Problem

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. For example, a 2DFA can recognize the language $\{a^n b^n c^n | n \geq 0\}$.

a. Let $A_{\text{2DFA}} = \{\langle M, x\rangle |\ M \text{ is a 2DFA and } M \text{ accepts } x\}.$ Show that A_{2DFA} is decidable.

b. Let $E_{\text{2DFA}} = \{\langle M\rangle |\ M \text{ is a 2DFA and } L(M) = \emptyset\}.$ Show that E_{2DFA} is not decidable.

Step-by-step solution

Step 1 of 2

Decidability of 2DFA

Consider a Turing machine to check whether *M* the 2DFA accept the input *x*. if any configuration is repeated in M then it will not terminate because it is a deterministic finite automata.

Consider a Turing machine *W* which encodes M the 2DFA and input *x*. It also simulate M on *x* and check *M* accepts *x*. *W* has four tapes.

- Input tape to store input.
- Work tape, which has two bidirectional head to read.
- Another work tape to store configuration occurs during simulation.
- Scratch tape used to create representation of *M* configuration which becomes helpful for work tape in searching and updating.

W Turing machine works as follow:

$W = \text{on input} \langle M, x\rangle, \text{where } M \text{ is a 2DFA and } x \text{ is a string}$

1. Check the input tape $\langle M, x\rangle$ has a proper legal encoding or not. If it does not contain legal coding, then reject and halt, otherwise continue.
2. Copy input to work tape. Initialize the two head of work tape so that they are their starting position. Also initialize the second work tape as empty.
3. When *M* current state has halt then accepts and halt the states. When *M* current state has no move then reject and halt the states.
4. Create configuration on scratch tape. When current configuration already exist in the work tape then reject and halt otherwise store configuration at the end of second work tape.
5. Simulate one move of *M* on input tape.
6. Move to step 3.

When the input *x* are accepted by 2DFA that is M, the simulation will completed this in finite number of steps, then *W* will accept the input. Otherwise, when input codes are not legal or M does not ends or terminates. *W* determines this in finite number of steps and rejects the input.

All this shows that language A_{2DFA} is decidable.

Comment

Step 2 of 2

Assume on contrary that E_{2DFA} is decidable. Consider *W* a decider Turing machine which decides the E_{2DFA}.

Now, create Turing machine E which is based on *W* for deciding E_{TM} which works as follow:

$E = \text{on input} \langle M\rangle$

Create another 2DFA *M'*. The accepting computation history of *M* is recognizing by this *M'*.

Execute *W* on *M'*. When *W* accepts, **accepts**. Else **reject**.

Since, the E_{TM} is not decidable therefore the contradiction occurs. Hence E_{2DFA} is undecidable.