Finite Automaton Variants

Mohamed M. El Wakil
mohamed@elwakil.net
Agenda

• Finite State Automaton/Machine (FSA)
  – State transition tables
  – Non-deterministic Finite State Automaton (NFA)
  – Two-way Deterministic Finite Automaton (2FA)

• Finite State Transducer
  – Mealy machine
  – Moore machine
Finite State Automaton/Machine

• It is a model of computation that consists of:
  – States
    • Start/Initial state
    • Acceptance states
    • Other states
  – Transitions
    • Moving from a state to another
  – Input string
    • Finite set of symbols
  – Actions
    • Entry action
    • Exit action
    • Transition action
    • Input action
Recognition/Acceptance

• A FSA is said to *recognize*, or *accept* an input string, if it ends at an acceptance state, and the whole input string has been processed.
Example

• This machine has 7 states, 8 transitions, and 0 actions.
• State “Start” is the start state.
• If the input is the word “nice”, this machine will end at the state “Success”, otherwise it will end at the “Error” state.
Transitions

- A transition is a move from a state A to another state B based on the current state (i.e. A), and the current symbol of the input string.

- Transitions can be viewed as a function with two inputs (current state, and current input symbol), and one output (next state). For example:
  - (S1, “n”) \rightarrow S2
  - (S1, “not_n”) \rightarrow S6

- When a symbol is input to a transition function, it is said to be consumed.

- Symbols in the input string are consumed from left to right.
State Transition Table

- Transition functions are expressed via state transition tables.
- State transition tables can be either one dimension or two dimensions.

- This machine accepts
  - 1111000, 1010, and 000
- but, does not accept
  - 1001, 111, or 001
State Transition Tables

• One dimension table:

<table>
<thead>
<tr>
<th>Input</th>
<th>Current State</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>P</td>
<td>Q</td>
</tr>
<tr>
<td>0</td>
<td>Q</td>
<td>R</td>
</tr>
<tr>
<td>0</td>
<td>R</td>
<td>S</td>
</tr>
<tr>
<td>1</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>1</td>
<td>Q</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>R</td>
<td>-</td>
</tr>
</tbody>
</table>

• Two dimension tables:

<table>
<thead>
<tr>
<th>Next Current</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Q</td>
<td>-</td>
<td>-</td>
<td>{0,1}</td>
<td>-</td>
</tr>
<tr>
<td>R</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input State</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Q</td>
<td>P</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
<td>-</td>
</tr>
</tbody>
</table>
Mathematically speaking

• A finite state machine is quintuple \( (\Sigma, S, s_0, \delta, F) \):
  
  - \( \Sigma \): Finite non-empty set of symbols (Input).
  - \( S \): Finite, non-empty set of states (States).
  - \( S_0 \): Initial state \( S_0 \) belongs to \( S \) (Start state).
  - \( F \): Set of final states, a subset of \( S \).
  - \( \delta \): Transition function (Transitions)
    - \( \delta : S \times \Sigma \rightarrow S \)
Non-deterministic Finite State Machine (NFA)

• The FSA is deterministic, If
  – for each pair of state and input symbol there is one and only one transition to a next state.

• The FSA is non-deterministic, if
  – it is allowed for a pair of state and input symbol to have more than one possible next state, or a transition can take place without consuming input symbol (ε transitions)
NFA Example 1

- This machine is non-deterministic as for the pair (P,0), there is more than one possible next state (P, and Q)

```
<table>
<thead>
<tr>
<th>Input State</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>P,Q</td>
<td>P</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
<td>-</td>
</tr>
</tbody>
</table>
```
NFA Example 2

• This machine is non-deterministic it can move from state Q to state R, without consuming any input.

<table>
<thead>
<tr>
<th>Input State</th>
<th>0</th>
<th>1</th>
<th>ε</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Q</td>
<td>P</td>
<td>-</td>
</tr>
<tr>
<td>Q</td>
<td>-</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Deterministic Vs. Non-deterministic Finite State Automaton/Machine

• NFA
  – Easier to design
  – Harder to implement
  – Slower
  – Smaller

• DFA
  – Easier to implement
  – Harder to design
  – Faster
  – Bigger
Two-way Finite Automaton

• In DFA, input can be read only once from left to right.

• In 2DFA, input can be read back and forth with no limit on how many times an input symbol can be read.

• A 2DFA is a generalization of the DFA.

• Transition functions in 2DFA, outputs a new state, and a direction (left, right, or stand still)
DFA Vs. 2DFA

• 2DFA can solve any problem solvable by DFA.
  – Why?

• Also, problems solvable by a 2DFA, can be solved by a DFA\(^1\).

• But, 2DFA is
  – much simpler, and
  – more realistic.

Finite State Transducers (FST)

- A FSA can be viewed as a function that maps an input string into the set \{0,1\}, \{True, False\}, \{Accept, Reject\}...

- A FST may generate a new string (output string) while consuming the input string.

- A FST is a FSA with actions producing output!
FST Example

• This machine outputs the one’s complement of its input, while it is recognizing this input.

• For input 000, it outputs 111
• For input 1010, it outputs 0101
FST variants

• A FST, may be non-deterministic, if it can produce more than one output string for a given input string.

• A FST is either a Mealy machine, or a Moore machine.
Mealy Machines

• A Mealy machine is finite state transducer that produces an output for each transition (i.e. has input actions)

• Example:
Moore Machines

• A Moore machine is finite state transducer that produces an output for each state it enters (i.e. has entry actions)

• Example:
  – We have to add a new state X
Mathematically speaking

• A finite state transducer is a sextuple ($\Sigma, \Gamma, S, S_0, \delta, \omega$):
  – $\Sigma$: Finite non-empty set of symbols (Input).
  – $\Gamma$: Finite non-empty set of symbols (Output).
  – $S$: Finite, non-empty set of states (States).
  – $S_0$: Initial state $S_0$ belongs to $S$ (Start state).
  – $\delta$: Transition function (Transitions)
    • $\delta: S \times \Sigma \to S$
  – $\omega$: Output function
    • $\omega: S \times \Sigma \to \Gamma$ (Mealy machine)
    • $\omega: S \to \Gamma$ (Moore machine)