cse 311: foundations of computing

Fall 2015

Lecture 22: Finite state machines

$(\,) \, ((\,))$

G: $S \to SS \mid (S) \mid \varepsilon$

- $\forall s$ if s has matched () $\to$ G produces s.

$P(n) =$ "$\forall s$ $len(s) = n$, if s has () $\to$ G prod. s."

Base Case: $P(0)$ $\quad s = \varepsilon \quad S \to \varepsilon$

IH: For some $n \geq 0$ and any $0 \leq j \leq n$, $P(j)$ holds

IS: Goal $P(n+1)$. Fix T with matched ().

$$(\underbrace{\_\ \_\ \_\ \_}_{k}) \underbrace{\_\ \_\ \_\ \_}_{n+1-k-2}$$

$S \to SS \to (S)S$

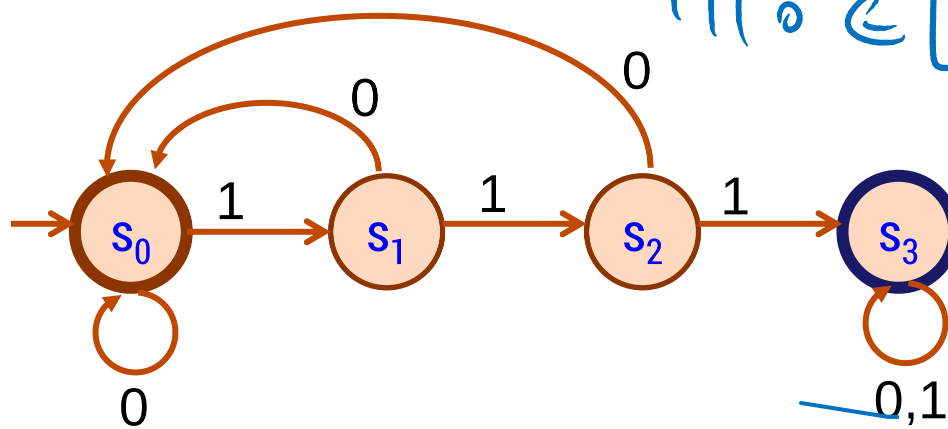
review: finite state machines

- 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

$1001 \notin L$

$01110 \in L$

State	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_3
s_3	s_3	s_3



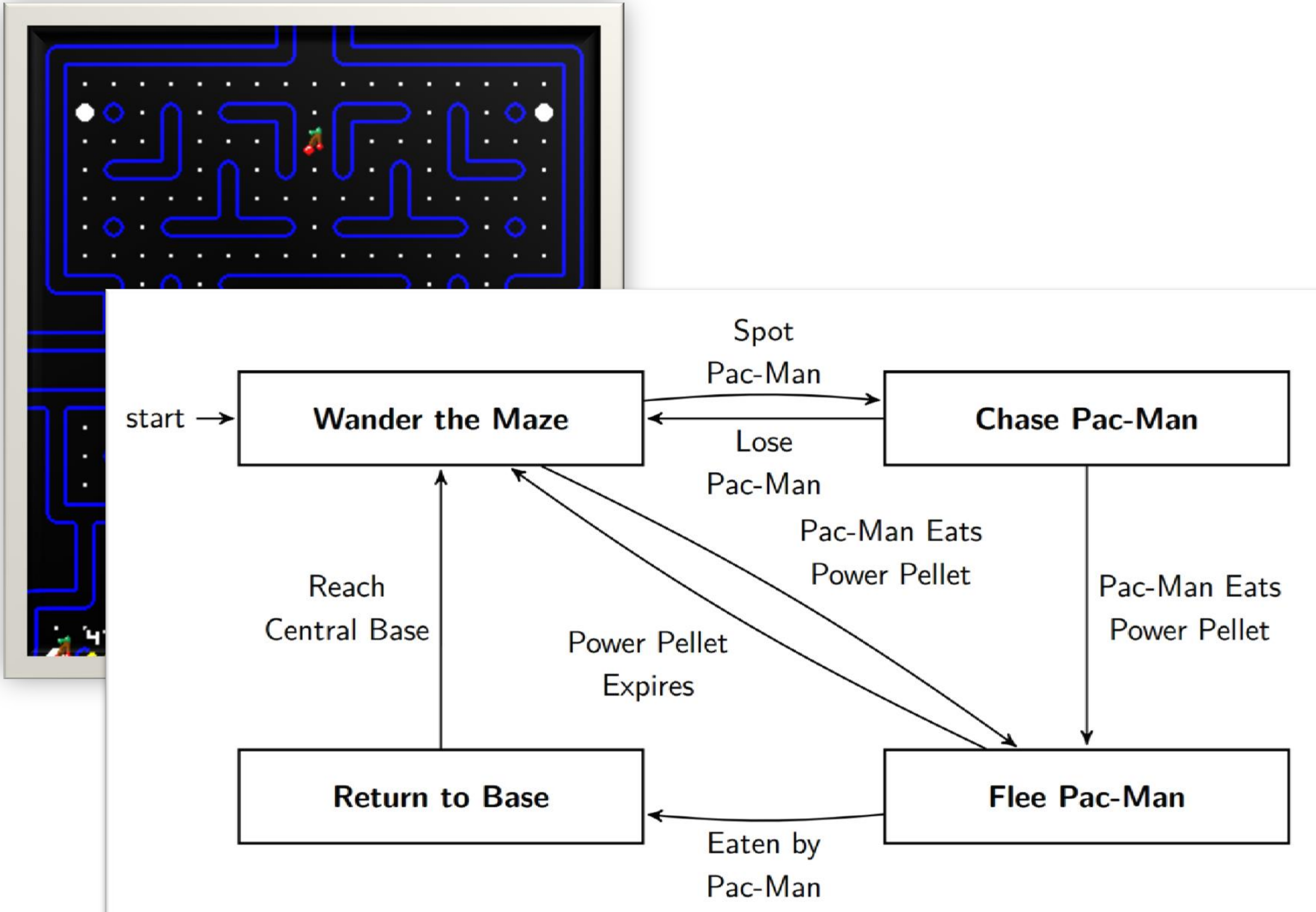
applications of FSMs (aka finite automata)

