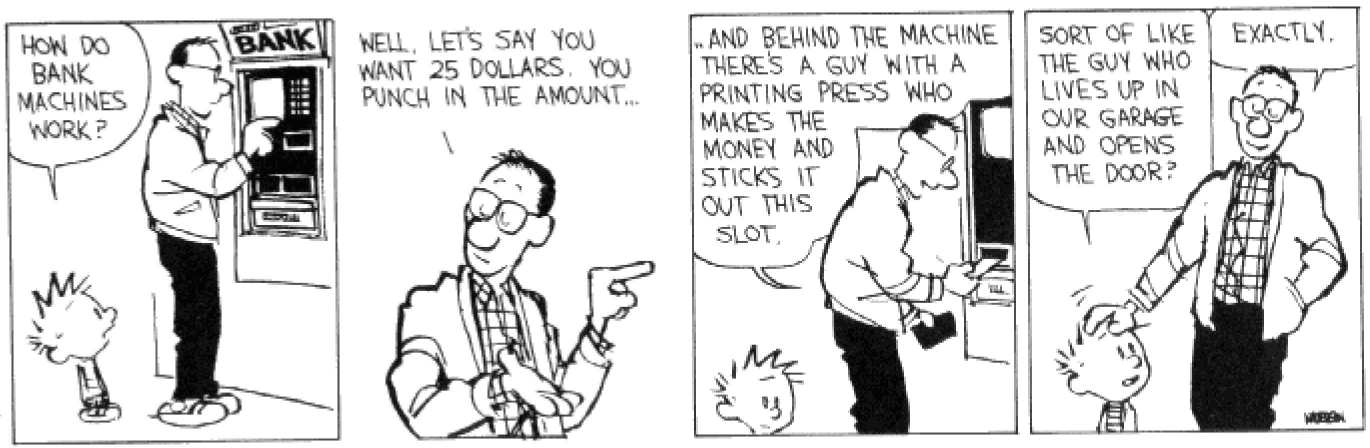
CSE 311: Foundations of Computing

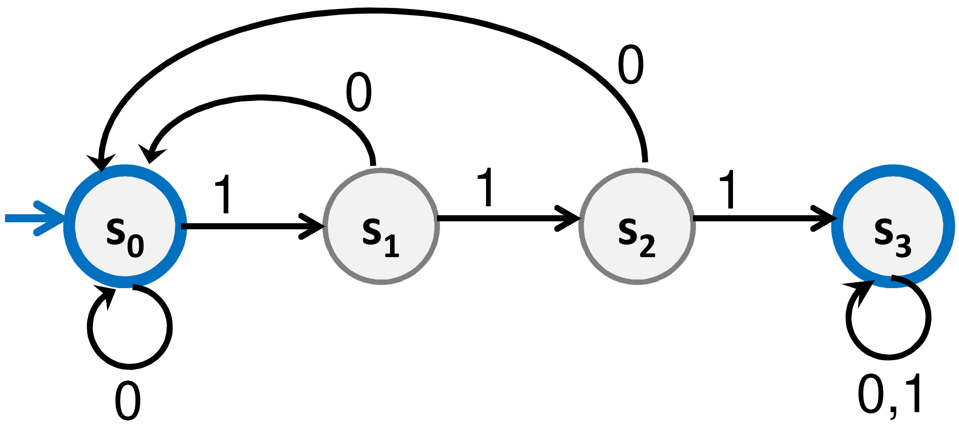
Lecture 21: DFAs and Finite State Machines with Output



Finite State Machines (DFA)

- 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

Old 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

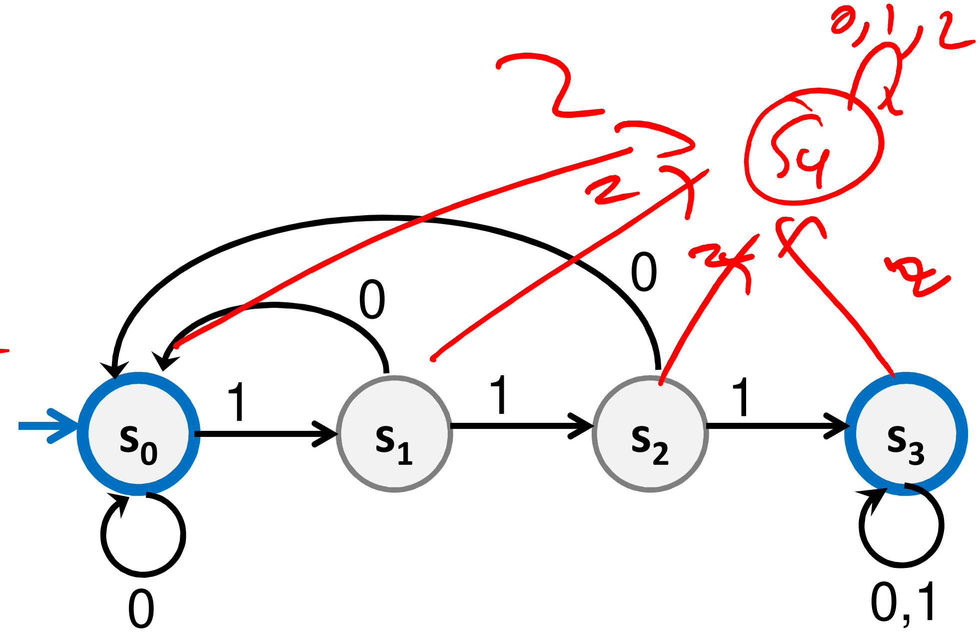


Finite State Machines

- **Each machine designed for strings over some fixed alphabet Σ.**

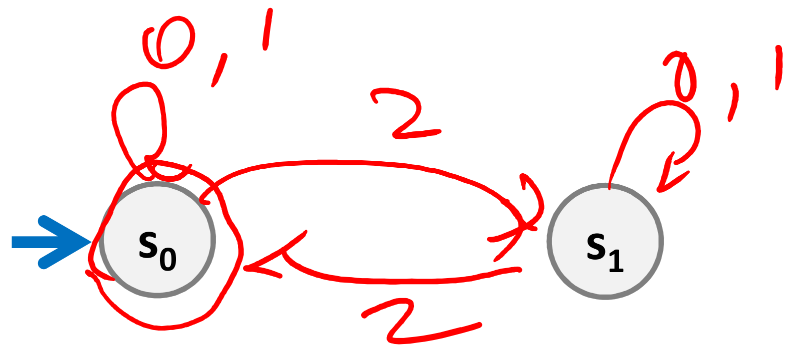
- **Must have a transition defined from each state for *every* symbol in Σ.**

