Turing Machines

8 Apr 2019

1. Design a Turing Machine for accepting $\{a^i b^j c^k | i, j, k \geq 1, i = j + k\}$. Draw the transition diagram and list all the transitions.
   Solution $M =$ On input w:
   (a) It matches a's and b's.
   (b) Read the first a and change it to X
   (c) Move right until b
   (d) Read the first b and change it to Y
   (e) Move left until X
   (f) Repeats steps (b) to (e) until all b's are exhausted.
   (g) After that, it matches a's with c's in a similar manner
   (h) The machine *accepts* when the number of a's is equal to the sum of the number of b's and the number of c's. Otherwise *rejects*

2. Define a two-headed finite automaton (2DFA) to be a deterministic finite automaton that has two read-only, bidirectional heads that start at the left-hand end of the input tape and can be independently controlled to move in either direction. The tape of a 2DFA is finite and is just large enough to contain the input plus two additional blank tape cells, one on the left-hand end and one on the right-hand end, that serve as delimiters. A 2DFA accepts its input by entering a special accept state. (to submit)
   (a) Describe how a 2DFA can recognize $\{a^n b^n c^n \mid n \geq 0\}$.
   (b) Let $A_{2DFA} = \{\langle M, x\rangle | M$ is a 2DFA and M accepts $x\}$. Show that A_{2DFA} is decidable. (Hint: if the 2DFA has s states and input size is x, compute an upper bound on the number of steps to simulate for acceptance.)

   Solution
   (a) Easy.
   (b) There are at most $s(x + 2)^2$ distinct configurations of the 2DFA. Hence if it accepts, it will inside that many steps.

3. Prove that EQ_{DFA} is decidable by testing the two DFAs on a finite number of strings. Calculate a number that works.
   Solution Let M_1 and M_2 be two given DFAs and let n_1 and n_2 be their respective numbers of states. We claim that $L(M1) = L(M2)$ if and only if for all strings w with $|w| \leq n_1 n_2$ we have $w \in L(M_1)$ if and only if $w \in L(M_2)$.
   The two languages differ such that $|t| = l > n_1 n_2$. Consider the sequence of states $q_0, \cdots, q_l$ and $r_0, \cdots, r_l$ that M_1 and M_2 enters respectively on input t. As M_1 has n_1 states and M_2 has n_2 states, only $n_1 n_2$ distinct pairs $\langle q, r \rangle$ exist where q is a state in M_1 and r is a state in M_2. By the pigeon hole principle, two pairs of states $\langle q_i, r_i \rangle$ and $\langle q_j, r_j \rangle$ must be identical because $l > n_1 n_2$. If we remove the portion of t from i to $j - 1$, we obtain a shorter string on which M_1 and M_2 differs even though t was supposed to be the shortest. This is a contradiction and thus our assumption $|t| > n_1 n_2$ is false.

4. Let $C = \{\langle G, x\rangle | G \text{ is a CFG}, x \text{ is a substring of some } y \in L(G)\}$. Show that C is decidable.
   Solution Consider the following Turing machine
   $M =$ On input $\langle G, x\rangle$, where G is a CFG

   (a) Let R be the following RE $R = \Sigma^* x \Sigma^*$
   (b) Create the CFG H such that $L(H) = L(G) \cap L(R)$
   (c) Submit $\langle G\rangle$ to the decider for E_{CFG}
   (d) If it accepts, then *reject*
   (e) Else, *accept*.

   M is clearly a decider since it halts. Also, M accepts $\langle G, x\rangle$ iff G generates some string in $\Sigma^* x \Sigma^*$ – i.e., some string with x as a substring.

5. A useless state in a pushdown automaton is never entered on any input string. Consider the problem of determining whether the pushdown automaton has any useless states. Formulate this problem as a language and show that it is decidable.(Home)
   Solution Let $\mathcal{P}$ be the set of all pushdown automata. Let the language $\mathcal{U} = \{x \in \mathcal{P} | x \text{ has a useless state}\}$. To show that $\mathcal{U}$ is decidable we design a Turing machine which accepts only strings in $\mathcal{U}$. We know that whether a PDA has an empty language is decidable, and we may reduce the question of whether a given state q is useless to this question by making q the only accept state and then determining whether the resulting PDA has an empty language. If it does, then q is a useless state. Our Turing machine may therefore solve the question of whether there are any useless states by performing this test for each state in order.

   Alternately, a direct solution is also possible where you check the reachability of each state from the starting state. But keep in mind that the transition function should take into account all possible transitions caused by the available stack alphabet also.

6. Let M be a Turing Machine with one semi-infinite tape and two read/write heads. A transition of M is of the form -
$\delta(q, a, b) = (q', c, d, D, D_2)$ where $D_1, D_2 \in L, R$.
If both the heads point to the same tape cell, the symbol is replaced by c. The first head moves in the direction D_1 and the second head moves in the direction D_2. Show that this two-head Turing machine can be simulated by a standard Turing machine M_1 with a semi-infinite tape with a single track and only one read/write head. (Home)