- 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)

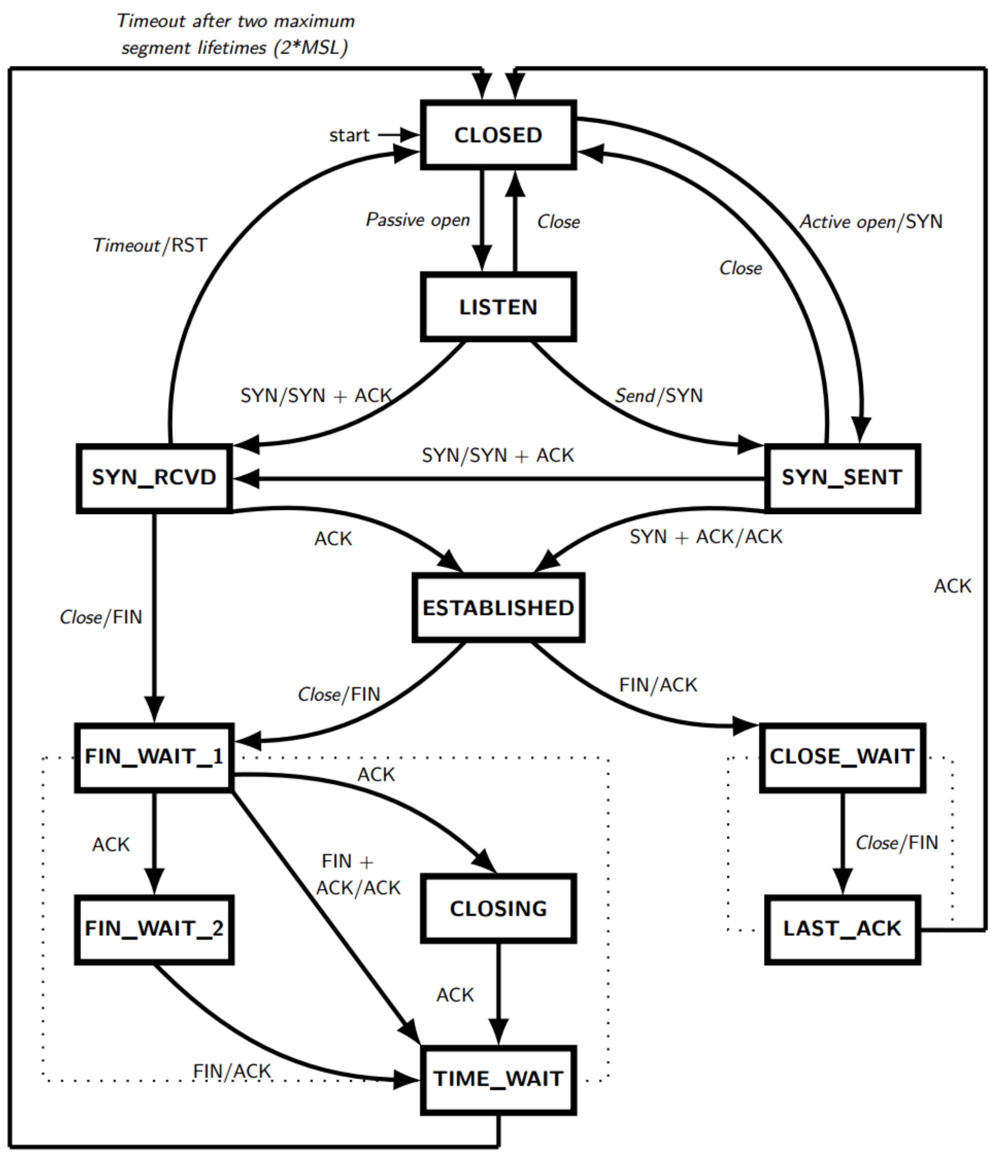
- Formal verification of systems
  - Is an unsafe state reachable?
- Computer games
  - FSMs provide worlds to explore
  - Character AI
- Minimization algorithms for FSMs can be extended to more general models used in
  - Text prediction
  - Speech recognition