Old 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

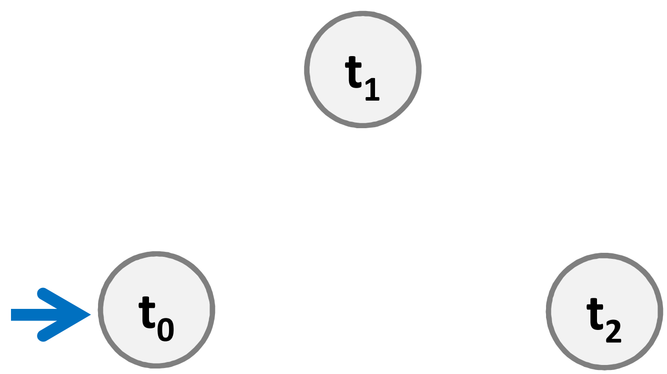


Strings over {0, 1, 2}

M_1: Strings with an even number of 2's

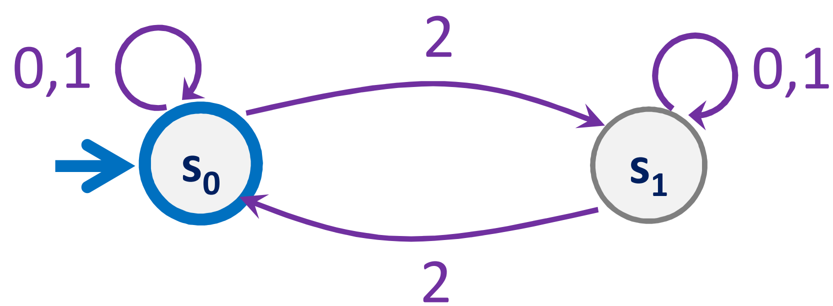


M_2: Strings where the sum of digits mod 3 is 0

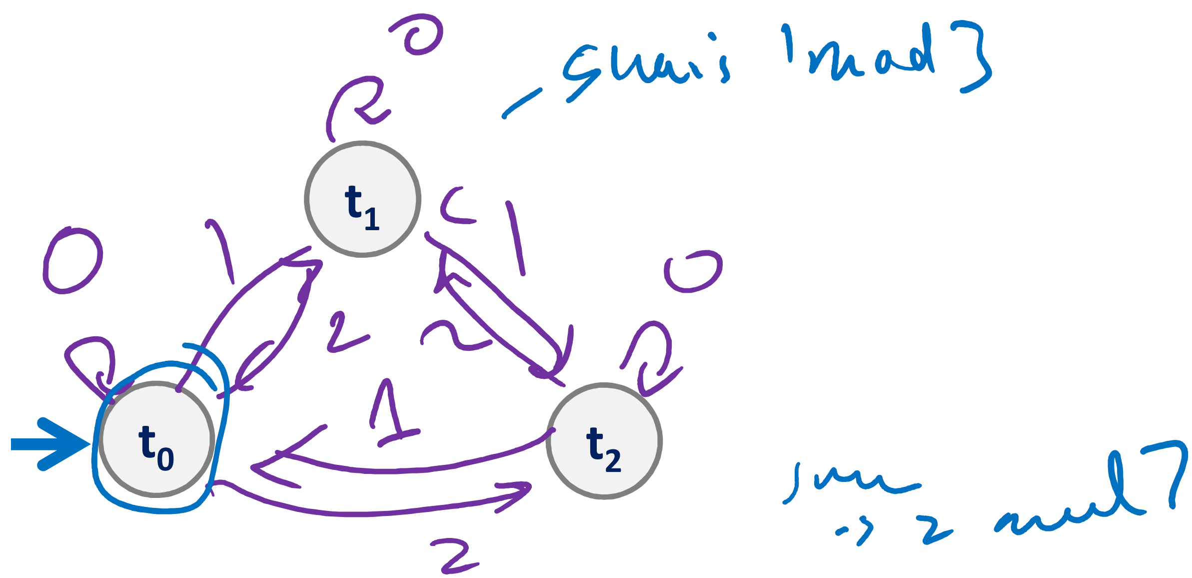


Strings over {0, 1, 2}

M_1: Strings with an even number of 2's

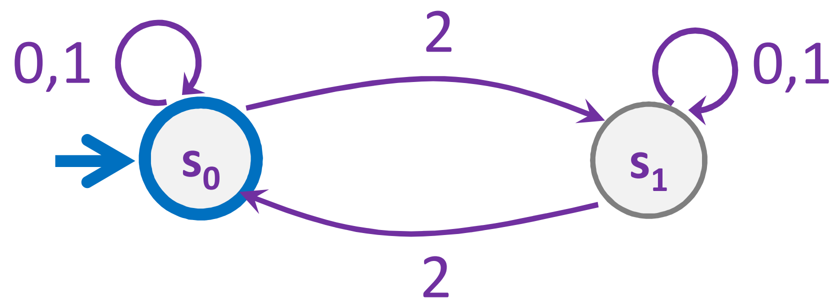


M_2: Strings where the sum of digits mod 3 is 0

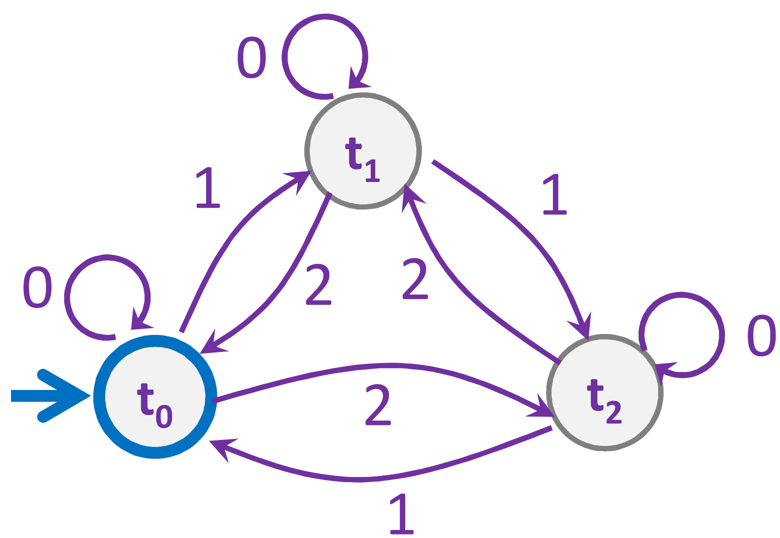


Strings over {0, 1, 2}

M_1: Strings with an even number of 2's

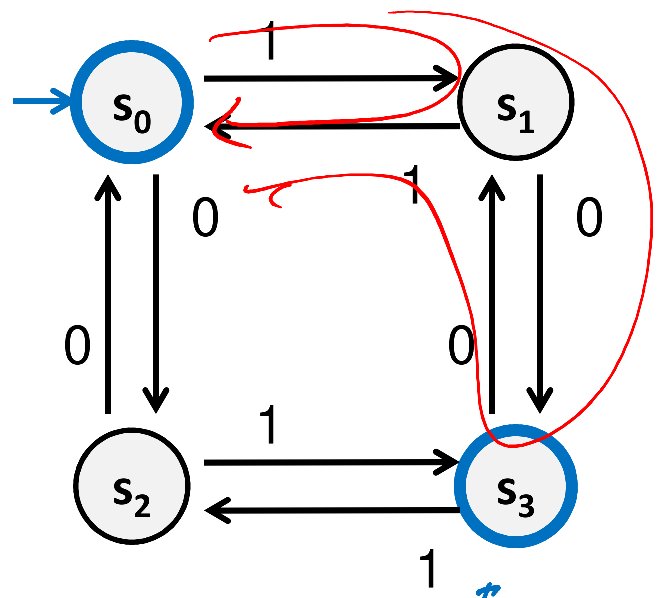


M_2: Strings where the sum of digits mod 3 is 0



What language does this machine recognize?

s_0 = even # of 1's and even # of 0's



s_3 = odd # of 1's and odd # of 0's

~~Alternate 1's and 0's all 1's for all 0's~~

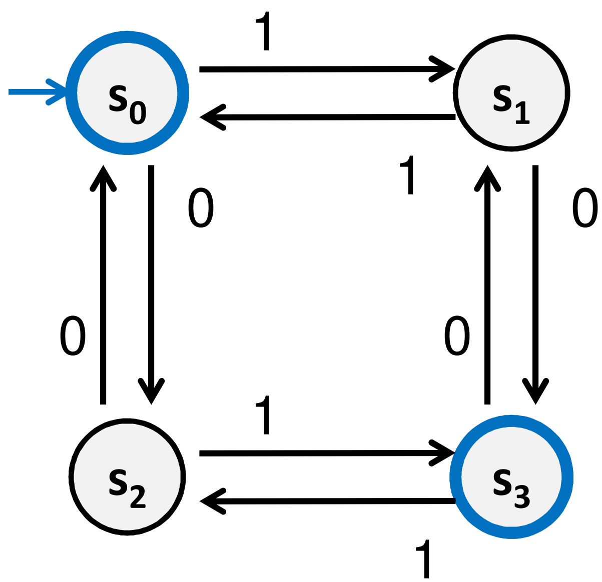
~~String 1st char is differ from last~~

Strings with even # of chars ✓

~~equal # 1's and 0's~~

parity of # 1's = parity of # of 0's

What language does this machine recognize?

