University of East Anglia

Foundations of Computing

Finite State Machine (FSM)

Dr Robert Blair

r.blair@uea.ac.uk

Introduction to basic analogue electronics

Outline

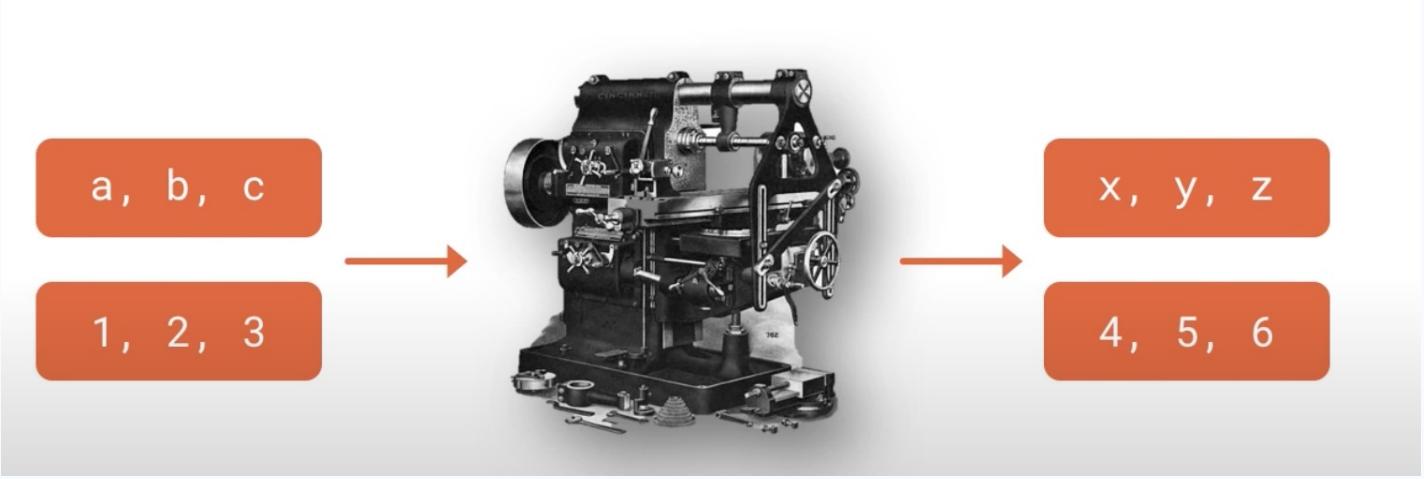
1. Example of FSM
2. Finite state (set)
3. State diagram
4. Basic control program
5. Code efficiency
6. Recap

Learning outcomes

- You will be able to define a Finite State Machine
- You can describe the characteristics of an FSM
- You will be aware of two ways in which an FSM can be represented
- You can create an FSM using an Arduino
- You can understand the basic program for an FSM using an Arduino
- You can improve the efficiency of the basic program

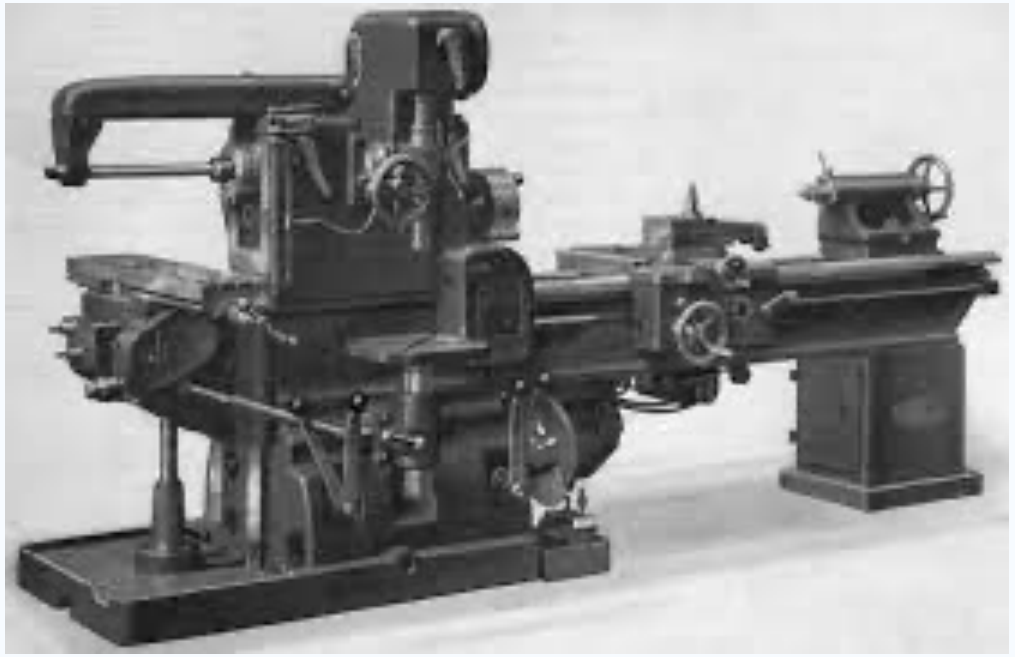
Finite State Machine

- A computing machine
- Fixed set of possible states – Finite states
- Accepts or does not accept an input
- Fixed set of possible inputs
- Fixed set of possible outputs
- Limited memory availability – Finite
- Output not always necessary