waka waka



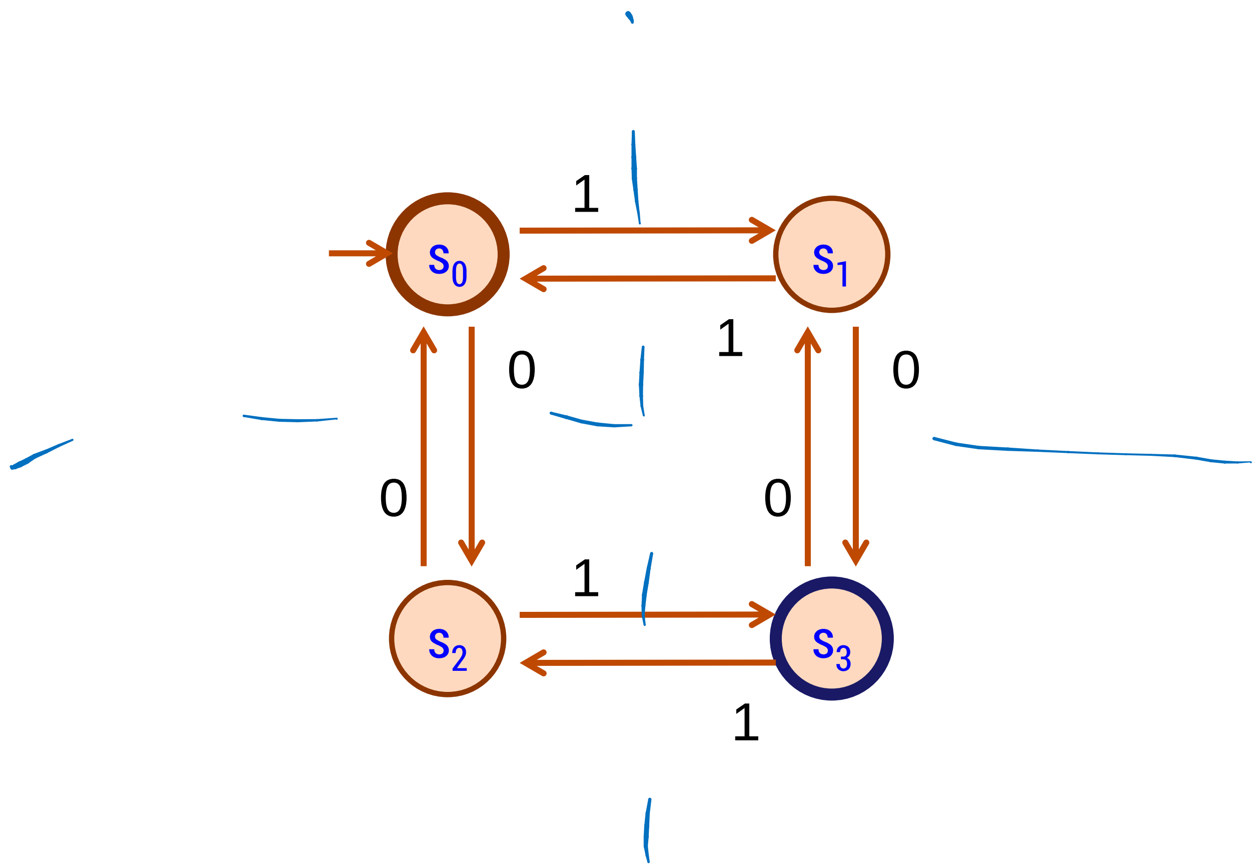
start
Wander the Maze
Spot
Pac-Man
Chase Pac-Man
Lose
Pac-Man
Pac-Man Eats
Power Pellet
Reach
Central Base
Power Pellet
Expires
Pac-Man Eats
Power Pellet
Return to Base
Flee Pac-Man
Eaten by
Pac-Man

tcp



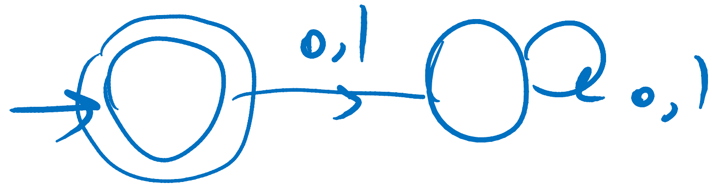
Timeout after two maximum segment lifetimes (2*MSL)
start
CLOSED
Passive open
Close
Timeout/RST
Active open/SYN
Close
LISTEN
SYN/SYN + ACK
Send/SYN
SYN_RCVD
SYN/SYN + ACK
SYN_SENT
ACK
SYN + ACK/ACK
ACK
ESTABLISHED
Close/FIN
Close/FIN
FIN/ACK
FIN_WAIT_1
ACK
CLOSE_WAIT
ACK
FIN + ACK/ACK
Close/FIN
CLOSING
FIN_WAIT_2
LAST_ACK
ACK
FIN/ACK
TIME_WAIT

what language does this machine recognize?

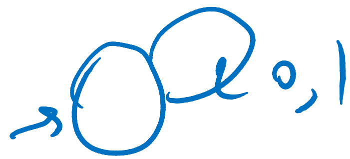


can we recognize these languages with DFAs?

