2]

[2] Theorem 7.46, pg. 286