CSE331 Automata and Computability

Lecture 5.1: Introduction to Finite Automata

Presenter: Warida Rashid *Scribe: Warida Rashid*

Finite Automata/ Finite State Machine

Finite Automata (FA) or Finite State Machine (FSM) is a mathematical model that recognizes patterns within input taken from some character set (or alphabet).

A Finite Automaton does the following things:

- Stores a finite amount of information
- Given a string of input symbol, it either accepts or rejects it

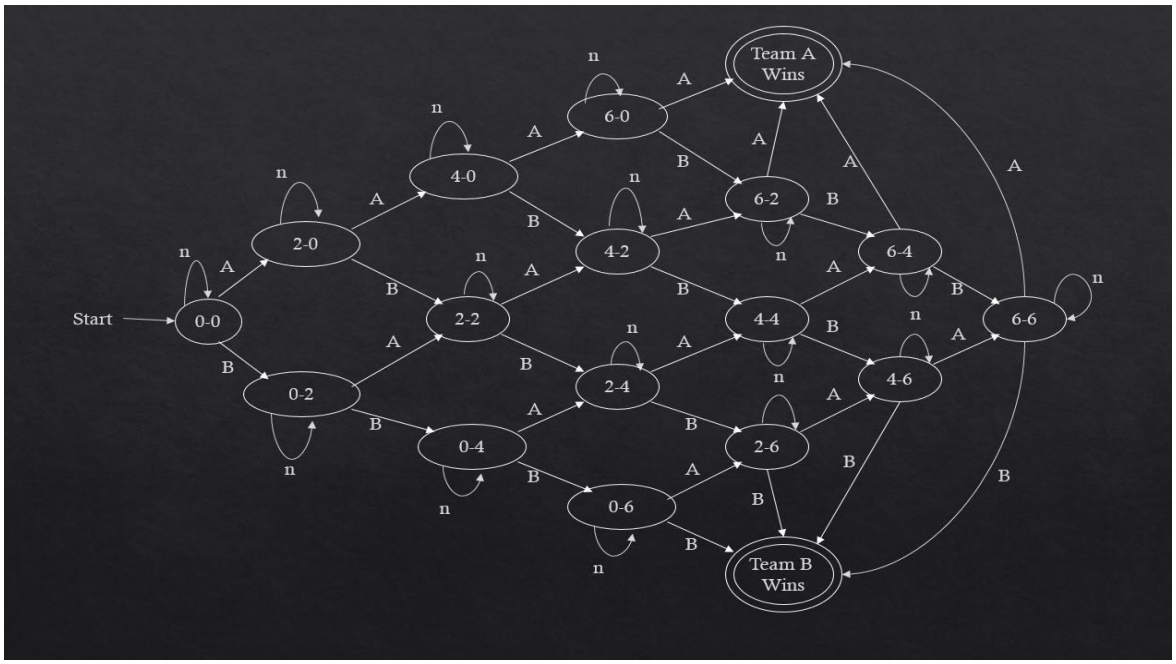
A Finite Automaton consists of:

- A finite number of states to represent a finite amount of information
- A designated start state
- A set of designated final states
- A set of rules that defines how the automaton moves from one state to another in response to input symbols

Example

Let's say we have to design an automaton for a buzzer quiz. The rules of the quiz are:

- The quiz is played between two teams: Team A and Team B.
- Both the teams are asked a question.
- Whichever team presses the buzzer first gets to answer the question.
- If the answer is correct, the team scores 2 points.
- If the answer is wrong, no one scores a point and the quiz moves on to the next question.
- The team that scores 8 points, wins.
- The quiz must have a winner. If a winner is not found, the quiz would be considered to be "Invalid".



Explanation:

1. The automaton is represented as a graph.
2. Each node represents a state.
3. The edges represent transitions from one state to another.
4. The label or name of the states represent the score of team A and team B in (Team A score - Team B score) format. The states store the information about the current scores of the teams.
5. The labels on the edges represent the team that scores a point.
6. There is a transition for each question depending on one of the 3 outcome :
    - Team A scores a point, marked as **A**
    - Team B scores a point, marked as **B**
    - None of the teams score a point, marked as **n**

Transitions

1. The automaton starts at state **0-0**, i.e. the scores of both team A and team B is 0 at the beginning.
2. After the 1st question at state **0-0** the following transitions are made:
    - If team A gets the correct answer, the automaton follows the transition labeled “A” and goes to state **2-0** i.e. the score of team A is 2 and team B is 0.
    - If team B gets the correct answer, the automaton follows the transition labeled “B” and goes to state **0-2** i.e. the score of team A is 0 and team B is 2.
    - If the answer was wrong, the scores are still **0-0** so the automaton follows the transition labeled **n** that loops back on the state **0-0.**

3. From the states **2-0** and **0-2**, considering the 3 outputs for each (as stated in #6 in the explanation part), the automaton has transitions to states **4-0**, **2-2, 2-0** and **2-2, 0-4, 0-2** respectively. All the other transitions are designed in the same way.
4. There are two final states: **Team A Wins** and **Team B Wins.**

Processing an Input Sequence/ String

1. The alphabet of the automaton is the set {A, B, n}
2. An example string: ABnnBAnBAA
3. Following the transitions labeled by the characters of the input string in order, the automaton moves through the sequence of states starting from 0-0:

   2-0, 2-2, 2-2, 2-2, 2-4, 4-4, 4-4, 4-6, 6-6, Team A Wins

4. Final output: After processing an input sequence of characters, the automaton reached the final state **Team A Wins**. Therefore, the input sequence is "valid". It is accepted by the automaton.