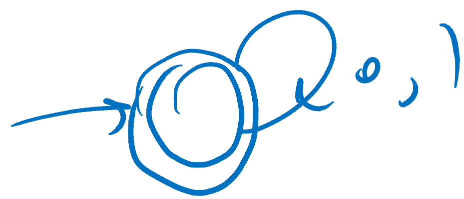
- ε



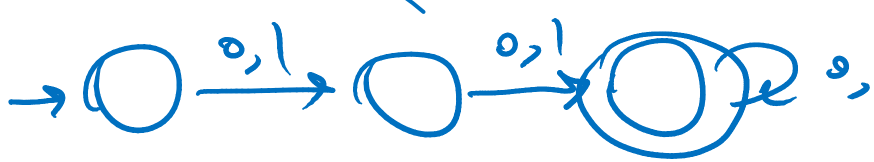
- ∅



- Σ^*

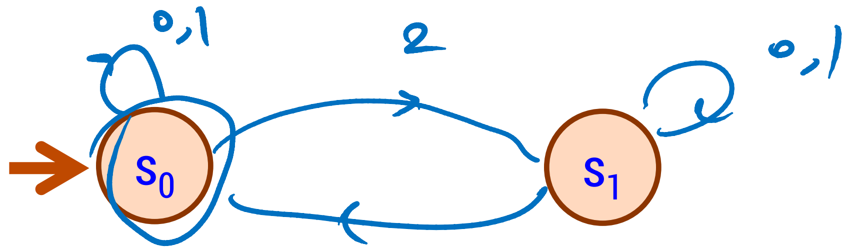


- $\{ x \in \{0,1\}^* : \text{len}(x) > 1\}$

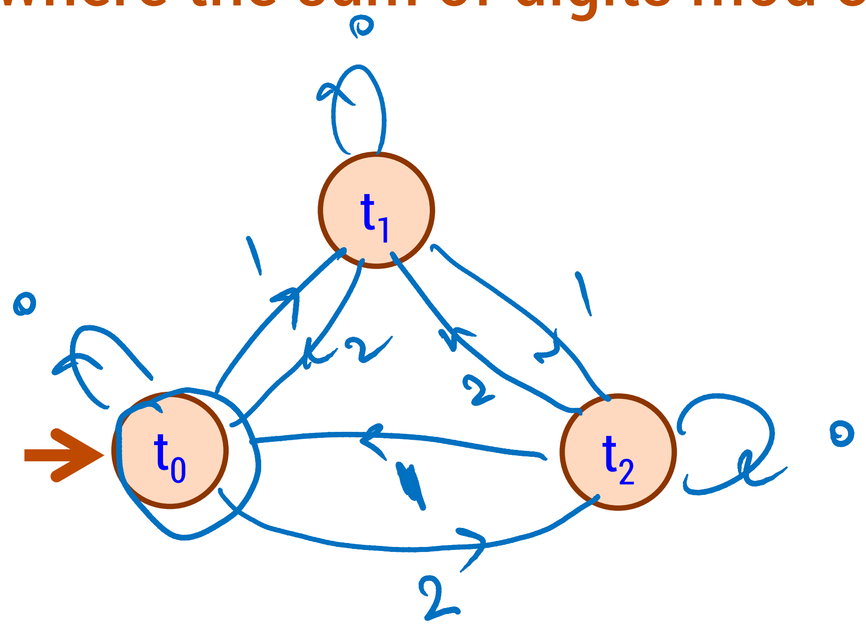


strings over {0, 1, 2}*

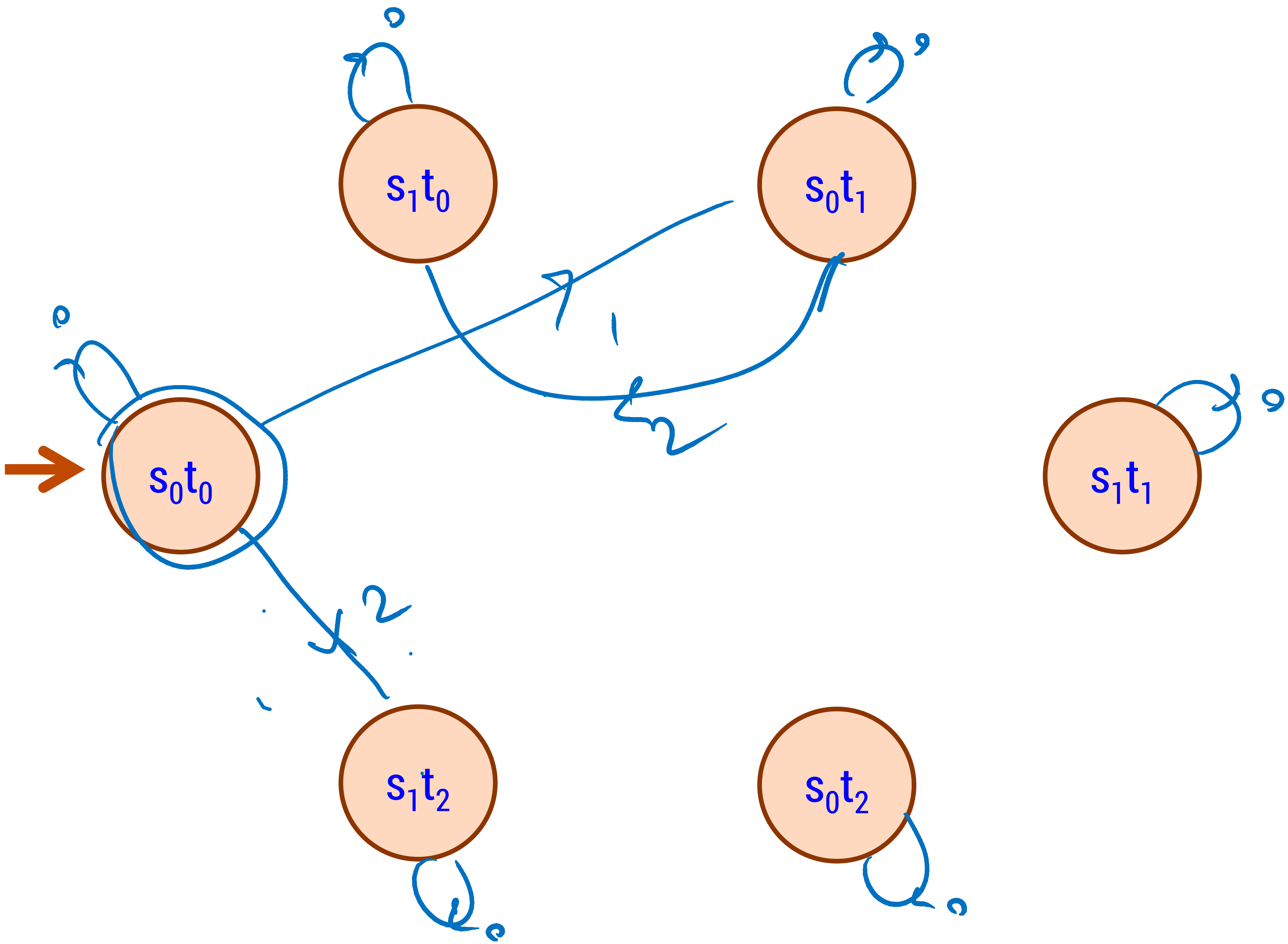