Finite State Machine

- Not a physical machine



- An abstract creation
  - model simple computation and decision making

Finite State Machine

- 'Machine' which takes an input
- Accepts input
- Changes state – or
- Remains in same state

New state = Current state + Input value

Finite State Machine

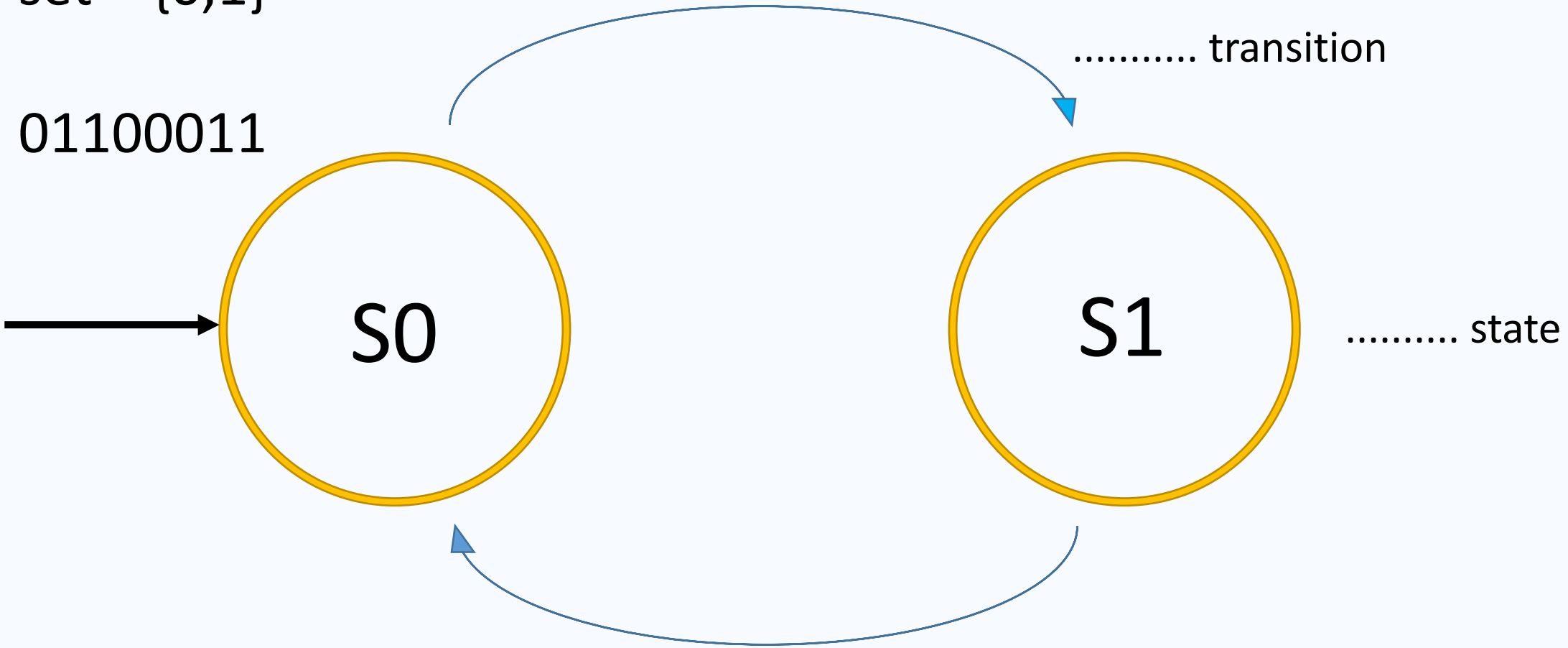
- Consider a ball point pen
- Click the pen button
- Change state
- Click button again
- Change state
- Same input

- State depends upon previous state
- History of states can be summed by current state

