AUTOMATON WITH OUTPUT

ITEU133

AUTOMATA AND THEORY OF COMPUTATION

COURSE SYLLABUS

V. Automaton with Output

- Moore Machines
- Mealy Machines
- Equivalence of Moore and Mealy Machine
- Applications of Automaton with Output

FINITE AUTOMATA AS MATHEMATICAL MODELS OF COMPUTERS

letters of words	⟷	input
transitions	⟷	execution of instructions (a program)
state	⟷	contents of memory
?	⟷	output

FINITE AUTOMATA WITH OUTPUT

- A finite-state machine *M* = (*Q*, S,*O*,d, l, *q*) 0 consists of a finite
- set *Q* of states, a finite input alphabet S, a finite output alphabet *O*,
- a transition function δ that assigns to each state and input pair a new state, an output function ε that assigns to each state and input pair an output, and an initial state *q*0.

FINITE AUTOMATA WITH OUTPUT

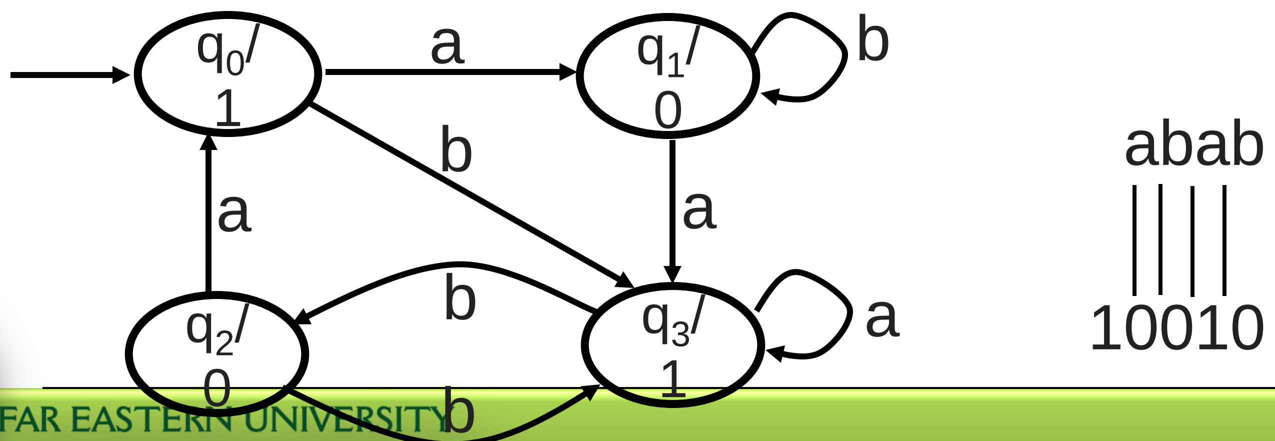
- A Moore machine is a collection of 5 things:
  1. a finite set of states $q_0, q_1, q_2, \ldots$, where q_0 is designated the start state
  2. an alphabet Σ of input letters
  3. an alphabet Γ of output characters
  4. a transition table that shows for each state and each input letter what state to go to next.
  5. an output table that shows what character is output (or printed) when entering a state.

$$(\text{state, letter from } \Sigma) \xrightarrow{\text{transition}} \text{state}$$