M_1: Strings with an even number of 2's



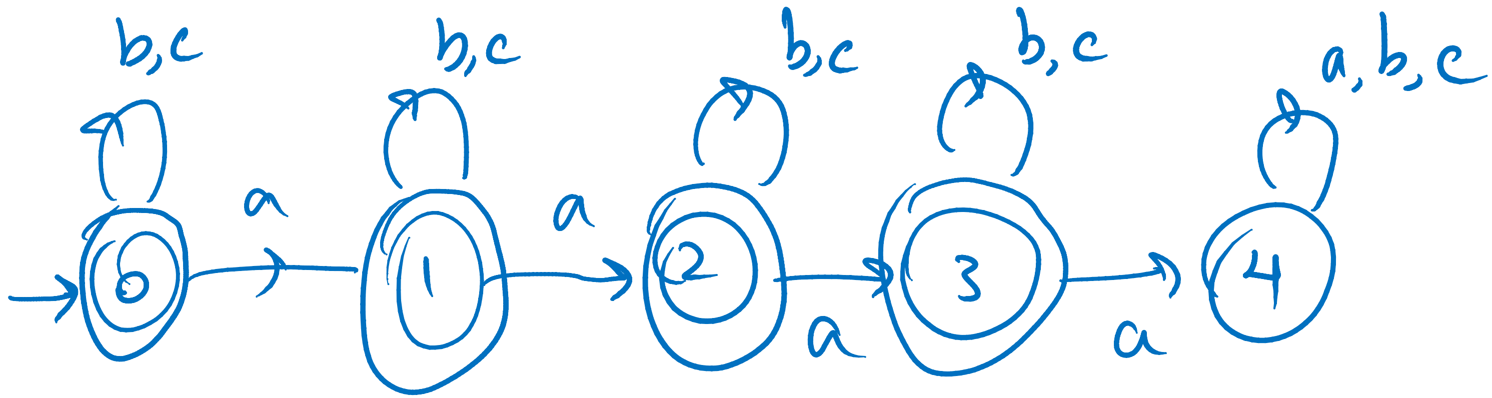
M_2: Strings where the sum of digits mod 3 is 0



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



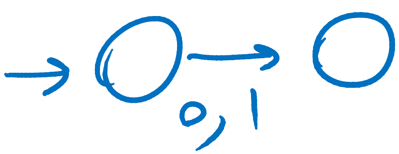
FSM that accepts binary strings with a 1 three positions from the end