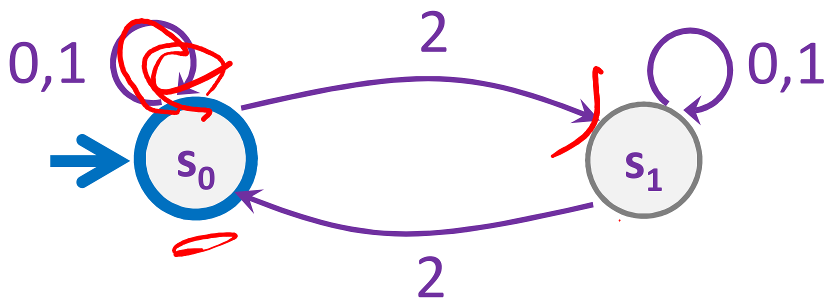


The set of all binary strings with # of 1's ≡ # of 0's (mod 2) (both are even or both are odd).

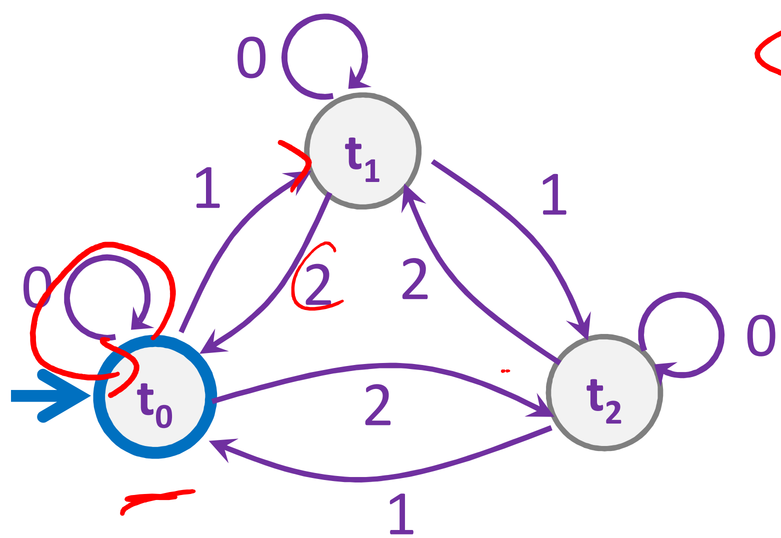
Can you think of a simpler description?