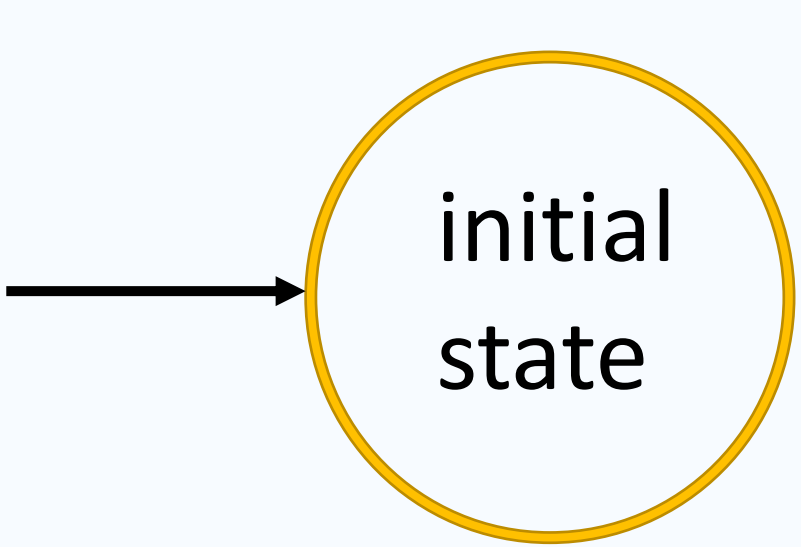
State Transition Diagrams

set = {0,1}

01100011

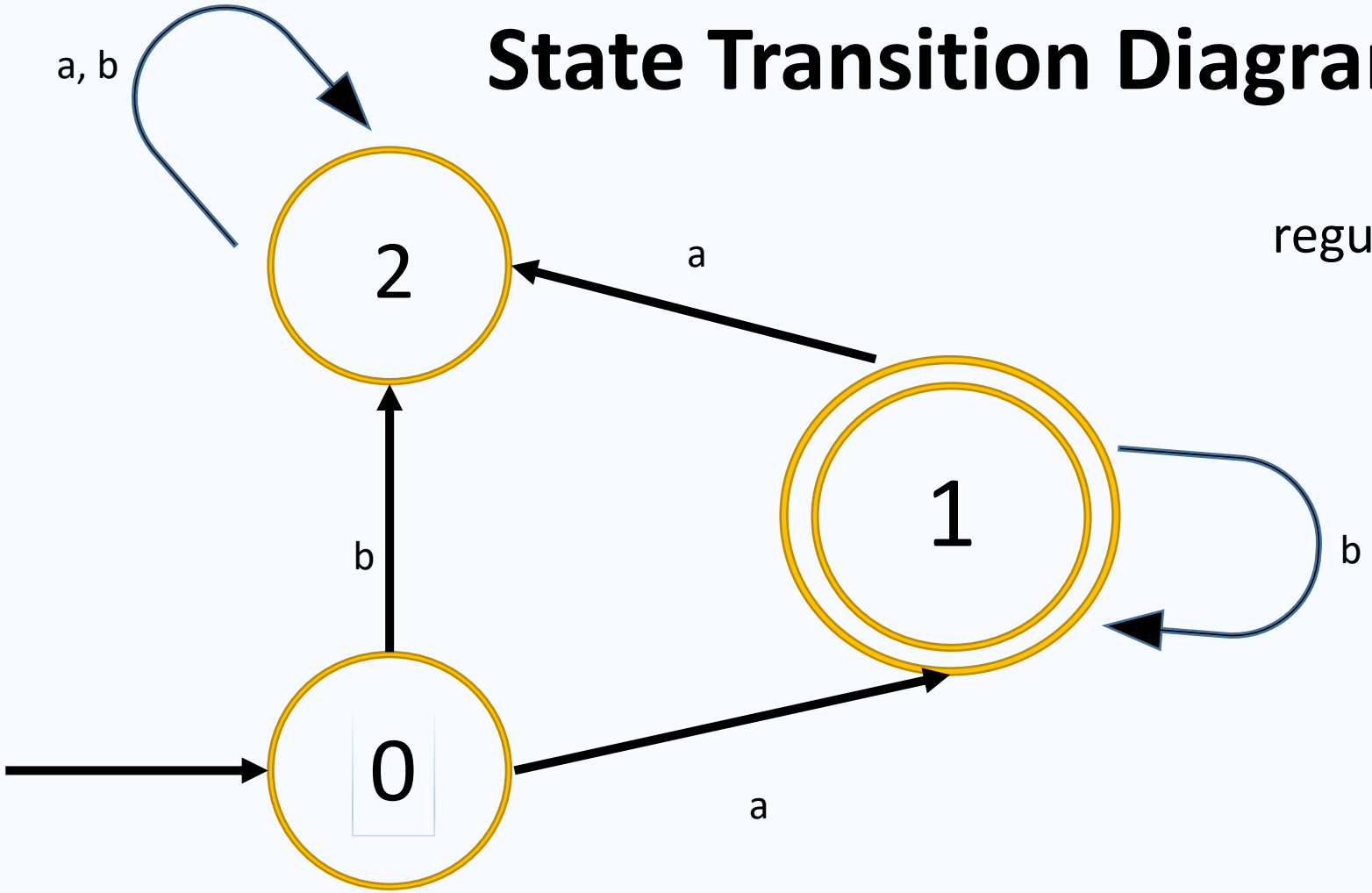


State Transition Diagrams



aka – accepting state

State Transition Diagrams



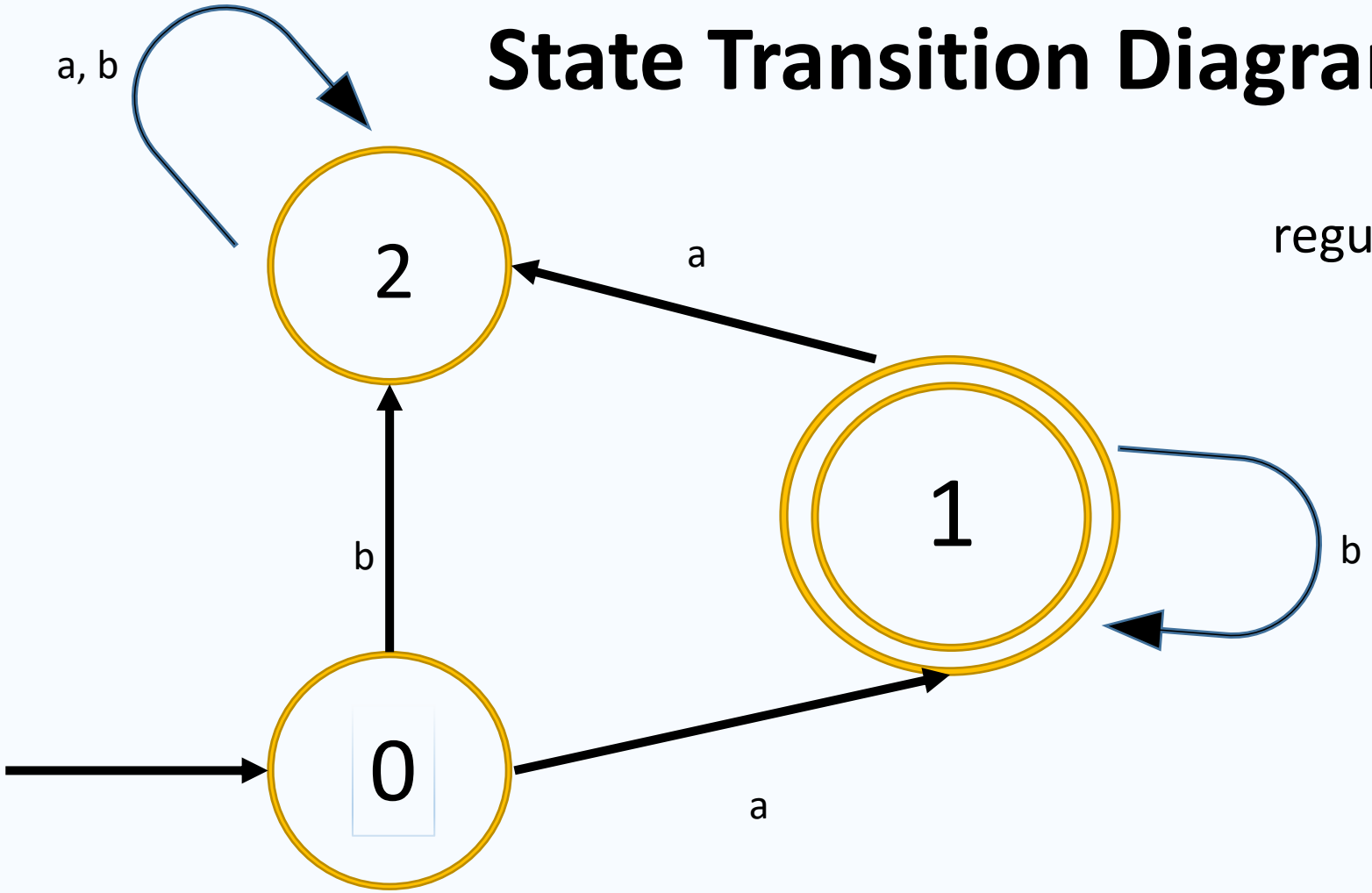