$0010 \notin L$

$0100 \in L$

$00001 \notin L$

$1110 \in L$



111

$1111 \in L$

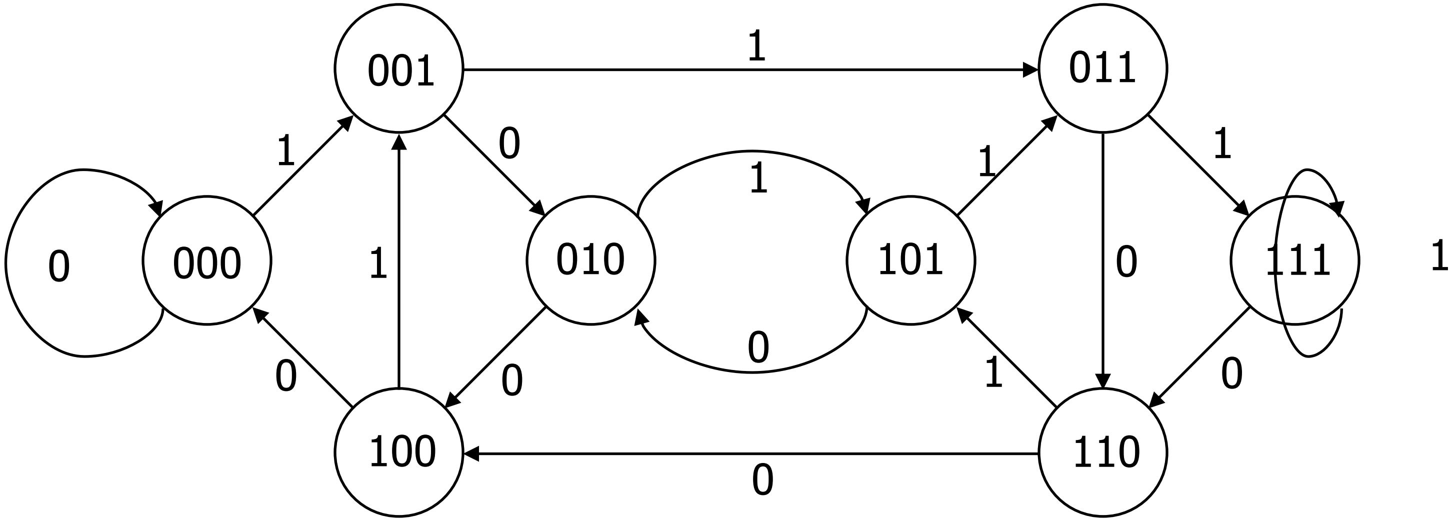
000

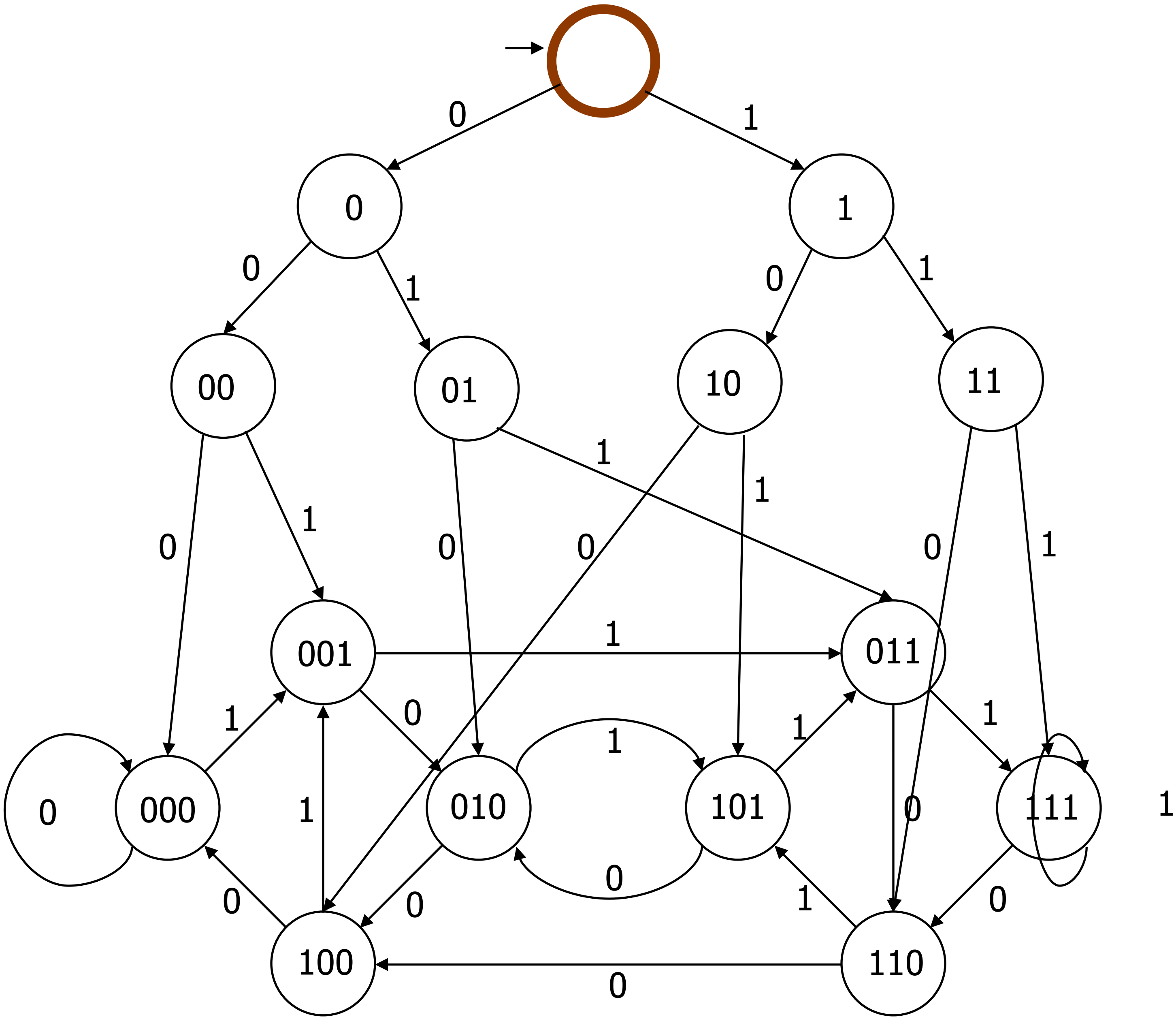
100

1

“Remember the last three bits”

3 bit shift register



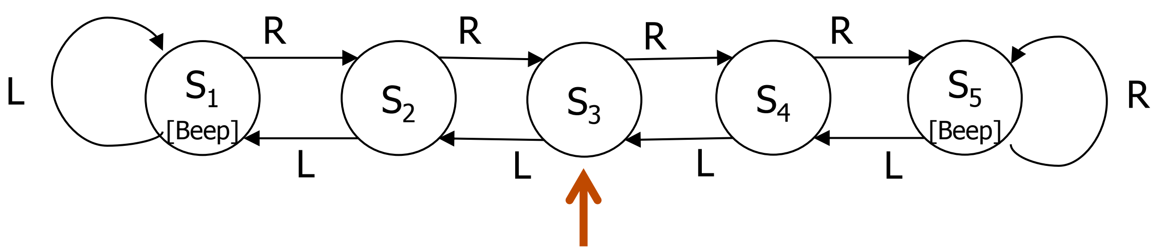


0
1
00
01
10
11
001
011
000
010
101
111
100
110

FSMs with output

“Tug-of-war”

	Input		Output
State	L	R	
s_1	s_1	s_2	Beep
s_2	s_1	s_3	
s_3	s_2	s_4	
s_4	s_3	s_5	
s_5	s_4	s_5	Beep



vending machine

We're only making $5.50/hour writing regular expressions.

Let's design a vending machine.

