CMPSC230
Lesson 3: Finite State Machines

Prof. John Wenskovitch
09/02/2014
Last Time

• Proofs
  – Terminology
  – Strategies
  – Proof by Construction
  – Proof by Contradiction
  – Proof by Induction
State

• “A unique configuration of information in a program or machine.”
• “A snapshot of the measure of various conditions in a system.”
• “All of the stored information at a given instant in time to which a circuit or program has access.”
• “The mode of operation of a computer during the execution of an instruction.”
Toy Example: Traffic Light

• How can we model a traffic light’s states?
• Now let’s add a left-turn arrow.
• How about two directions?
• How about a four-way intersection?

(gets complicated fast, right?)
Finite Automaton

• A finite automaton is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\), where:
  
  – \(Q\) is a finite set called the **states**
  
  – \(\Sigma\) is a finite set called the **alphabet**
  
  – \(\delta: Q \times \Sigma \rightarrow Q\) is the **transition function**
  
  – \(q_0 \in Q\) is the **start state**
  
  – \(F \subseteq Q\) is the set of **accept states**
Example 1

Graph: A finite state machine with states $q_1$, $q_2$, and $q_3$. The transitions are:
- $q_1$ to $q_2$ on 0
- $q_2$ to $q_1$ on 1
- $q_2$ to $q_3$ on 1
- $q_3$ to $q_2$ on 0,1
- $q_3$ to $q_3$ on 1,0
Example 1

- states: \( Q = \{q_1, q_2, q_3\} \)
- alphabet: \( \Sigma = \{0, 1\} \)
- transition function: \( \delta: Q \times \Sigma \rightarrow Q \)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_1 )</td>
<td>( q_1 )</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>( q_3 )</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>( q_3 )</td>
<td>( q_2 )</td>
<td>( q_2 )</td>
</tr>
</tbody>
</table>

- start state: \( q_1 \)
- accept states: \( F = \{q_2\} \)
- all together: \( M = (\{q_1, q_2, q_3\}, \{0, 1\}, \delta, q_1, \{q_2\}) \)
Example 2

\[ q_1 \xrightarrow{0} q_2 \quad q_2 \xrightarrow{0} q_1 \]

\[ q_2 \xrightarrow{1} q_1 \quad q_1 \xrightarrow{1} q_2 \]
Example 2

– states: \( Q = \{q_1, q_2\} \)

– alphabet: \( \Sigma = \{0, 1\} \)

– transition function: \( \delta: Q \times \Sigma \rightarrow Q \)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_1 )</td>
<td>( q_1 )</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>( q_1 )</td>
<td>( q_2 )</td>
</tr>
</tbody>
</table>

– start state: \( q_1 \)

– accept states: \( F = \{q_2\} \)

– all together: \( M_2 = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_2\}) \)
What Strings Will $M_2$ Accept?

<table>
<thead>
<tr>
<th>Accepts 1</th>
<th>Rejects 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accepts 11</td>
<td>Rejects 00</td>
</tr>
<tr>
<td>Accepts 01</td>
<td>Rejects 10</td>
</tr>
<tr>
<td>Accepts 1101</td>
<td>Rejects 0010</td>
</tr>
<tr>
<td>Accepts 1001001101</td>
<td>Rejects 0110110010</td>
</tr>
</tbody>
</table>

All end with 1

- Accepts 1
- Accepts 11
- Accepts 01
- Accepts 1101
- Accepts 1001001101
What Strings Will $M_2$ Accept?

- $M_2$ accepts every string that ends with a 1.
- $L(M_2) = \{w | w \text{ ends in a 1}\}$
- $L(M) = A$

Read as “$w$ such that $w$ ends in a 1”

Read as “the language of machine M is A” or “M recognizes A” or “M accepts A”
Example 3

Finite State Machines
Example 3

- states: $Q = \{s, q_1, q_2, r_1, r_2\}$
- alphabet: $\Sigma = \{a, b\}$
- transition function: $\delta: Q \times \Sigma \rightarrow Q$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>$q_1$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$r_1$</td>
<td>$r_2$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>$r_2$</td>
<td>$r_2$</td>
<td>$r_1$</td>
</tr>
</tbody>
</table>

- start state: $s$
- accept states: $F = \{q_1, r_1\}$
What Strings Will $M_3$ Accept?

- Accepts $a$
- Accepts $b$
- Accepts $ababa$
- Accepts $babab$
- Accepts $bbabaabbab$

- Rejects $ab$
- Rejects $ba$
- Rejects $ababab$
- Rejects $bababa$
- Rejects $bbabaabba$

**Same letter begin and end**

**Different letter**
What Strings Will $M_3$ Accept?

- $M_3$ accepts every string that begins and ends with the same letter
- $L(M_3) = \{w \mid w \text{ begins and ends with the same letter}\}$

Language function of $M_3$  Language of $M_3$  Read as “$w$ such that $w$ begins and ends with the same letter”
Example 4

• Design a machine that will accept only strings that contain an odd number of 1s. ($\Sigma = \{0, 1\}$)
  – The empty string has an even number of 1s, so don’t start in an accept state.
  – Every time we read a 0, our decision to accept is \textbf{NOT} affected.
  – Every time we read a 1, our decision to accept \textbf{IS} affected.
Example 4

- Design a machine that will accept only strings that contain an odd number of 1s. \( \Sigma = \{0, 1\} \)
Example 4

- Design a machine that will accept only strings that contain an odd number of 1s. ($\Sigma = \{0, 1\}$)
  - states: $Q = \{q_{even}, q_{odd}\}$
  - alphabet: $\Sigma = \{0, 1\}$
  - transition function: $\delta: Q \times \Sigma \rightarrow Q$
    
    |       | 0     | 1     |
    |-------|-------|-------|
    | $q_{even}$ | $q_{even}$ | $q_{odd}$ |
    | $q_{odd}$  | $q_{odd}$  | $q_{even}$ |
  - start state: $q_{even}$
  - accept states: $F = \{q_{odd}\}$
Any Questions?

HOMEWORK (due 9/9)
1.1, 1.2, 1.3, 1.4acef, 1.6ab

Note: DFA = “Deterministic Finite Automata”...
it’s a finite state machine