Strings over {0, 1, 2}

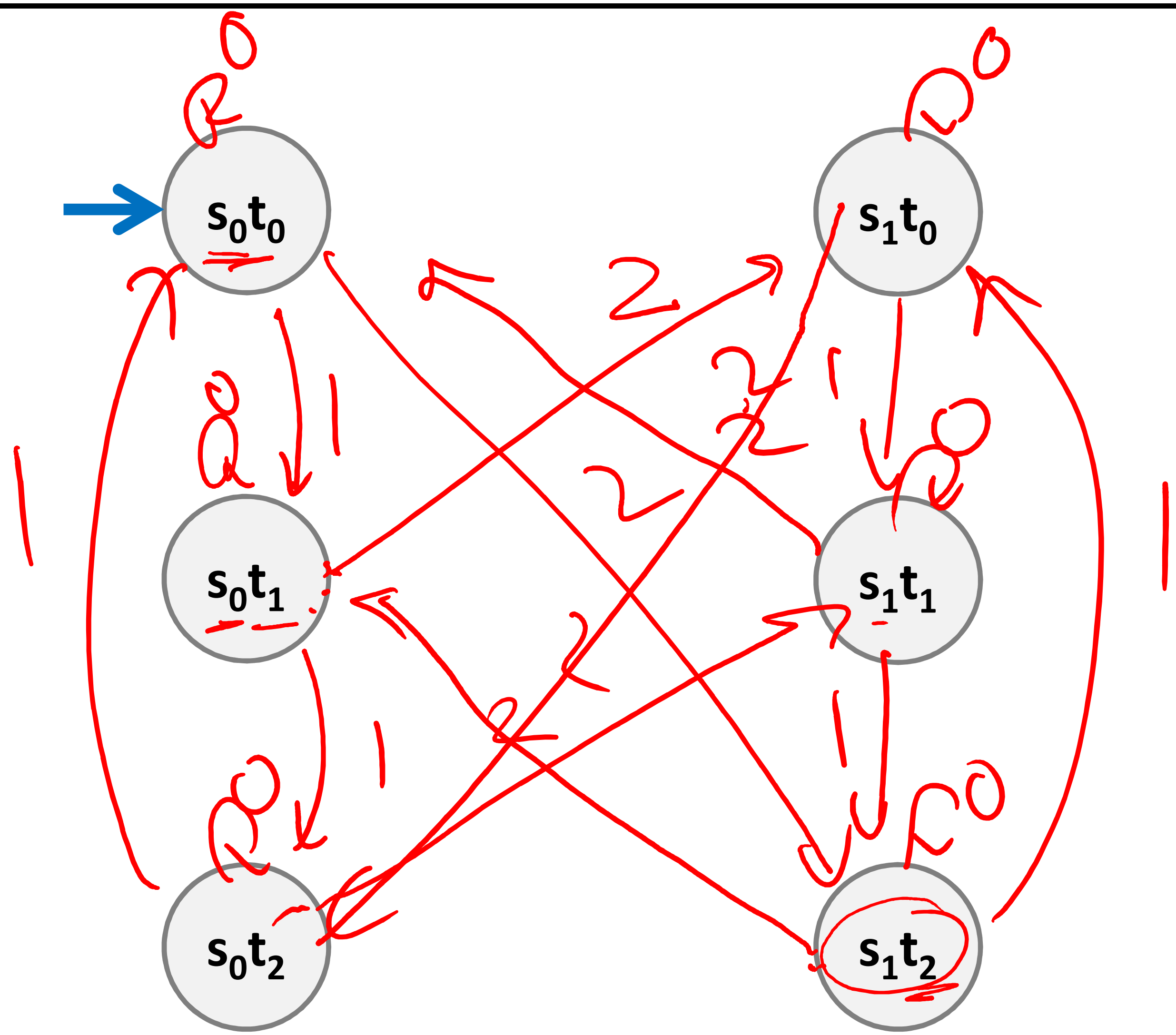
M_1: Strings with an even number of 2's



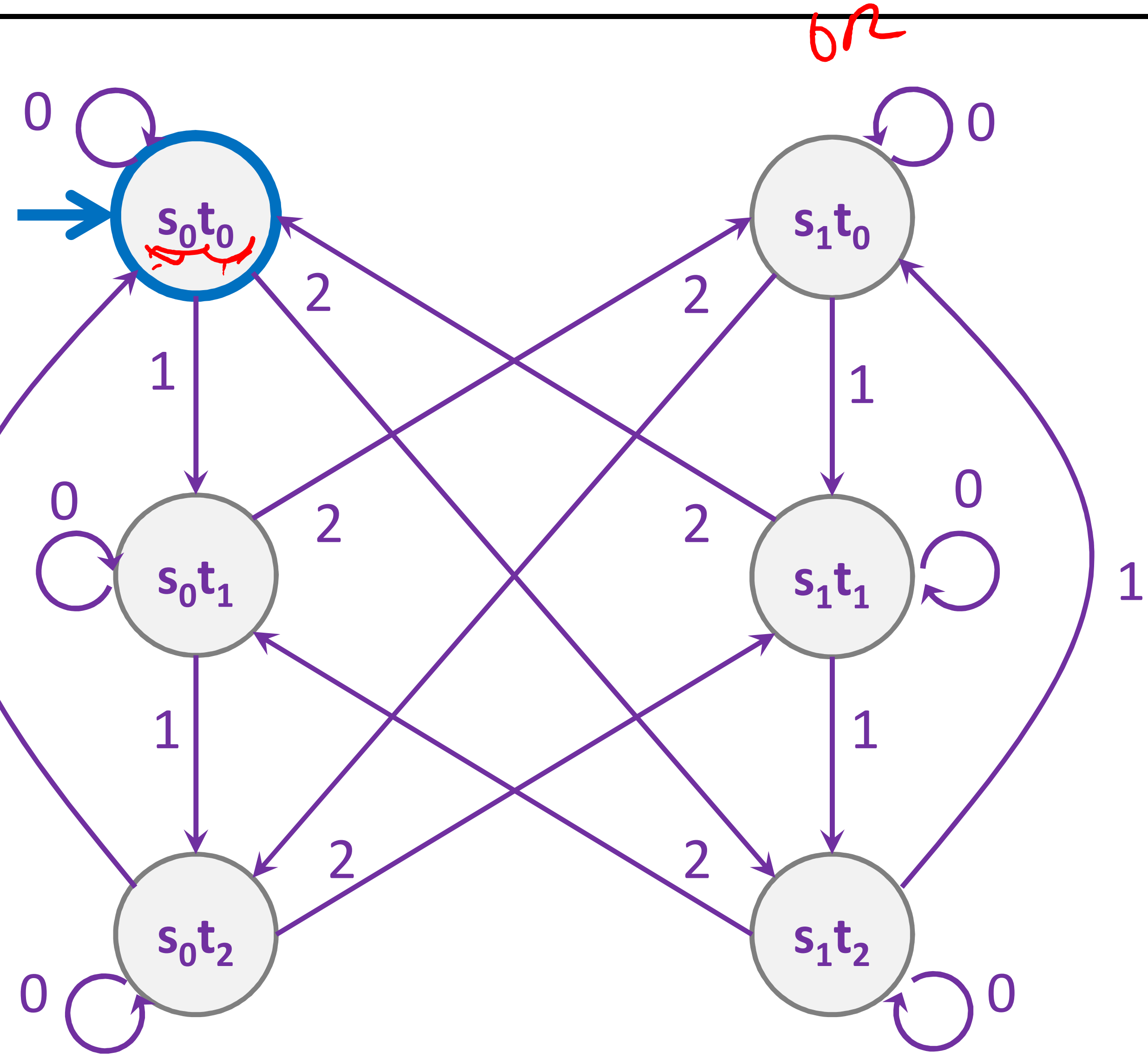
M_2: Strings where the sum of digits mod 3 is 0



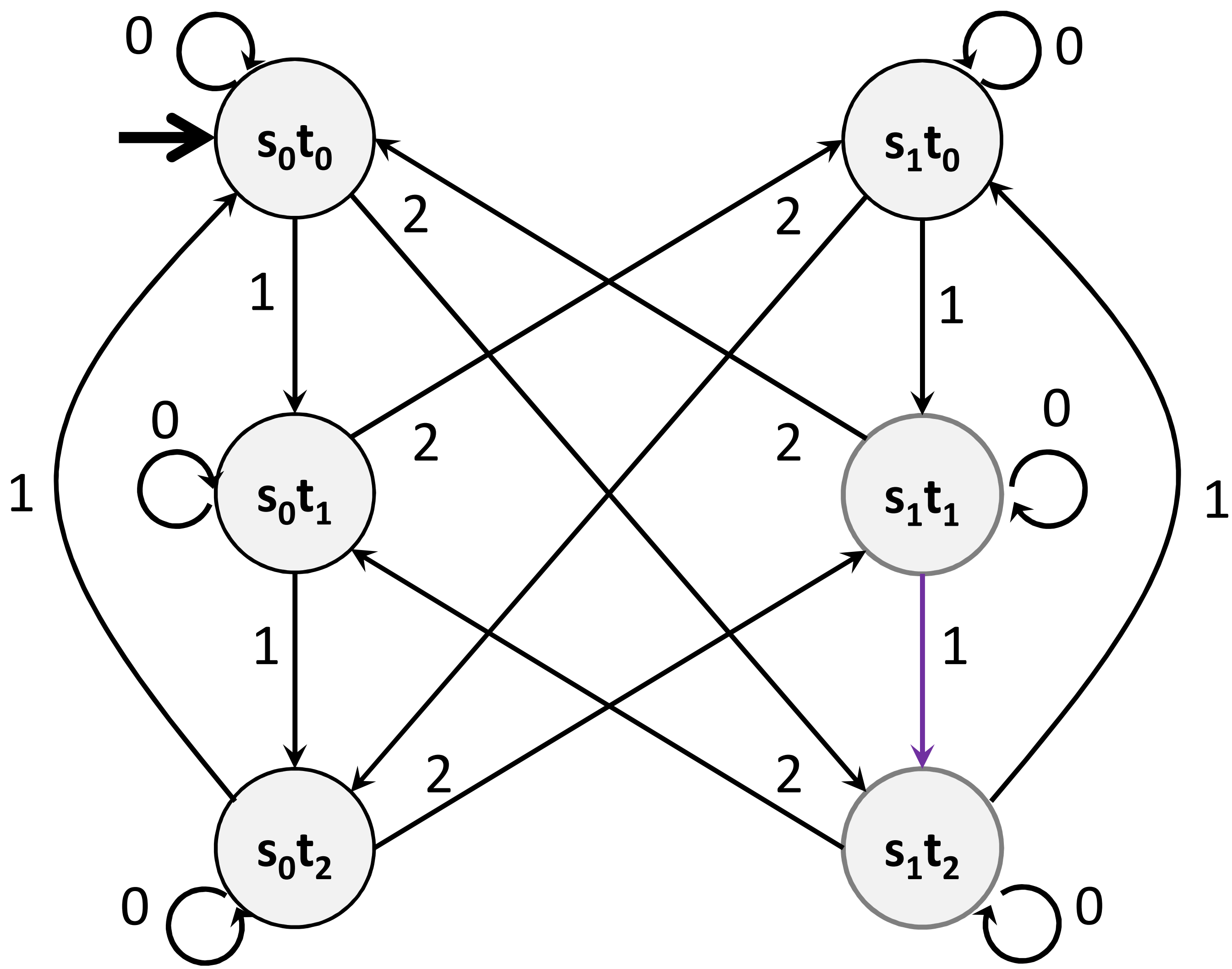
Strings over {0,1,2} w/ even number of 2's and mod 3 sum 0