FINITE AUTOMATA WITH OUTPUT

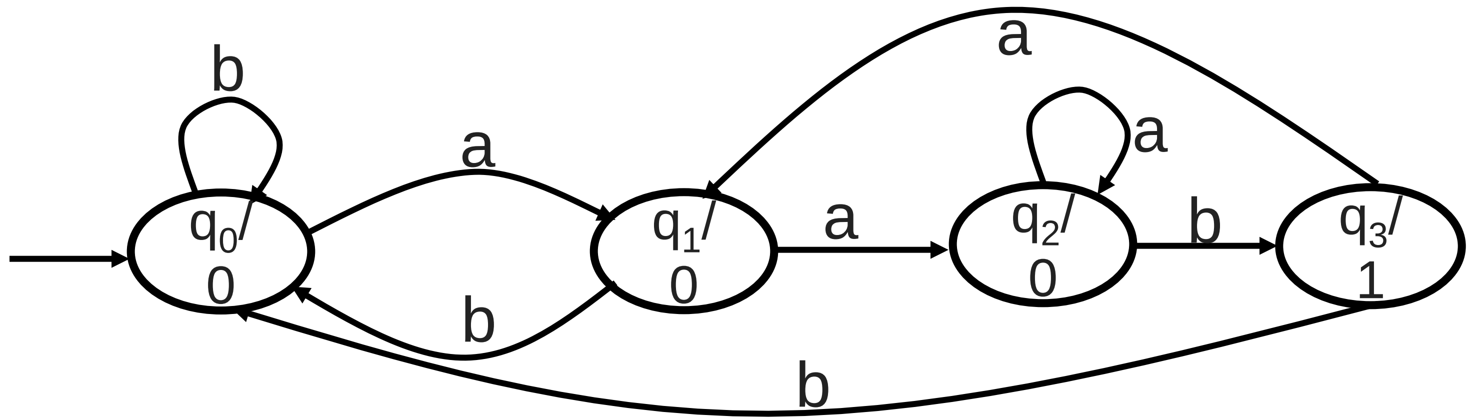
Example: states = $\{q_0,q_1,q_2,q_3\}$ $\Sigma=\{a,b\}$ $\Gamma=\{0,1\}$

Old state	Output by the old state	New state After input a	After input b
$-q_0$	1	q_1	q_3
q_1	0	q_3	q_1
q_2	0	q_0	q_3
q_3	1	q_3	q_2



FINITE AUTOMATA WITH OUTPUT

Example: A machine that counts 'aab's



Input		a	a	a	b	a	b	b	a	a	b	b
State	q_0	q_1	q_2	q_2	q_3	q_1	q_0	q_0	q_1	q_2	q_3	q_0
Output	0	0	0	0	1	0	0	0	0	0	1	0