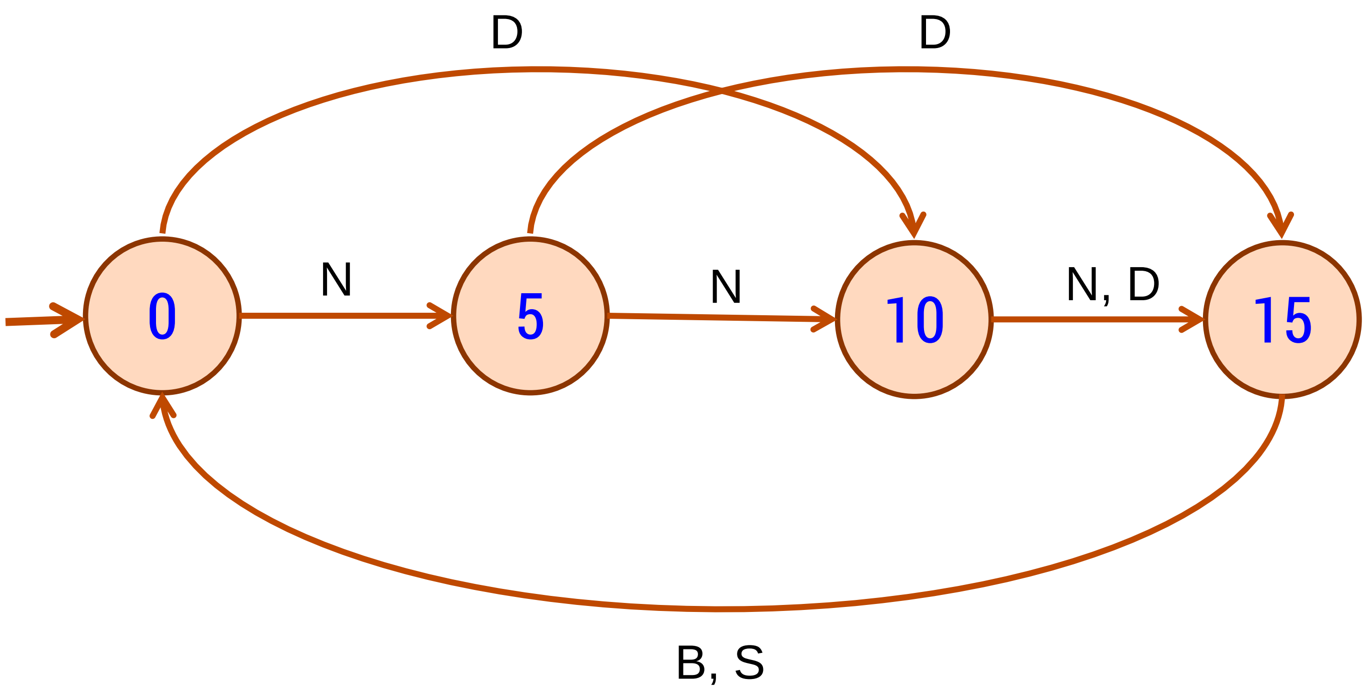
"He does not think like normal people, and as a result his tests are quite difficult. His lectures are amusing and get the material across, but his office hours are not always too helpful. **Beware the vending machine final.**"

Vending spec:

Enter 15 cents in dimes or nickels

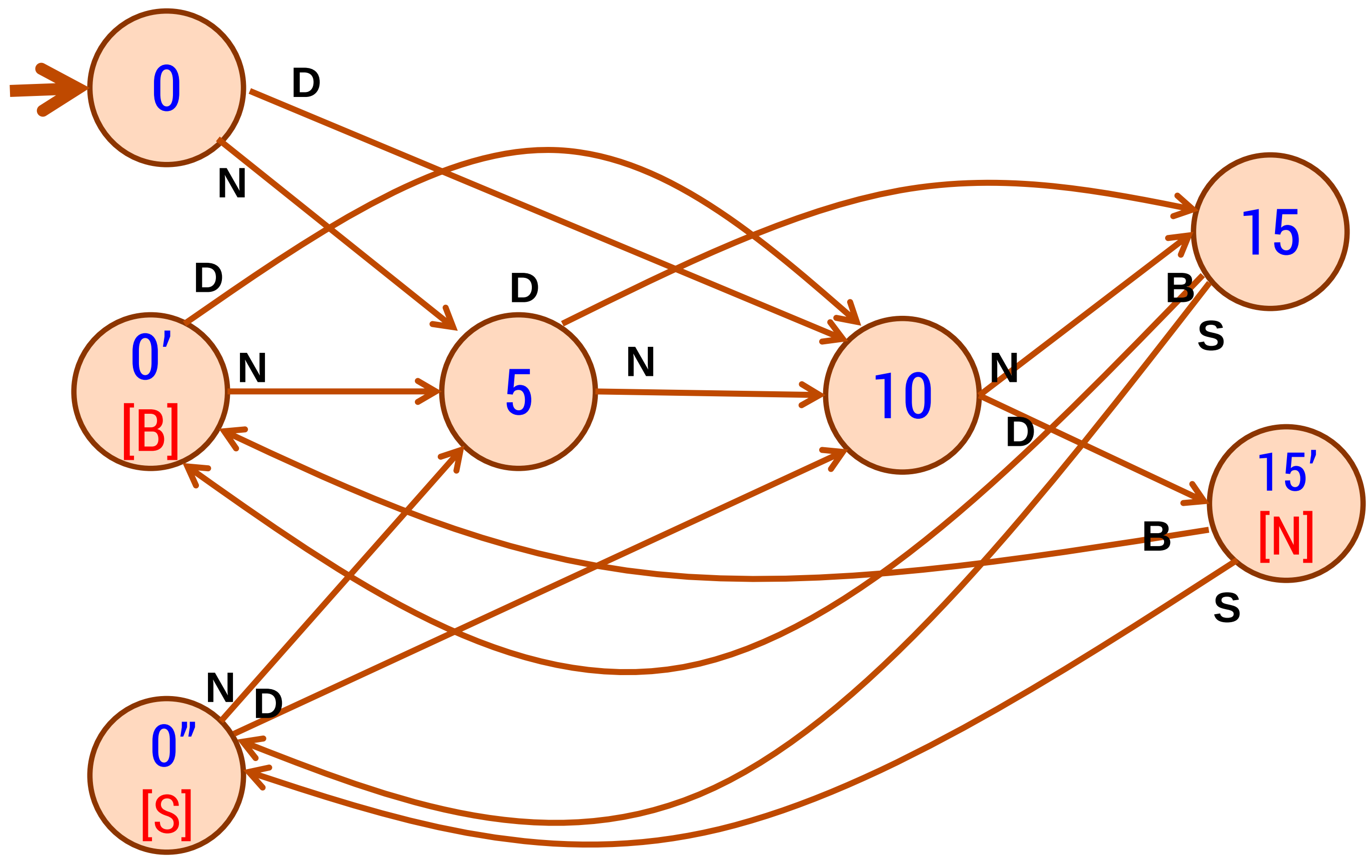
Press **S** or **B** for a candy bar

vending machine v0.1



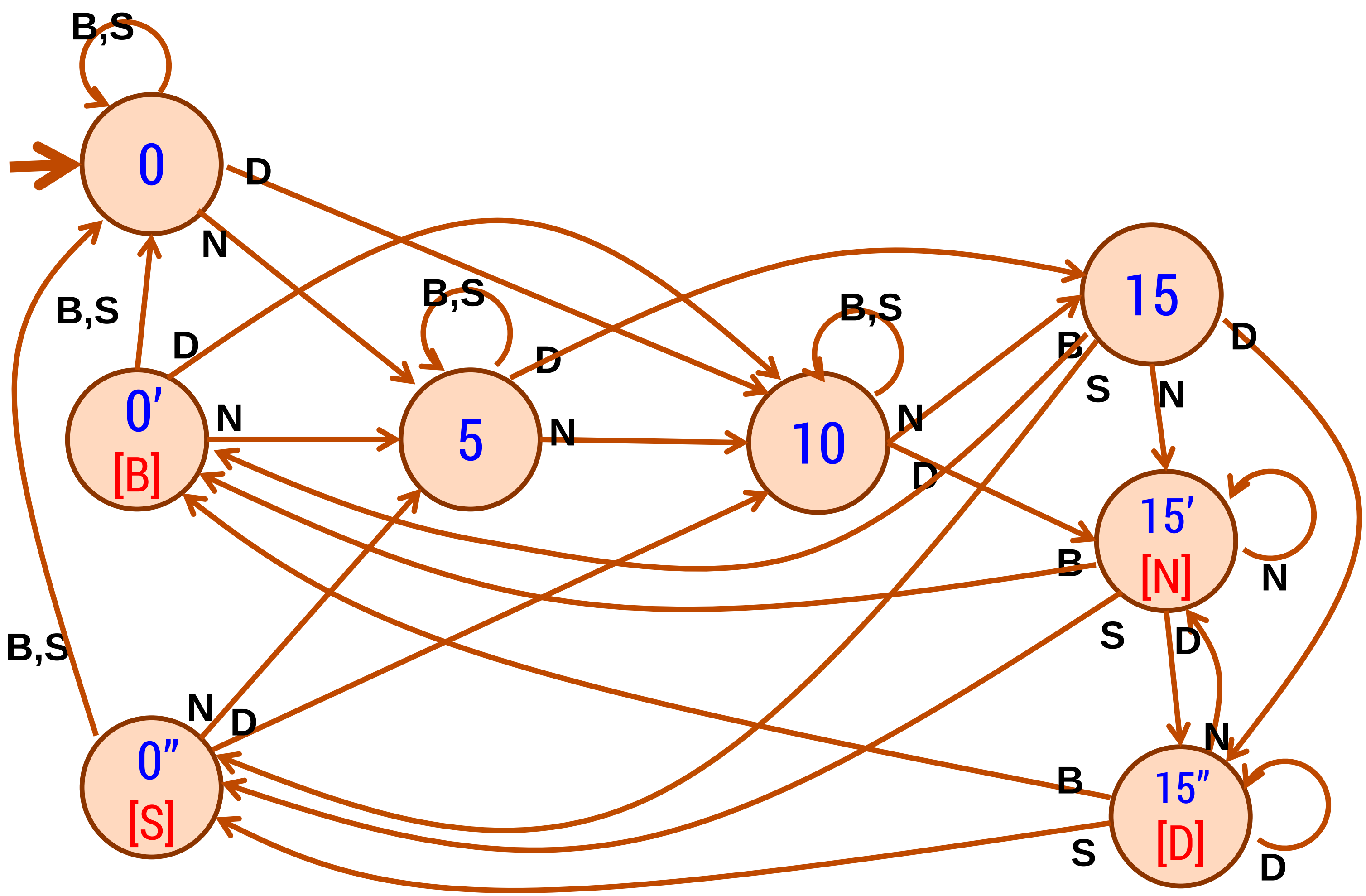
Basic transitions on N (nickel), D (dime), B (butterfinger), S (snickers)

vending machine v0.2



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

vending machine v1.0



Adding additional “unexpected” transitions