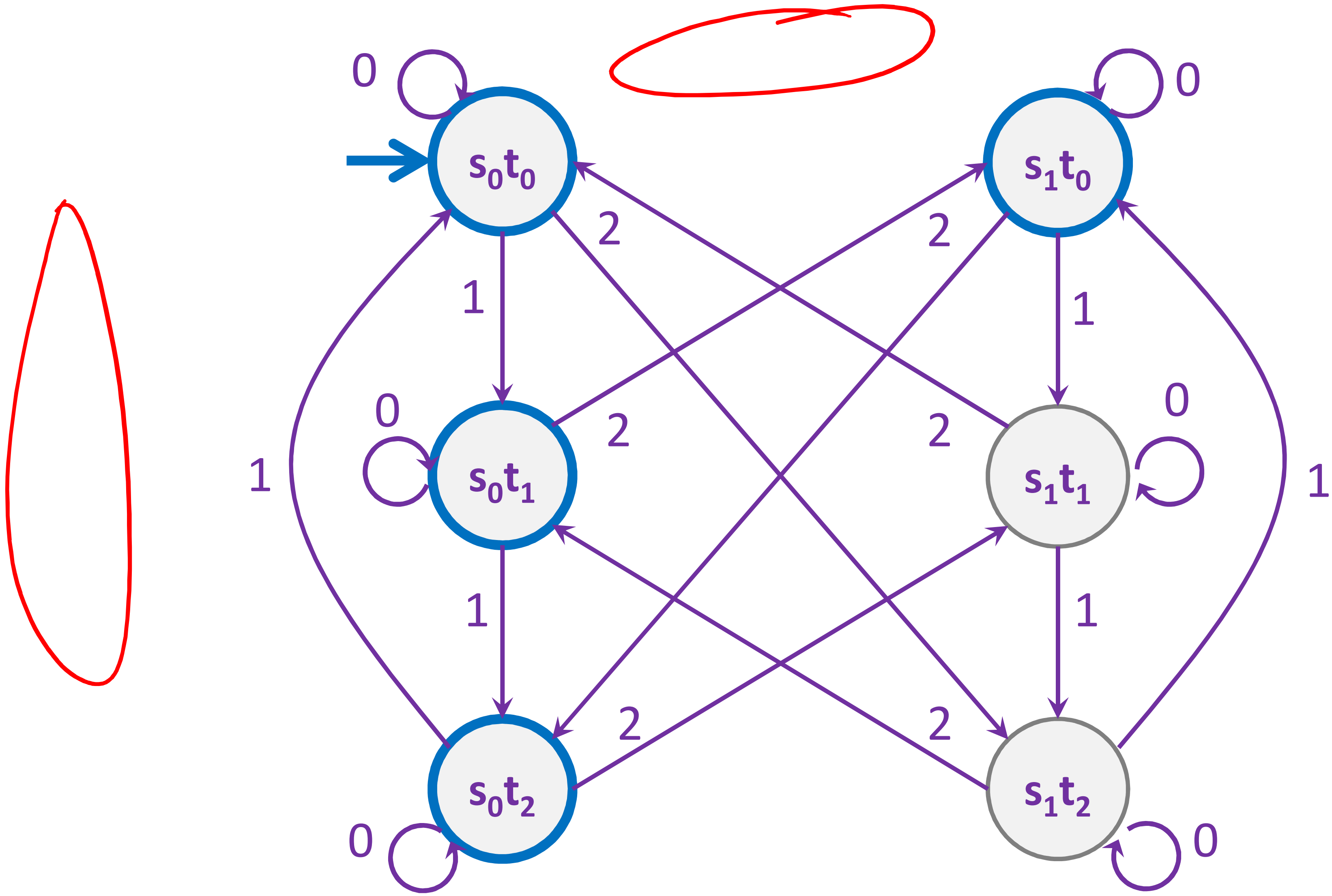
Strings over {0,1,2} w/ even number of 2's and mod 3 sum 0



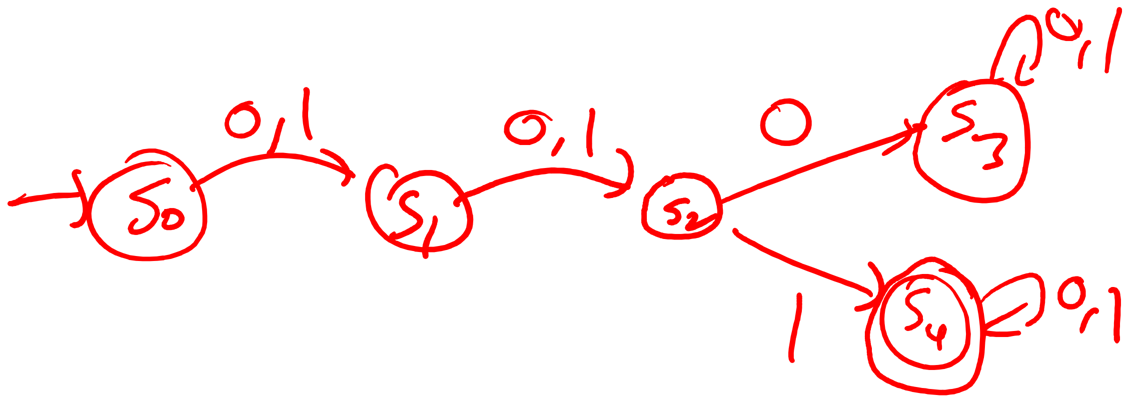
Strings over {0,1,2} w/ even number of 2's OR mod 3 sum 0?



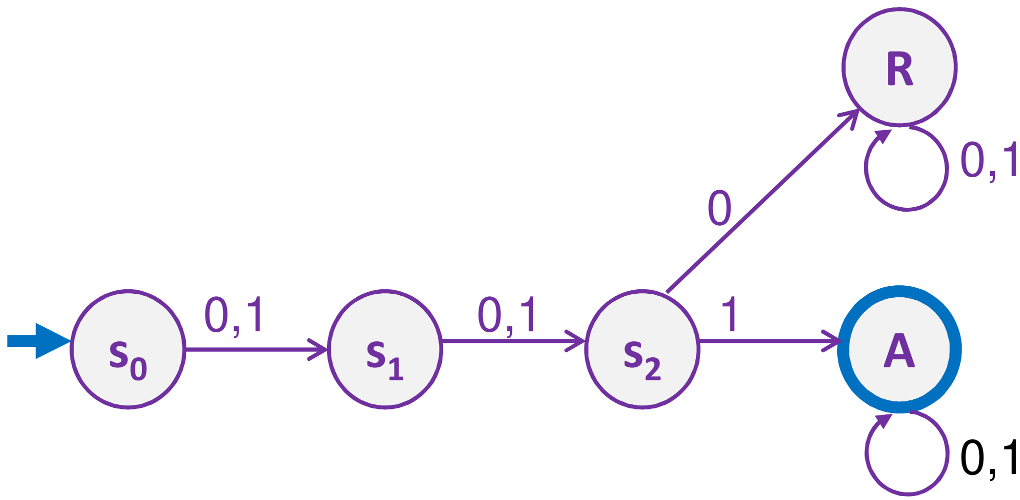
Strings over {0,1,2} w/ even number of 2's OR mod 3 sum 0



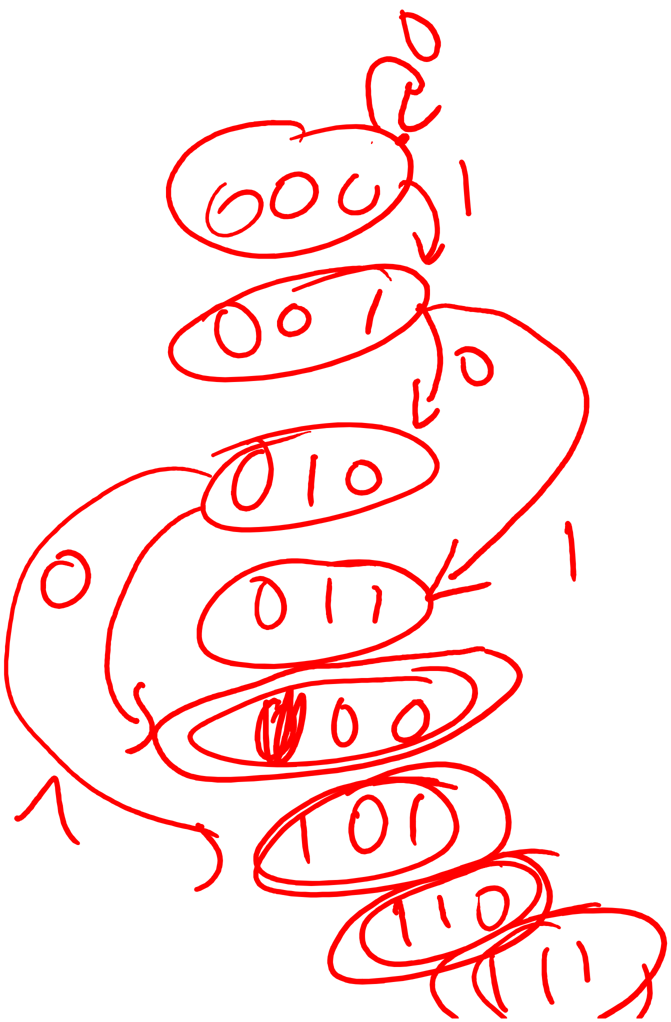
The set of binary strings with a 1 in the 3rd position from the start



