---

**Problem 5.26.** 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 $a^n b^n c^n \mid n \geq 0$.

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

*Proof.* First, we define a notation for representing different configurations of a 2DFA, and then we calculate the number of distinct configurations of a 2DFA for an input of length $n$.

For a state $q$ and two strings $u$ and $v$ over the input alphabet $\Sigma$, we write $q \sqcup \circ\ u \bullet v\ \sqcup$ for the configuration where the current state is $q$, the current input is $uv$, the current first and second head locations are the first symbols of $u$ and $v$ respectively. For a 2DFA with $q$ states, there are exactly $q(n+2)^2$ distinct configurations for an input of length $n$. Construct decider $S$ for $A_{2DFA}$ as follows.

$S =$ "On input $\langle M, x\rangle$, where $M$ is a 2DFA and $x$ is a string:

1. Let $n$ be the length of string $x$.
2. Simulate $M$ on $x$ for $q(n+2)^2$ steps or until it halts.
3. If $M$ has halted, *accept* if it has accepted and *reject* if it has rejected. If it has not halted, *reject*."

□

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

*Proof Idea.* The proof is by reduction from $A_{TM}$. We show that if $E_{2DFA}$ were decidable, $A_{TM}$ would also be. For a **TM** $M$ and an input $w$, we can determine whether $M$ accepts $w$ by constructing a certain 2DFA $B$, such that the language that $B$ recognizes comprises all accepting computation histories for $M$ on $w$.

We construct $B$ to accept its input $x$ if $x$ is an accepting computation history $C_1, C_2, \ldots, C_l$ for $M$ on $w$. We assume that the accepting computation history is presented as a single string with the configurations separated from each other by the # symbol, such as $\#C_1\#C_2\#\ldots\#C_l\#$. Given and input $x$, $B$ determines whether the $C_i$'s satisfy the three conditions of an accepting computation history.

1. $C_1$ is the start configuration for $M$ on $w$.
2. Each $C_{i+1}$ legally follows from $C_i$.

3. $C_l$ is an accepting configuration for $M$.

The start configuration $C_1$ for $M$ on $w$ is the string $q_0 w_1 w_2 \ldots w_n$, where $q_0$ is the start state for M on w. $B$ has this string directly built in, so it is able to check the first condition. An accepting configuration is one that contains the $q_{accept}$ state, so $B$ can check the third condition by scanning $C_l$ for $q_{accept}$. To make sure that the second condition is satisfied, $B$ checks on whether $C_{i+1}$ legally follows from $C_i$. This step involves verifying that $C_i$ and $C_{i+1}$ are identical except for the positions under and adjacent to the head in $C_i$. These positions must be updated according to the transition function of $M$.

*Proof.* Assume that **TM** $R$ decides $E_{2DFA}$. Construct **TM** $S$ to decide $A_{TM}$ as follows.

$S =$ "On input $\langle M, w \rangle$, where $M$ is a **TM** and $w$ is a string:

1. Construct 2DFA $B$ from $M$ and $w$ as described in the proof idea.
2. Run $R$ on $\langle B \rangle$.
3. If $R$ rejects, $M$ accepts $w$, so *accept*. Otherwise, *reject*."

Thus, if **TM** $R$ exists, we can decide $A_{TM}$, but we know that $A_{TM}$ is undecidable[1]. By virtue of this contradiction, we can conclude that $R$ does not exist. Therefore, $E_{2DFA}$ is undecidable. □

[1]Theorem 4.11 $A_{TM}$ is undecidable.