regular expression = a,b*

- *one or more

- abbbb ✓
- ab ✓
- aaa ✓
- ba *X*

aka – accepting state

State Transition Diagrams



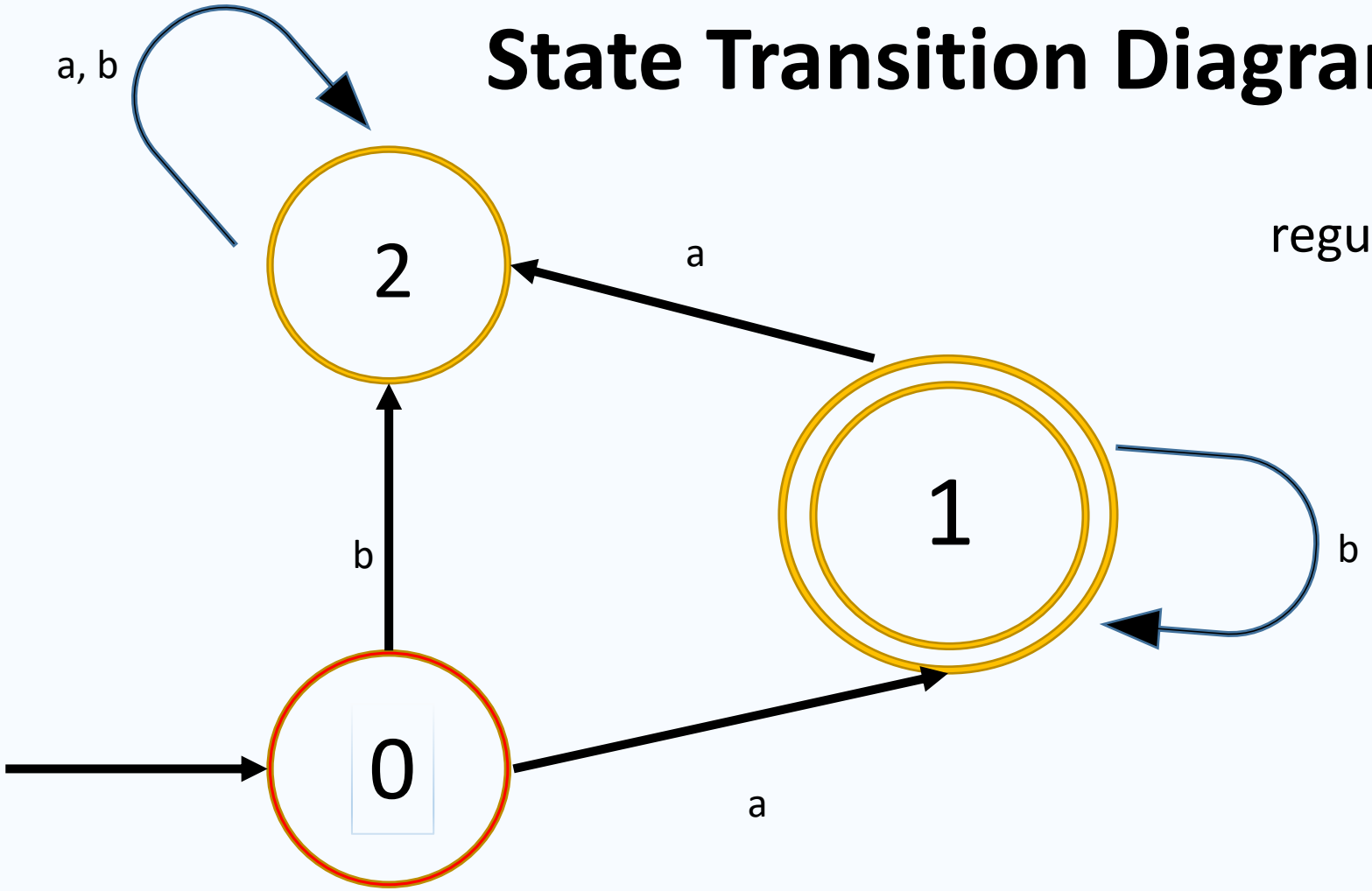
regular expression = a,b*

- *one or more

- abbbb ✓
- ab ✓
- aaa *X*
- ba *X*