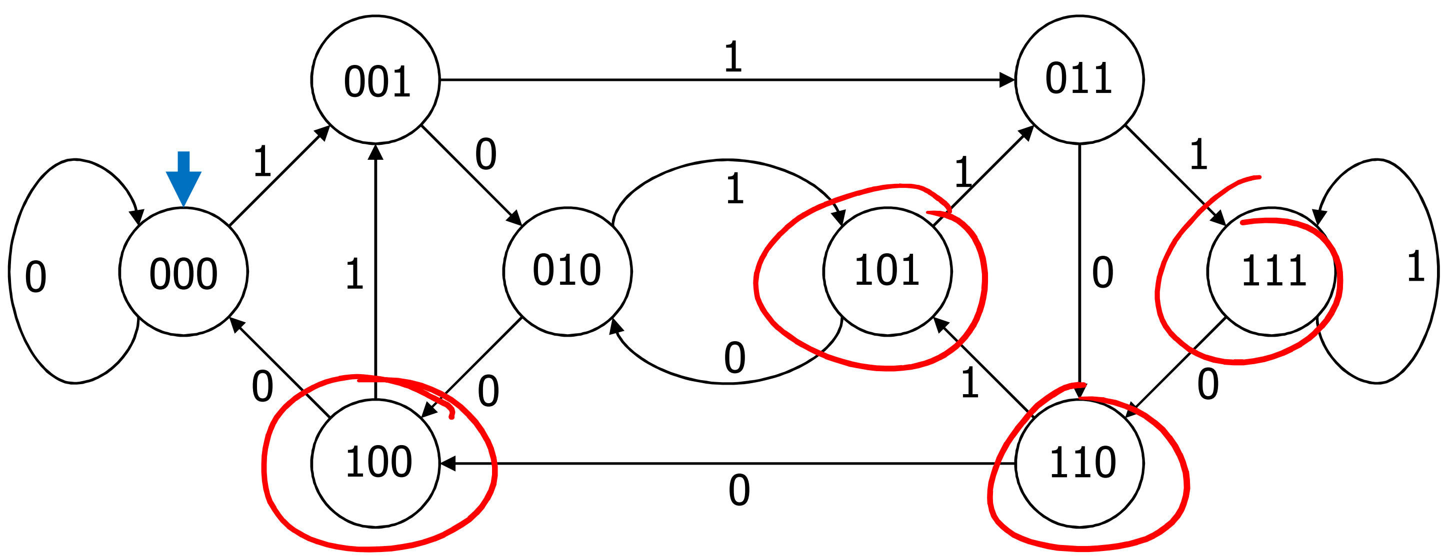
The set of binary strings with a 1 in the 3rd position from the start



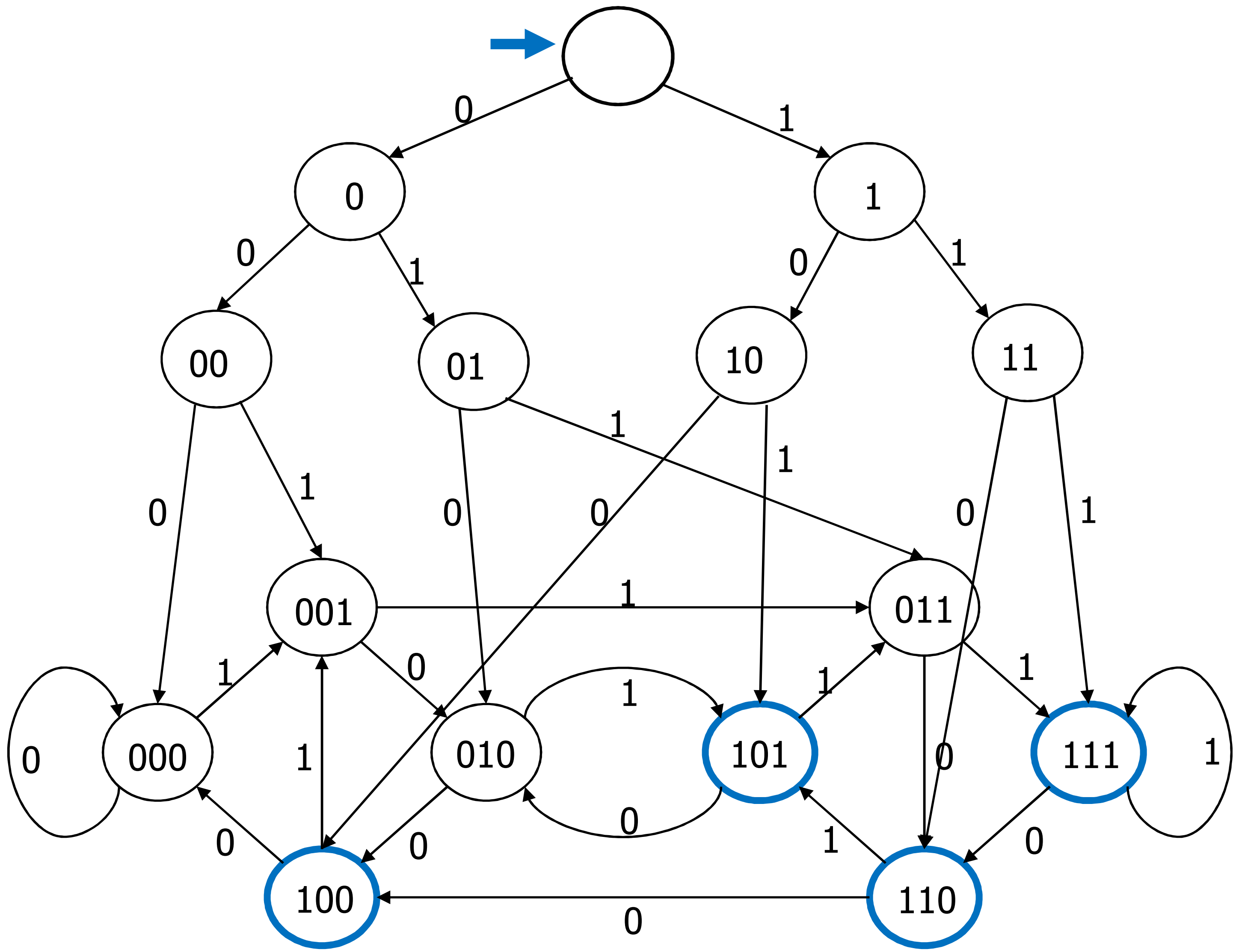
The set of binary strings with a 1 in the 3rd position from the end