Printing 1 ≈ final state

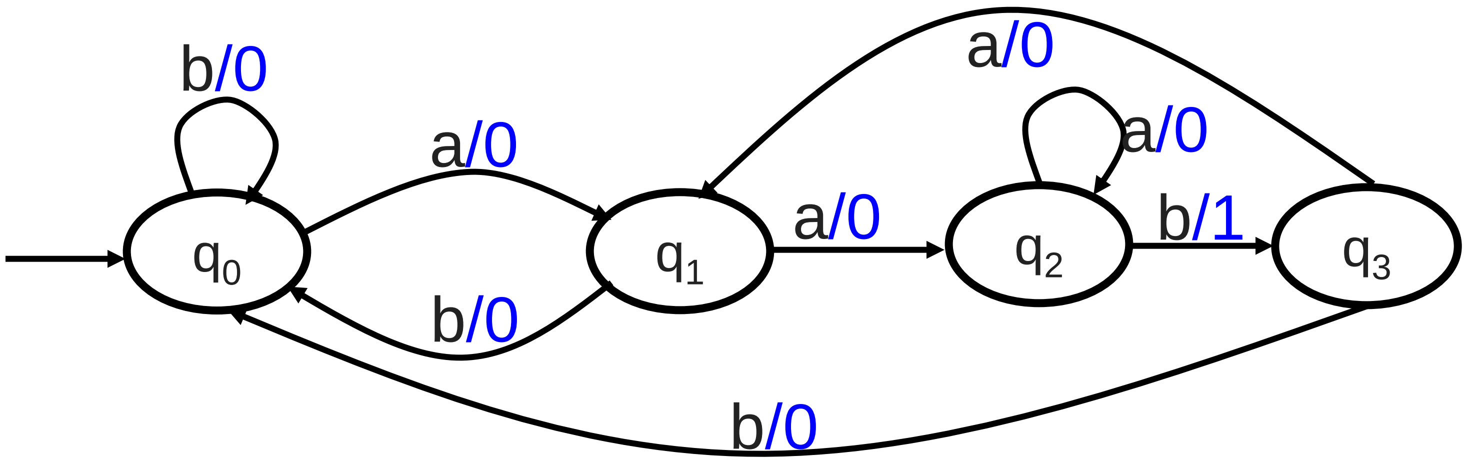
words that end in aab

FINITE AUTOMATA WITH OUTPUT

- A Mealy machine is:
  1. a finite set of states $q_0, q_1, q_2, \ldots$, where q_0 is designated the start state
  2. an alphabet Σ of input letters
  3. an alphabet Γ of output characters
  4. a finite set of transitions that indicate, for each state and letter of the input alphabet, the state to go to next and the character that is output (printed).

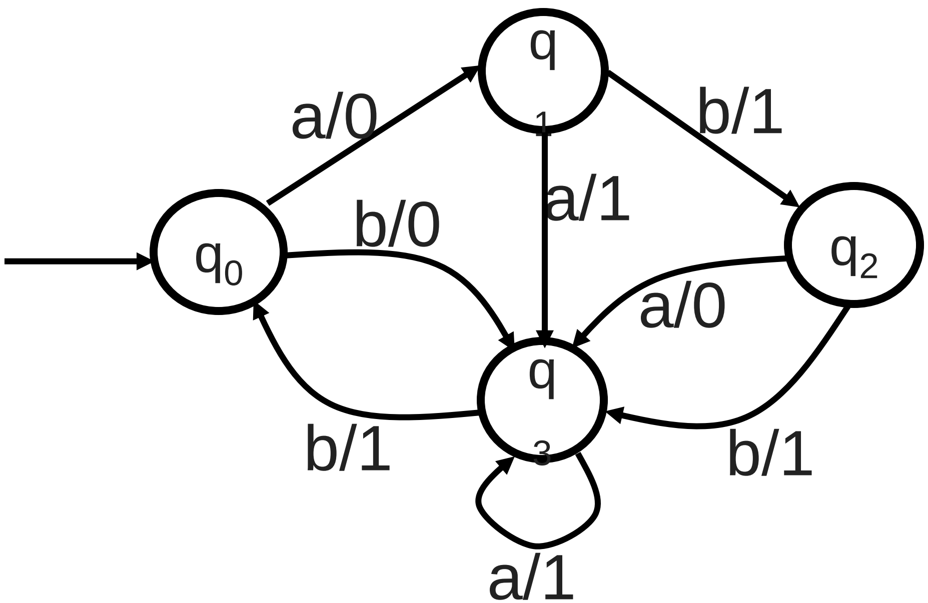
$$\text{state} \xrightarrow{\text{input/output}} \text{state}$$

FINITE AUTOMATA WITH OUTPUT



FINITE AUTOMATA WITH OUTPUT

Example



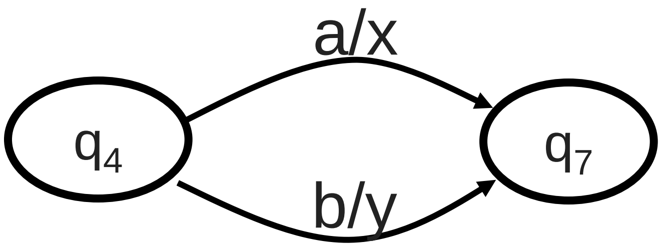
baaba

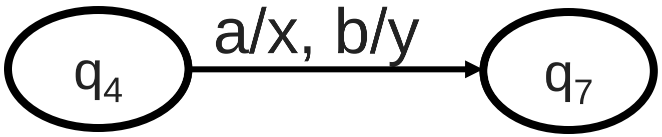
| | | | |

01110

FINITE AUTOMATA WITH OUTPUT

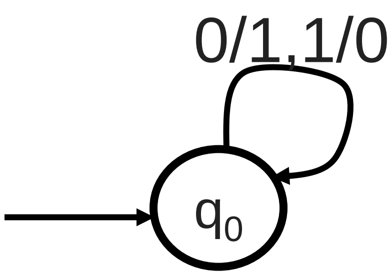
Abbreviation:





