CS 373 Homework 8 Anchu A. Lee
November 20, 2017

1. This exercise concerns TM M_2 , whose description and state diagram appear in Example 3.7. In each of the parts, give the sequence of configurations that M_2 enters when started on the indicated input string.

   (a) 0.
   $q_1 0\sqcup \quad \sqcup q_2 \sqcup \quad \sqcup\sqcup q_{accept}$

   (b) 000.
   $q_1 000\sqcup \quad \sqcup q_2 00\sqcup \quad \sqcup\sqcup q_3 0\sqcup \quad \sqcup\sqcup 0 q_4 \sqcup \quad \sqcup\sqcup 0\sqcup q_{reject}$

2. This exercise concerns TM M_1 , whose description and state diagram appear in Example 3.9. In each of the parts, give the sequence of configurations that M_1 enters when started on the indicated input string.

   (a) 1#1.
   $q_1 1\#1\sqcup \quad \mathrm{x} q_3 \#1\sqcup \quad \mathrm{x}\# q_5 1\sqcup \quad \mathrm{x} q_6 \#\mathrm{x}\sqcup \quad q_7 \mathrm{x}\#\mathrm{x}\sqcup \quad \mathrm{x} q_1 \#\mathrm{x}\sqcup \quad \mathrm{x}\# q_8 \mathrm{x}\sqcup \quad \mathrm{x}\#\mathrm{x} q_8 \sqcup \quad \mathrm{x}\#\mathrm{x}\sqcup q_{accept}$

   (b) 1##1.
   $q_1 1\#\#1\sqcup \quad \mathrm{x} q_3 \#\#1\sqcup \quad \mathrm{x}\# q_5 \#1\sqcup \quad \mathrm{x}\#\# q_{reject} 1\sqcup$

3. Describe a Turing machine, sequence of steps, that recognizes $\{w \mid w$ is an element of $\{a, b, c\}^*$ such that the number of a's in $w <$ the number of b's in w and the number of a's in $w =$ the number of c's in $w\}$

   (1) Place symbol at the left side of tape
   (2) Scan right for a, if found: mark it, else: go to step 6
   (3) Rewind
   (4) Scan right for b, if found: mark it, else: Halt and Reject (a must be $< b$)
   (5) Rewind and go to step 2.
   (6) Rewind
   (7) Scan right for a', if found: mark it, else: go to step 11
   (8) Rewind
   (9) Scan right for c, if found: mark it, else: Halt and Reject (a must be $= c$)
   (10) go to step 6.
   (11) Scan right for c, if found: Halt and Reject, else: Halt and Accept.

4. Show the equivalent transitions for a 2-PDA for the Turing machine transitions $(q_i, X) \to (q_j, A, L)$ and $(q_i, X) \to (q_j, A, R)$ (in state q_i read X, write A, and move left or right and transition to state q_j). The transitions for a 2-PDA are of the form $(q_i, X, S_1, S_2) \to (q_j, T_1, T_2)$ (in state q_i, read X, pop S_1 from stack 1, pop S_2 from stack 2, transition to state q_j, push T_1 onto stack 1 and push T_2 onto stack 2). You don't have to prove the transitions are equivalent, just tell me what they are.

5. Give implementation-level descriptions of Turing machines that decide the following languages over the alphabet $\{0, 1\}$.
   $\{w \mid w$ does not contain twice as many 0's as 1's$\}$

   (1) Place symbol at left side of tape
   (2) Rewind
   (3) Scan right for 1, if found: mark it, else: go to step 9
   (4) Rewind
   (5) Scan right for 0, if found: mark it, else: Halt and Accept
   (6) Rewind
   (7) Scan right for 0, if found: mark it, else: Halt and Accept
   (8) Go to step 2.
   (9) Rewind
   (10) Scan right for $0'$, if found: Halt and Reject, else: Halt and Accept

6. Prove the class of Turing recognizable languages is closed under the union operation (construction and proof)

Proof. Let M_1 and M_2 be two Turing machines that recognize languages L_1 and L_2 respectfully. Then let M_3 be a machine that will run input w alternately between machines M_1 and M_2. If a machine accepts, M_3 accepts. If both machines reject then M_3 rejects. As $w \in L_1 \cup L_2$, the string can be $w \in L_1$, then the M_1 portions of M_3 will accept it. If the string is $w \in L_2$, then the M_2 portions of M_3 will accept it. If the string is $w \notin L_1 \cup L_2$ then $w \notin L_1$ and $w \notin L_2$ and so M_3 will not accept w. Therefore M_3 recognizes $L_1 \cup L_2$. □

7. Prove the class of decidable languages is closed under concatenation (construction and proof)

   Proof. Let M_1 and M_2 be two Turing machines that recognize languages L_1 and L_2 respectfully. Then let M_3 be a machine that will run input w on M_1 and M_2 by splitting w into every possible two parts. If both machines accept then M_3 accepts. If not, then the M_3 continues to the next two substrings.That means every possible combination of two substrings of the string w will be run through M_1 and M_2. When all substrings are tried and did not reach an accepting state, then reject w. That way the w must be $L_1 \circ L_2$ as the first substring is in M_1 and the second substring will be accepted by M_2. Otherwise w will be rejected. □

8. Prove the class of decidable languages is closed under intersection (construction and proof)

   Proof. Let M_1 and M_2 be two Turing machines that recognize languages L_1 and L_2 respectfully. Then let M_3 be a machine that will run input w on M_1 then M_2. If both machines reject, then w is not in the languages of $L_1 \cap L_2$. If one or more machines accepts then w is in the language. □

9. Prove the class of Turing recognizable languages is closed under the star operation (construction and proof)

   Proof. Let M_1 be a Turing machine that recognizes the language L_1 and w be a string of the form L_1^*. Then have M_2 be a machine that splits the input into individual cuts of the input. e.g. $s = s_1 s_2 s_3 ... s_n$ for n can be from 0 to the length of s. M_2 then runs each substring into M_1. If it rejects then M_2 tries the next cuts of the string. If M_2 accepts for all cuts of a string, then the language is accepted. As M_2 tries every possible split for the input, it will eventually find the right match for L_1 and therefore recognizes L_1^*. □

10. Show that a language is decidable iff some enumerator enumerates the language in the standard string order.

    Proof. ($\Rightarrow$) There must exist a TM M that decides L, a decidable language. M can reconstruct the enumerator E. Define a TM D that: (ignores input)

    (a) Set an iterator i to one of $1, 2, 3, ...$
    (b) Get the ith string of the languages L, w_i
    (c) Run machine M with string w_i. Write w_i to tape if M accepts.

    D outputs all strings in L in standard string order.
    ($\Leftarrow$) For a language L with enumerator E, constructor a TM M for L which decides the language. M then can be run with input w:

    (a) Run through E and save the strings outputted to tape as $w_1, w_2, w_3 ...$
    (b) Check the tape against the input string w. If $w = w_i$ for some i, then halt and accept.
    (c) If the machine encounters a string w_i from the E that comes after the input string w in standard string order, then halt and reject.

    □