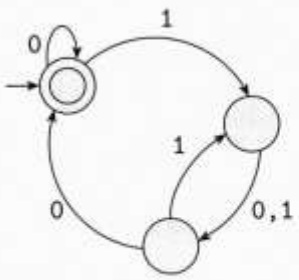
Theory of Computation, CSCI 438 spring 2022
Decidability: Problems Concerning Finite Automaton, pg. 193-197, April 4

1.

A4.1 Answer all parts for the following DFA M and give reasons for your answ



a. Is $\langle M, 0100 \rangle \in A_{\text{DFA}}$?
b. Is $\langle M, 011 \rangle \in A_{\text{DFA}}$?
c. Is $\langle M \rangle \in A_{\text{DFA}}$?
d. Is $\langle M, 0100 \rangle \in A_{\text{REX}}$?
e. Is $\langle M \rangle \in E_{\text{DFA}}$?
f. Is $\langle M, M \rangle \in EQ_{\text{DFA}}$?

a. Yes
b. No
c. No
d. No
e. No
f. Yes

2. The following is called the acceptance problem for DFAs.

   A_{DFA} = {<D,w> | D is a DFA, w is a string in the language of D, and D accepts w}

   (page 194)

   Is there an algorithm that decides A_{DFA}? If so, give it and tell if it is an a decider or a recognizer.

   Yes, there is a decider.

   Version 1:
   M= "On input <D,w> that encodes a DFA and a string in the language of the DFA,

   1. Call the start state the current state.
   2. Repeat for each symbol in w beginning with the first symbol
      i. Take current state and the symbol, run through the transitions to find the new state given this current state and symbol. Make the new state be the current state.
   3. Accept if the current state is a final state. Otherwise, reject."

   Version 2:
   M= "On input <D,w> that encodes a DFA and a string in the language of the DFA,

   1. If the input doesn't correctly encode a DFA and a string, reject.
   2. Simulate the DFA on w.
   3. If the DFA accepts w, accept. If the DFA rejects w, reject."

   Version 3:
   M= "On input <D,w> that encodes a DFA and a string in the language of the DFA,

   1. Simulate the DFA on w.
   2. If the DFA accepts w, accept. If the DFA rejects w, reject."

   The algorithm is the same in each version, only the level of detail and what is assumed is different.

3. Consider the acceptance problem for NFAs,

   A_{NFA} = {<D,w> | D is an NFA, w is a string in the language of NFA and $w \in \mathcal{L}(D)$}

   Is A_{NFA} decidable.

   Yes, there is a decider. Here is one decider:

   M= "On input <D,w> that encodes a NFA and a string in the language of the NFA,

   1. If the input doesn't correctly encode a DFA and a string, reject.
   2. Using the method described in class and the text, convert the NFA to a DFA.
   3. Simulate the DFA on w. (I.e. use the algorithm in problem 1.)
   4. If the DFA accepts w, accept. If the DFA rejects w, reject."

4. Empty DFA.

E_{DFA} = {<D> | D is a DFA whose language is empty}

Is E_{DFA} decidable?

Can create a recognizer for E_{DFA} by feeding all strings in $\sum^*$ into D and rejecting if any of these strings are accepted. If $\mathcal{L}$(D) is empty, however, this algorithm will loop forever since $\sum^*$ is infinite.

This problem is very similar to the graph problem in the previous exercises. A decider for this problem is:

M= "On input <D> that encodes a DFA

1. Mark the start state.
2. Repeatedly, run through all transitions marking all states that can be gotten to from a marked state. If a final state is reached, reject. When going through all of the transitions no longer yields additional marked states and no marked states are marked, accept."

5. Equivalent DFAs.

   EQ_{DFA} = {<D,E> | D and E are DFAs and they both recognize the same language}

   Is EQ_{DFA} decidable?

   Two sets A and B are equivalent iff $(A \cap \overline{B}) \cup (\overline{A} \cap B) = \Phi$. The following algorithm uses this to decide E_{DFA}.

   M = "On input <D,E> which are DFAs.

   1. Create a DFA D' that accepts the complement of $\mathcal{L}$(D) and another DFA E' that accepts the complement of $\mathcal{L}$(E), that is B'.
   2. Create a DFA for the intersection of the two DFAs D and E' using the method described in class and in the text. Similarly, create a DFA for the intersection of the two DFAs E and D'.
   3. Create a DFA that will accept the union of the above two DFAs using the method described in class and in the text.
   4. Feed the resulting DFA into the algorithm for E_{DFA}. If it is accepted, accept. If it is rejected, reject."