

Computational Models and Finite State Machines



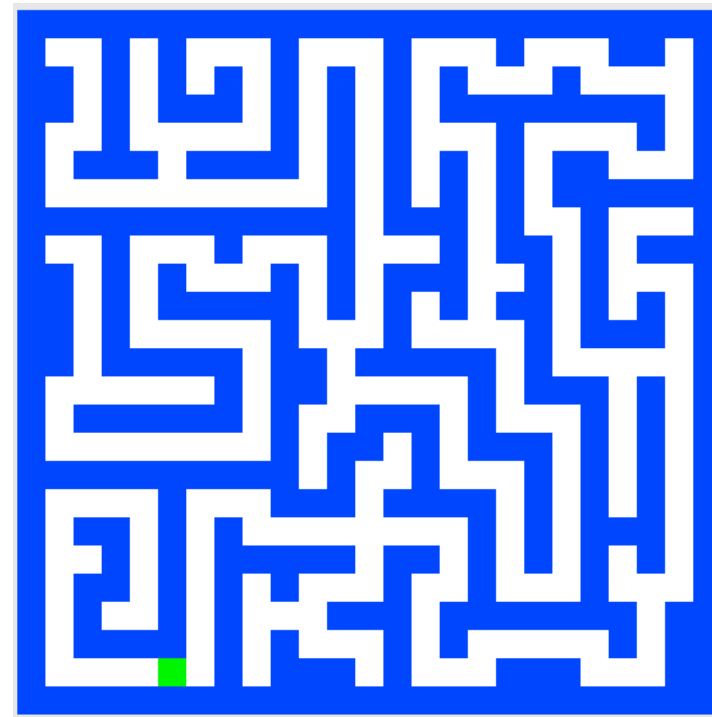
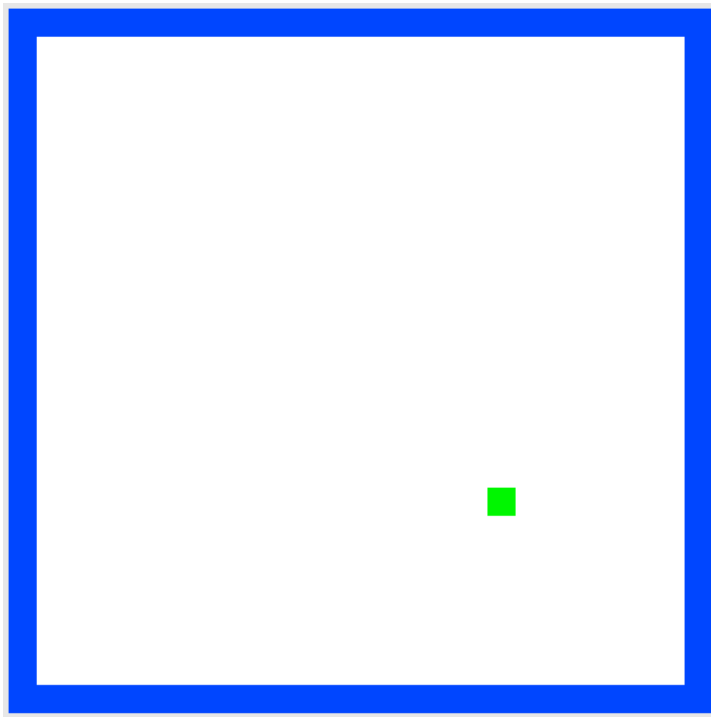
CS 100: Introduction to the Profession
Michael Saelee; saelee@iit.edu

Debate!

Recently, GoDaddy and CloudFlare terminated accounts related to a white supremacist website called The Daily Stormer. Later, the CloudFlare called the decision “dangerous” and that it set a precedent for censorship online.

CloudFlare serves 10% of all internet traffic. Should they have the power to shut down websites unanimously? Should they have shut down The Daily Stormer?





How to best go about coming up with a solution? (States, Patterns, Rules, etc.)



Observation: immediately sitting down to write code (sometimes: “hacking”) is not typically the best approach!



It can help to have a *computational model* that helps us reason about the problem.



