FSM highlights

Finite state machines
- States, transitions, start state, final states
- Languages recognized by FSMs

FSM that accepts strings of a’s, b’s, c’s with no more than 3 a’s

FSM that accepts binary strings with a 1 three positions from the start
FSM that accepts binary strings with a 1 three positions from the end

Strings over \{0, 1, 2\}*

\(M_1\): Strings with an even number of 2’s

\(M_2\): Strings where the sum of digits mod 3 is 0

Strings with an even number of 2’s and a mod 3 sum of 0

3 bit shift register  “Remember the last three bits”
State Machines with Output

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_1)</td>
<td>(s_1)</td>
<td>(s_2)</td>
</tr>
<tr>
<td>(s_2)</td>
<td>(s_1)</td>
<td>(s_3)</td>
</tr>
<tr>
<td>(s_3)</td>
<td>(s_2)</td>
<td>(s_4)</td>
</tr>
<tr>
<td>(s_4)</td>
<td>(s_3)</td>
<td>(s_4)</td>
</tr>
</tbody>
</table>

“Tug-of-war”

Vending Machine

Enter 15 cents in dimes or nickels
Press S or B for a candy bar
**Vending Machine, v0.1**

Basic transitions on N (nickel), D (dime), B (butterfinger), S (snickers)

**Vending Machine, v0.2**

Adding output to states: N – Nickel, S – Snickers, B – Butterfinger

**Vending Machine, v1.0**

Adding additional “unexpected” transitions