Example: binary complement

$\Sigma=\{0,1\}$ $\Gamma=\{0,1\}$

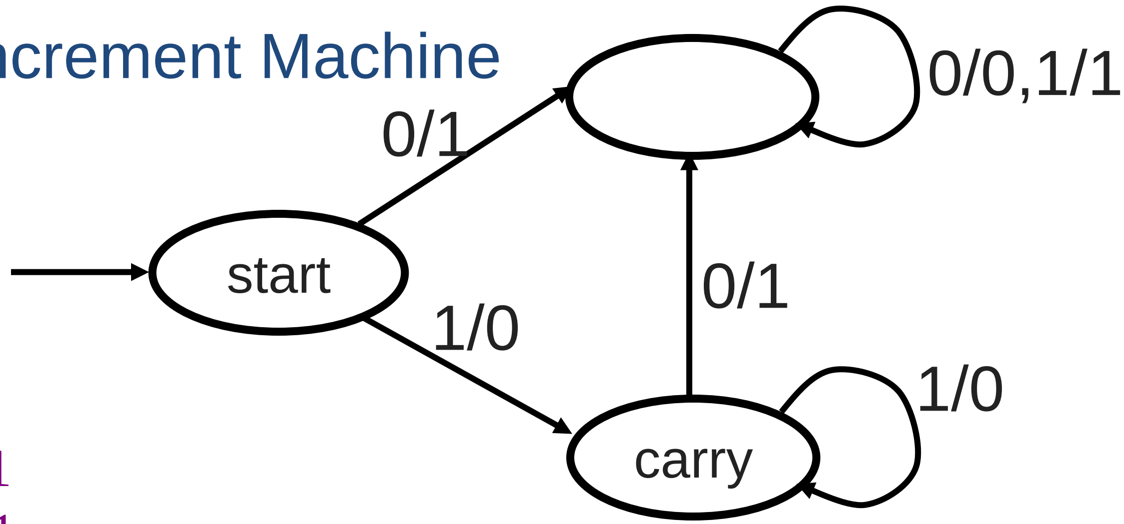


101 001010

010 110101

FINITE AUTOMATA WITH OUTPUT

Example: an Increment Machine



decimal: 11
binary : 1011
reverse: 1101

00 11
reverse: 1100
decimal: 12

1111

(overflow)

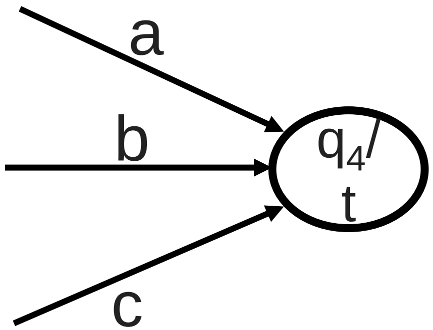
0000

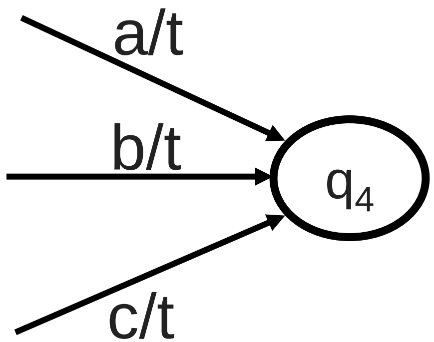
FINITE AUTOMATA WITH OUTPUT

Definition: Let Mo be a Moore machine that prints x in the start state. Let Me be a Mealy machine. The two machines are equivalent if for every input, whenever the output from Me is w, the output from Mo is xw.