A Finite State Machine is a computational model consisting of:

- a finite number of states
- an initial state
- conditions (input/output/actions) for transitions between states

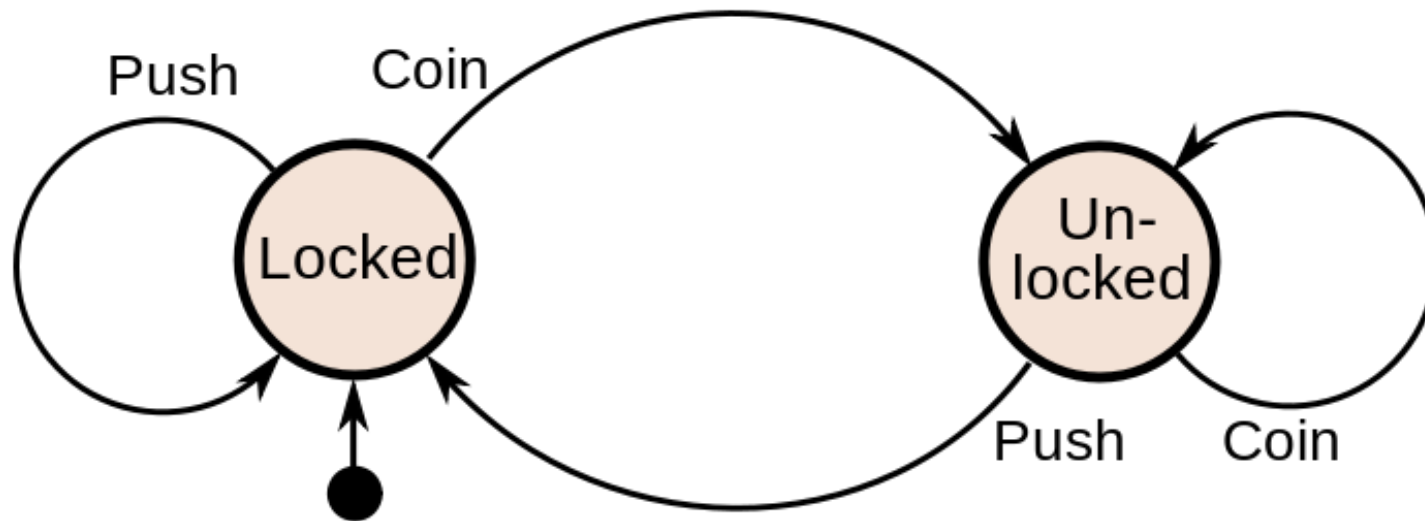


A FSM is a a sort of *abstract machine*, whose behavior we can easily implement using programming languages, microcontrollers (for robots), etc.



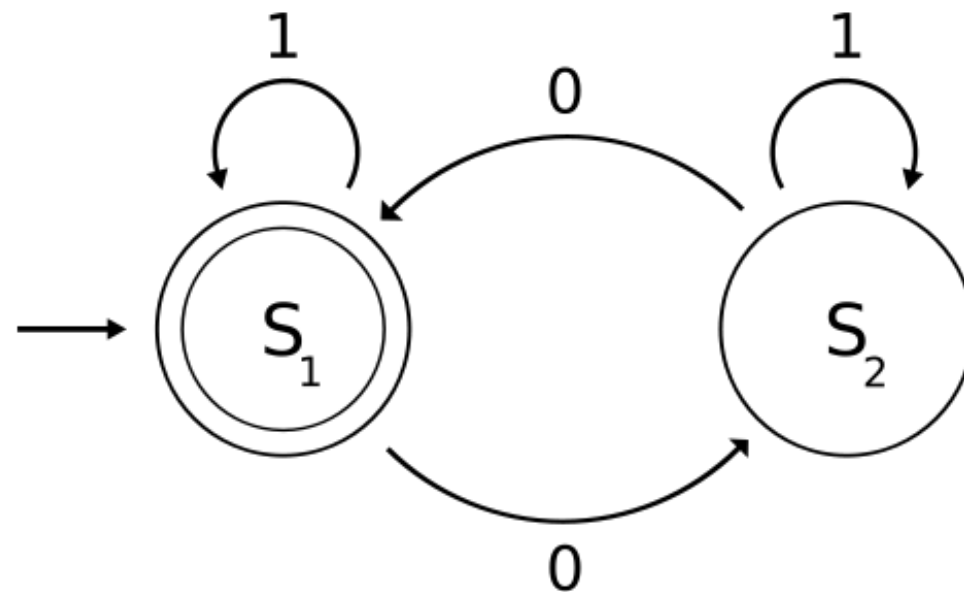
We often use *state transition diagrams* to detail the behavior of a FSM.

E.g., for a turnstile (courtesy Wikipedia):



FSMs can implement “acceptors”, which only accept a sequence of input when it terminates on one or more *accepting states*

What input does this FSM accept?



Can you design a FSM that accepts:

- Either “CAT” or “DOG”?
- A (possibly floating point) number?
- An e-mail address?



FSMs may also perform *output* or *actions* on each state transition.

Can we use this to reason about Picobot?