is this input (a,b*) acceptable according to the regular expression?

State Transition Diagrams

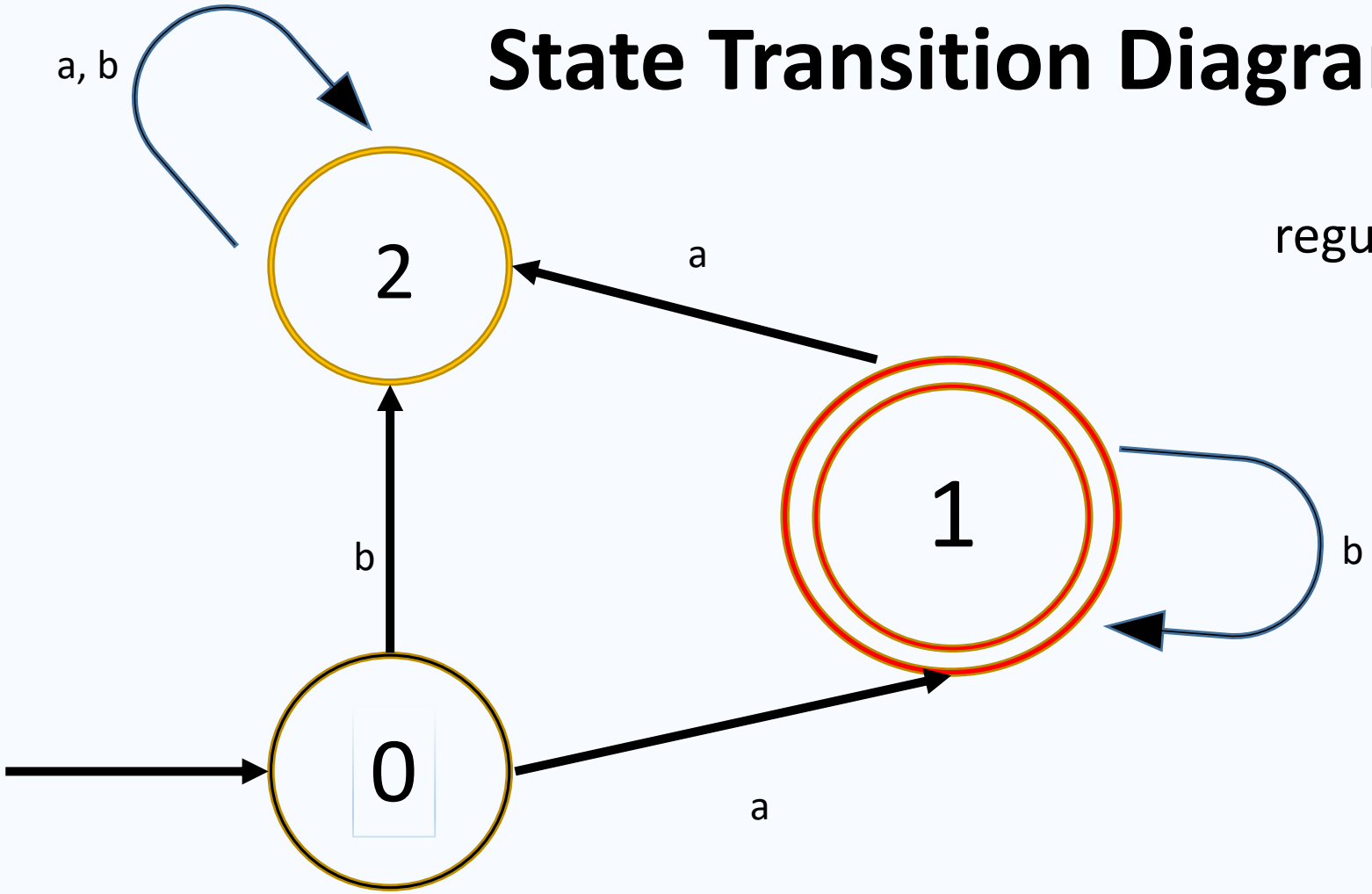


regular expression = a,b*

- *one or more
- a

is this input (a,b*) acceptable according to the regular expression?

