Lecture 22: Finite State Machines
About the Midterm

We will release midterm grades at the end of the day today.

Preliminary information:
Median: 81  Average: 75  Standard Deviation: 19

Grade Distribution:
90+   25%
80’s  29%
70’s  14%
60’s  11%
50’s  9%
40’s  7%
<40   5%
Last class: Strings this machine says are OK?

The set of all binary strings that end in 0
Finite State Machines

- States
- Transitions on input symbols
- Start state and final states
- The “language recognized” by the machine is the set of strings that reach a final state from the start

<table>
<thead>
<tr>
<th>Old State</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>$s_0$</td>
<td>$s_1$</td>
</tr>
<tr>
<td>$s_1$</td>
<td>$s_0$</td>
<td>$s_2$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>$s_0$</td>
<td>$s_3$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>$s_3$</td>
<td>$s_3$</td>
</tr>
</tbody>
</table>
Finite State Machines

- Each machine designed for strings over some fixed alphabet $\Sigma$.
- Must have a transition defined from each state for every symbol in $\Sigma$.

<table>
<thead>
<tr>
<th>Old State</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>$s_0$</td>
<td>$s_1$</td>
</tr>
<tr>
<td>$s_1$</td>
<td>$s_0$</td>
<td>$s_2$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>$s_0$</td>
<td>$s_3$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>$s_3$</td>
<td>$s_3$</td>
</tr>
</tbody>
</table>
What language does this machine recognize?

Binary strings
contain 111
or end in 0
or ε

<table>
<thead>
<tr>
<th>Old State</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>$s_0$</td>
<td>$s_1$</td>
</tr>
<tr>
<td>$s_1$</td>
<td>$s_0$</td>
<td>$s_2$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>$s_0$</td>
<td>$s_3$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>$s_3$</td>
<td>$s_3$</td>
</tr>
</tbody>
</table>
What language does this machine recognize?

The set of all binary strings that contain **111** or don’t end in 1

<table>
<thead>
<tr>
<th>Old State</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>$s_0$</td>
<td>$s_1$</td>
</tr>
<tr>
<td>$s_1$</td>
<td>$s_0$</td>
<td>$s_2$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>$s_0$</td>
<td>$s_3$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>$s_3$</td>
<td>$s_3$</td>
</tr>
</tbody>
</table>
Strings over \{0, 1, 2\}

\(M_1\): Strings with an even number of 2’s
Strings over \{0, 1, 2\}

M₁: Strings with an even number of 2’s
State Machine Design Recipe

Given a language, how do you design a state machine for it?

Create states to remember enough
(about the portion of the input string that it has already seen)
to correctly answer “accept/reject” on the whole string after seeing the rest.

Add labeled edges to show how the memory (state) should be updated for each new symbol.
Strings over \{0, 1, 2\}

\(M_2\): Strings where the sum of digits mod 3 is 0
Strings over \( \{0, 1, 2\} \)

\( M_2 \): Strings where the sum of digits mod 3 is 0
Strings over \{0, 1, 2\}

\(M_2: \) Strings where the sum of digits mod 3 is 0
What language does this machine recognize?

The machine is a deterministic finite automaton (DFA) with states $s_0$, $s_1$, $s_2$, and $s_3$. The transitions are as follows:

- From $s_0$: 1 to $s_1$, 0 to $s_2$.
- From $s_1$: 0 to $s_0$, 1 to $s_3$.
- From $s_2$: 1 to $s_3$, 0 to $s_2$.
- From $s_3$: 1 to $s_2$, 0 to $s_0$.

The input symbols are 0 and 1. The machine accepts the following languages:

- Good strings: 6, 0, 00, 01, 11
- Bad strings: 1, 11, 10, 101, 000

The machine will stay in an even state of bits.
What language does this machine recognize?

The set of all binary strings with \# of 1's \equiv \# of 0's (mod 2) (both are even or both are odd).

Can you think of a simpler description?
Strings over \(\{0, 1, 2\}\)

M₁: Strings with an even number of 2’s

M₂: Strings where the sum of digits mod 3 is 0
Strings over \{0,1,2\} w/ even number of 2’s and mod 3 sum 0
Strings over \(\{0,1,2\}\) w/ even number of 2’s and mod 3 sum 0
Strings over \{0, 1, 2\} w/ even number of 2’s OR mod 3 sum 0
The set of binary strings with a 1 in the $3^{rd}$ position from the start
The set of binary strings with a 1 in the 3\textsuperscript{rd} position from the start
The set of binary strings with a 1 in the 3rd position from the end
3 bit shift register  “Remember the last three bits”
The set of binary strings with a 1 in the 3rd position from the end.
The set of binary strings with a 1 in the 3\textsuperscript{rd} position from the end
The beginning versus the end
Adding Output to Finite State Machines

• So far, we have considered finite state machines that just accept/reject strings
  – called “Deterministic Finite Automata” or DFAs

• Now we consider finite state machines with output
  – These are the kinds used as controllers
Vending Machine

Enter 15 cents in dimes or nickels
Press S or B for a candy bar