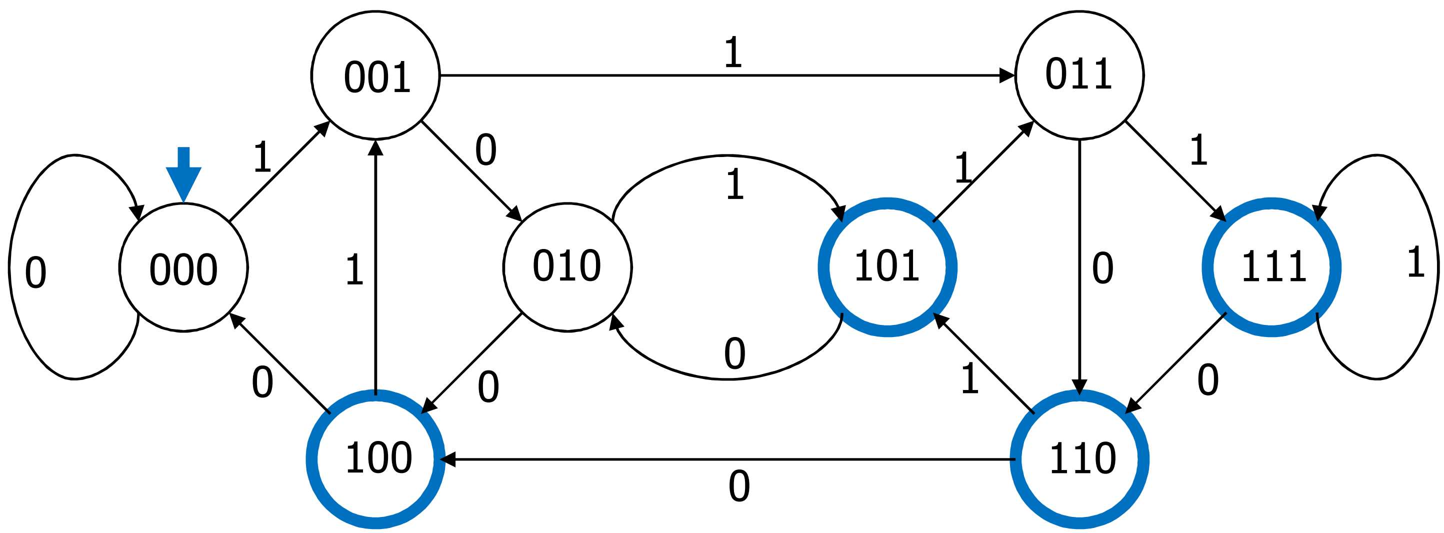
3 bit shift register "Remember the last three bits"



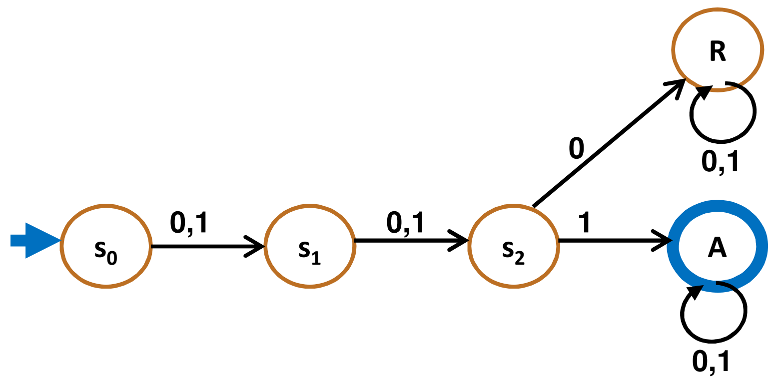
The set of binary strings with a 1 in the 3rd position from the end



The set of binary strings with a 1 in the 3rd position from the end

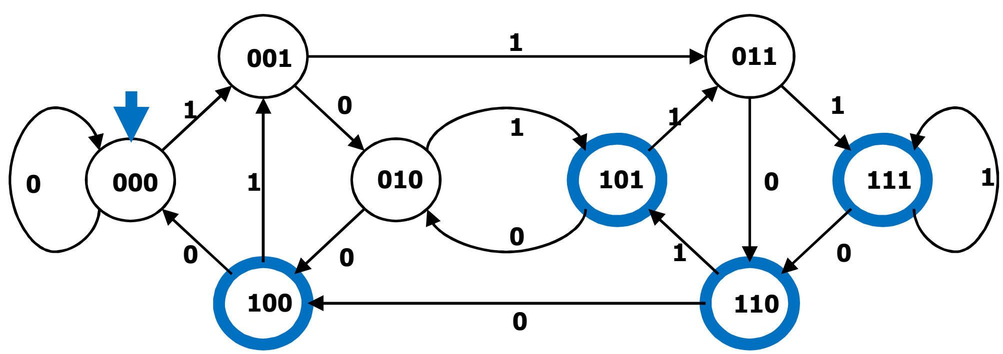


The beginning versus the end



nth position

n+2 state



2^n

Adding Output to Finite State Machines

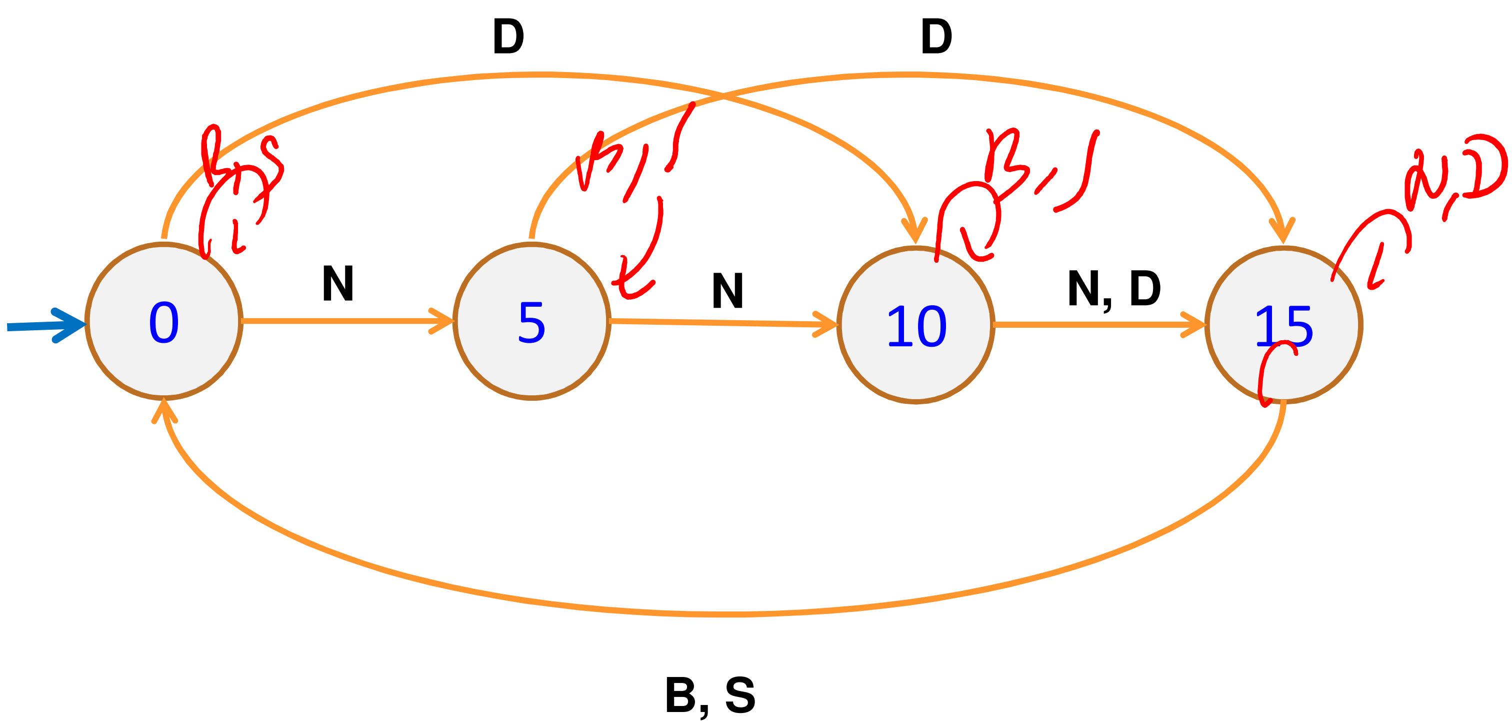
- So far we have considered finite state machines that just accept/reject strings
  - called "Deterministic Finite Automata" or DFAs

- Now we consider finite state machines that with output
  - These are the kinds used as controllers