State Transition Diagrams



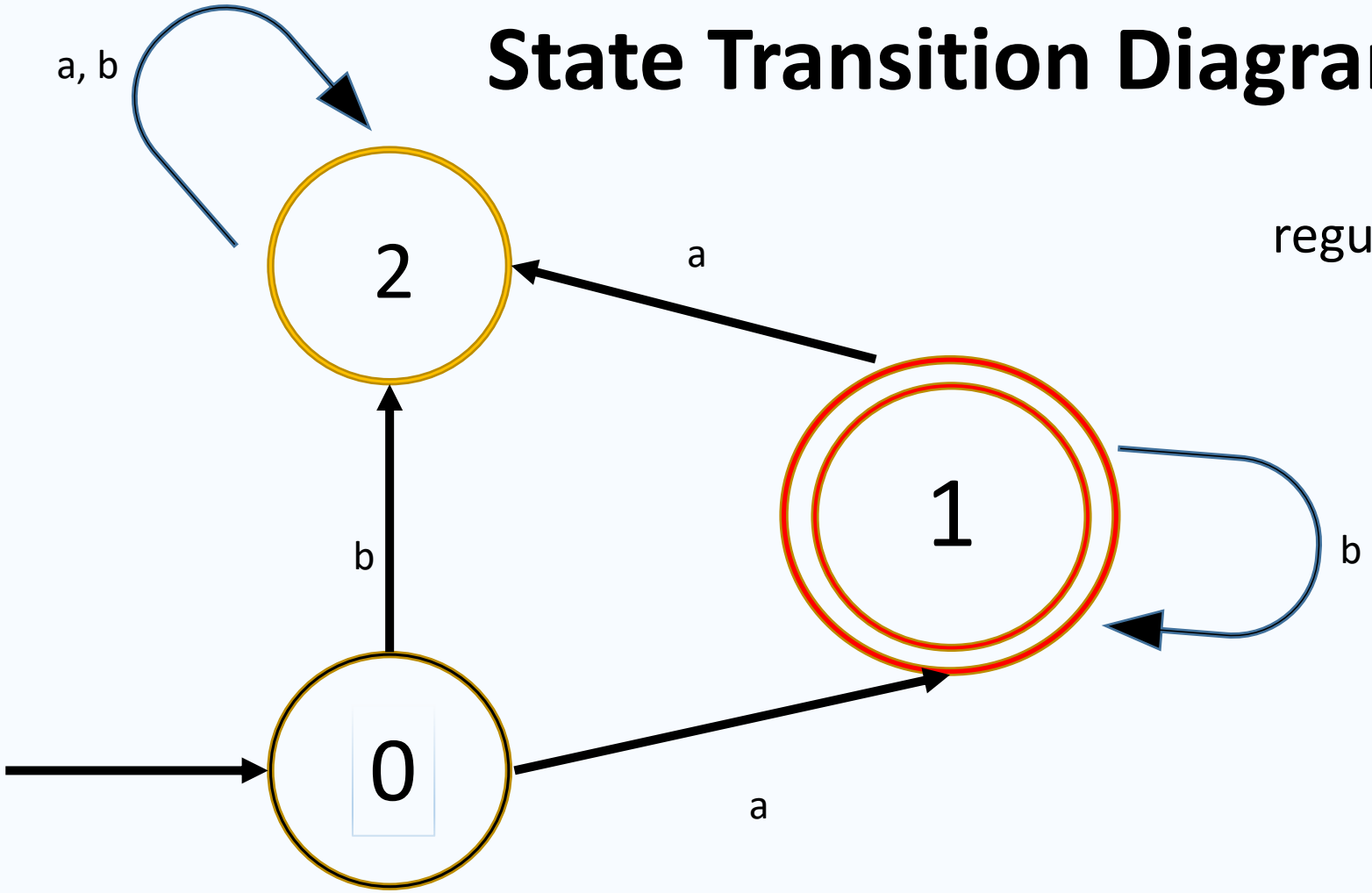
regular expression = a,b*

- *one or more

- a
- b

is this input (a,b*) acceptable according to the regular expression?

