review: finite state machines

- States
- Transitions on inputs
- Start state and final states
- The language recognized by a machine is the set of strings that reach a final state

<table>
<thead>
<tr>
<th>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>

applications of FSMs (aka finite automata)

- Implementation of regular expression matching in programs like `grep`
- Control structures for sequential logic in digital circuits
- Algorithms for communication and cache-coherence protocols
  - Each agent runs its own FSM
- Design specifications for reactive systems
  - Components are communicating FSMs

applications of FSMs (aka finite automata)

- Formal verification of systems
  - Is an unsafe state reachable?
- Computer games
  - FSMs provide worlds to explore
  - Character AI
- Minimization algorithms for FSMs can be extended to more general models used in
  - Text prediction
  - Speech recognition

waka waka

tcp
what language does this machine recognize?

![Finite State Machine Diagram]

can we recognize these languages with DFAs?

- $\emptyset$
- $\Sigma^*$
- $\{ x \in \{0,1\}^* : \text{len}(x) > 1 \}$

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

both: even number of 2's and sum mod 3 = 0

DFA that accepts strings of a's, b's, c's with no more than 3 a's
3 bit shift register

```
Remember the last three bits
```

```

start and accept states
```

```

FSMs with output
```

```

"Tug-of-war"
```

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
vending machine
```

```
We're only making $5.50/hour
writing regular expressions.
Let's design a vending machine.
```

```
"He does not think like normal people, and as a result his tests are quite difficult. His lectures are amusing and get the material across, but his office hours are not always too helpful. Beware the vending machine final."
```

```
Vending spec:
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)
```
Adding output to states: N – Nickel, S – Snickers, B – Butterfinger

Adding additional "unexpected" transitions