Vending Machine

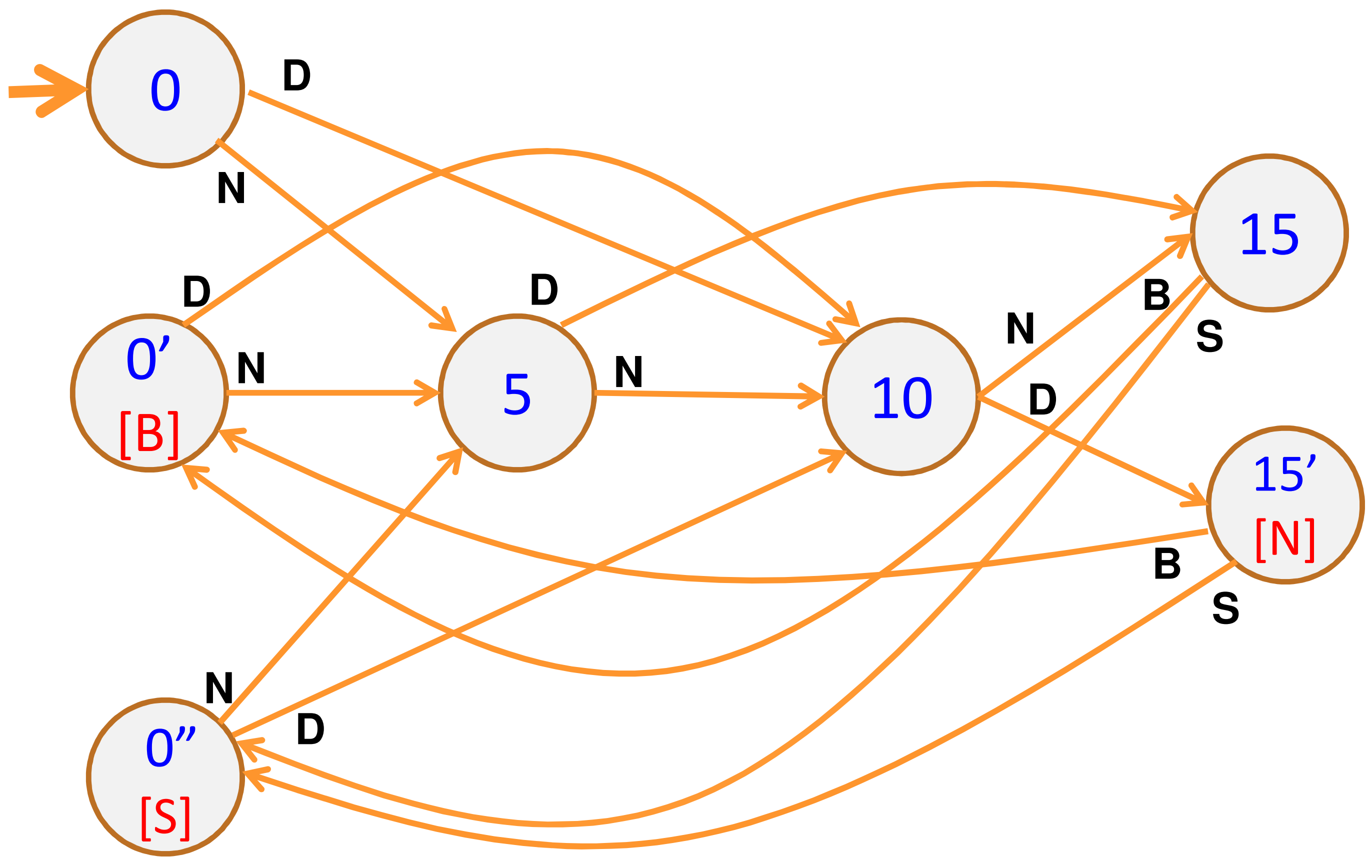
Enter 15 cents in dimes or nickels
Press S or B for a candy bar

Vending Machine, v0.1



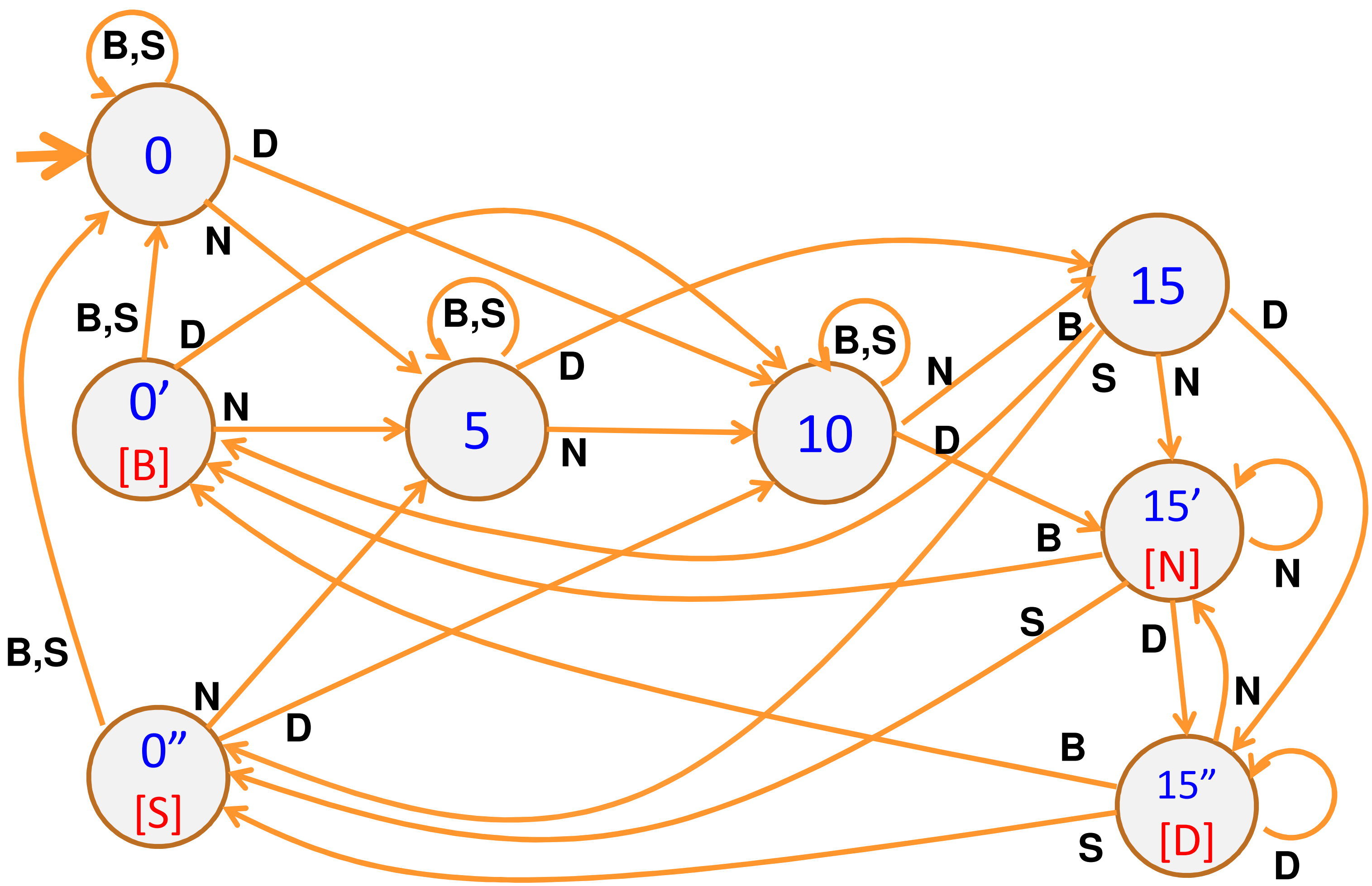
Basic transitions on **N** (nickel), **D** (dime), **B** (butterfinger), **S** (snickers)

Vending Machine, v0.2



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

Vending Machine, v1.0



Adding additional “unexpected” transitions to cover all symbols for each state