State Transition Diagrams



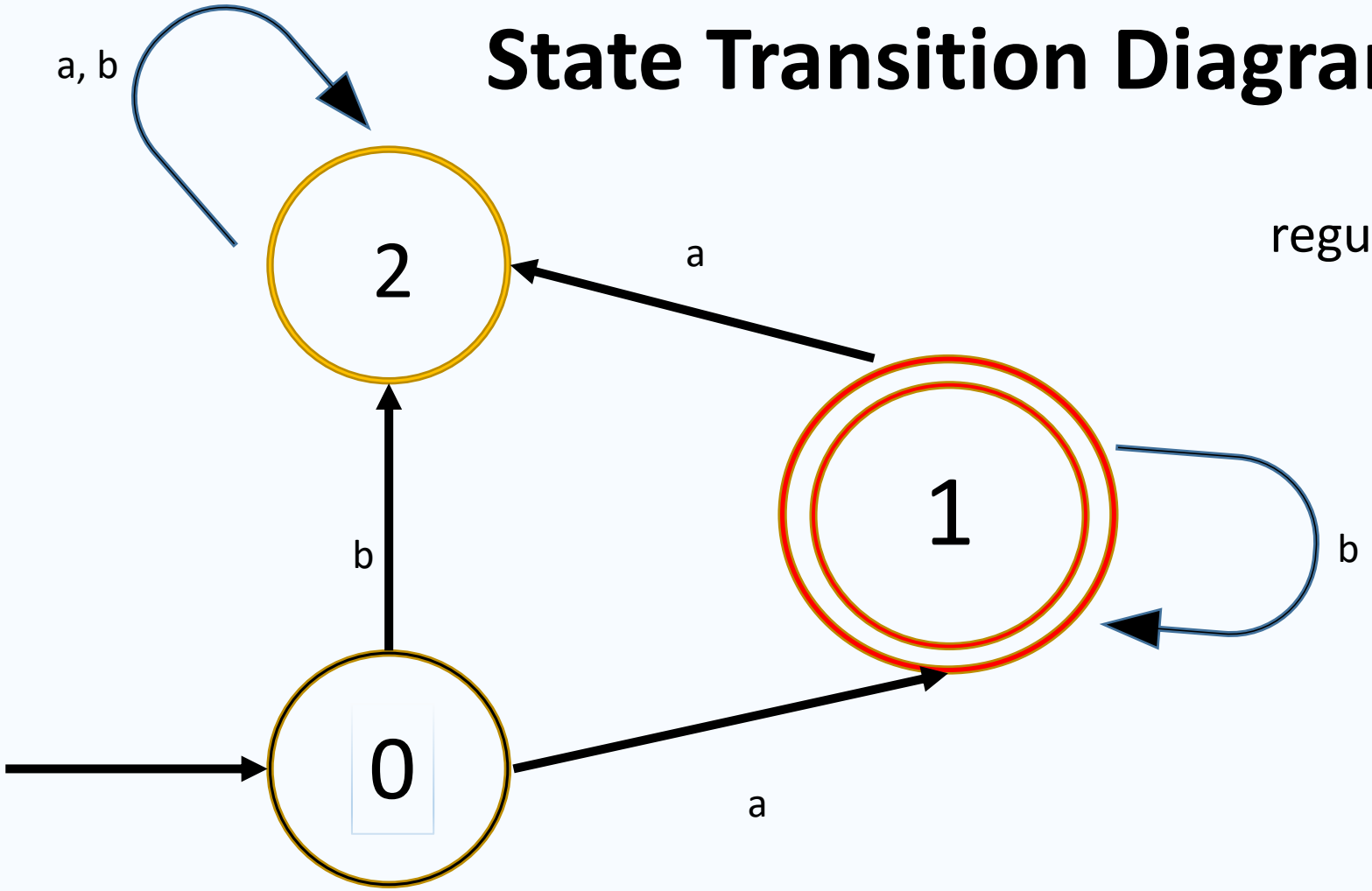
regular expression = a,b*

- *one or more

- input a at state 1?
- input b at initial state?

is this input (a,b*) acceptable according to the regular expression?

State Transition Diagrams



regular expression = a,b*

- *one or more

- input a at state 1?
- input b at initial state?

No path to 1 therefore
an unaccepting state

is this input (a,b*) acceptable according to the regular expression?