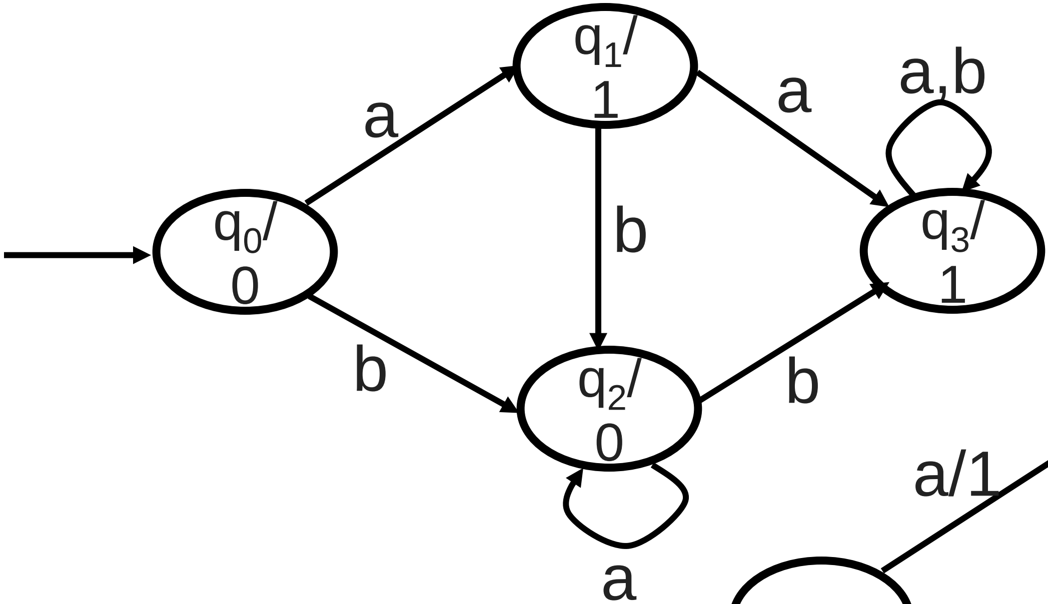
Theorem: For every Moore machine, there exists a Mealy machine that is equivalent to it.