State Transition Tables

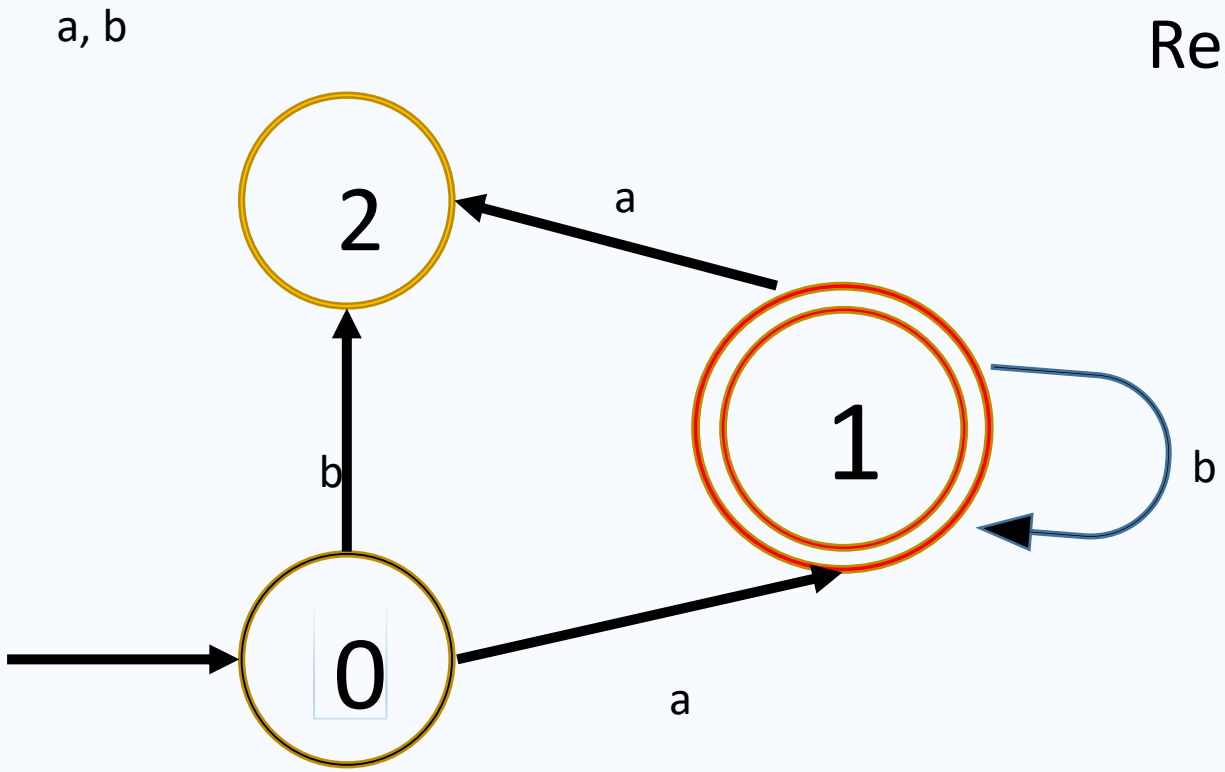
Consider pen state

Before input

After input

INPUT	CURRENT STATE	NEXT STATE
BUTTON PRESSED	NIB RETRACTED	NIB EXTENDED
BUTTON PRESSED	NIB EXTENDED	NIB RETRACTED

State Transition Tables



Regular expression state transition table

INPUT	CURRENT STATE	NEXT STATE
a	0	1
b	0	2
a	1	2
b	1	1
a	2	2
b	2	2

Summary

An FSM is an abstract computing machine that has

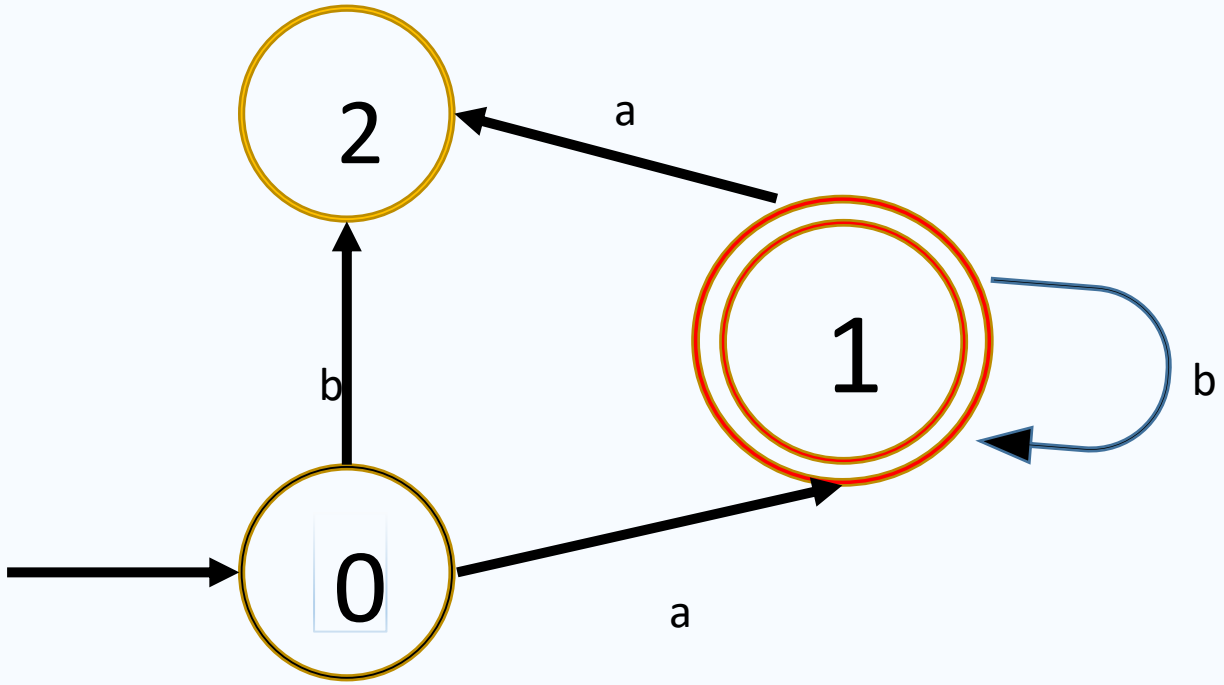
- a fixed set of possibl states
- a set of inputs that change a state
- a set of possible outputs

Characteristics of FSM: next state = Current state + Input value

Summary

Representations

a, b



Regular expression state transition table

INPUT	CURRENT STATE	NEXT STATE
a	0	1
b	0	2
a	1	2
b	1	1
a	2	2
b	2	2

Example – Arduino