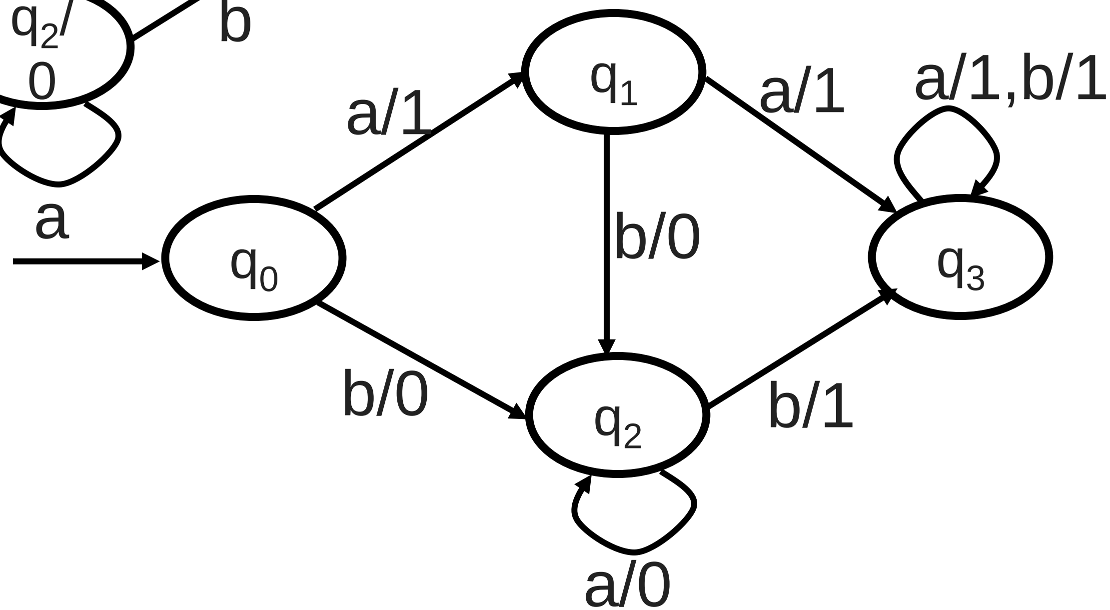
Proof: By constructive algorithm





FINITE AUTOMATA WITH OUTPUT



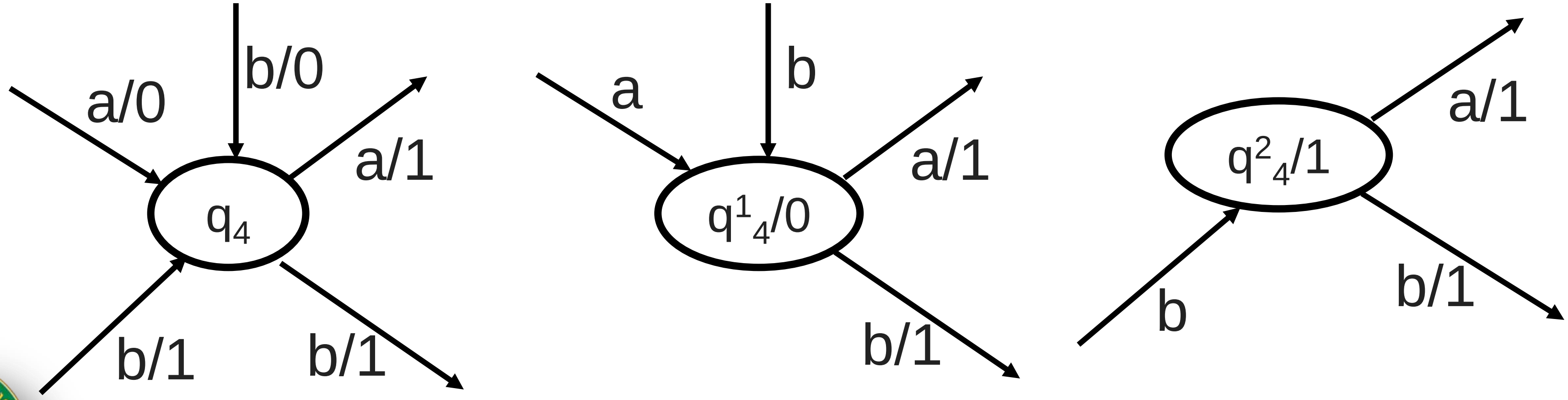


FINITE AUTOMATA WITH OUTPUT

Theorem: For every Mealy machine, there exists a Moore machine that is equivalent to it.

Proof: By constructive algorithm

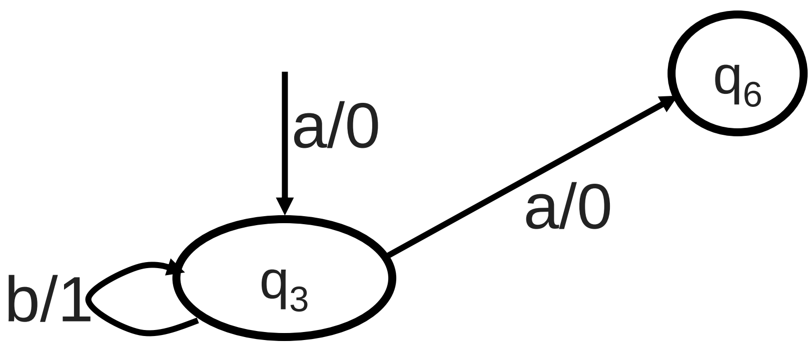
One copy of the state for each letter in Γ that labels an arrow entering it.

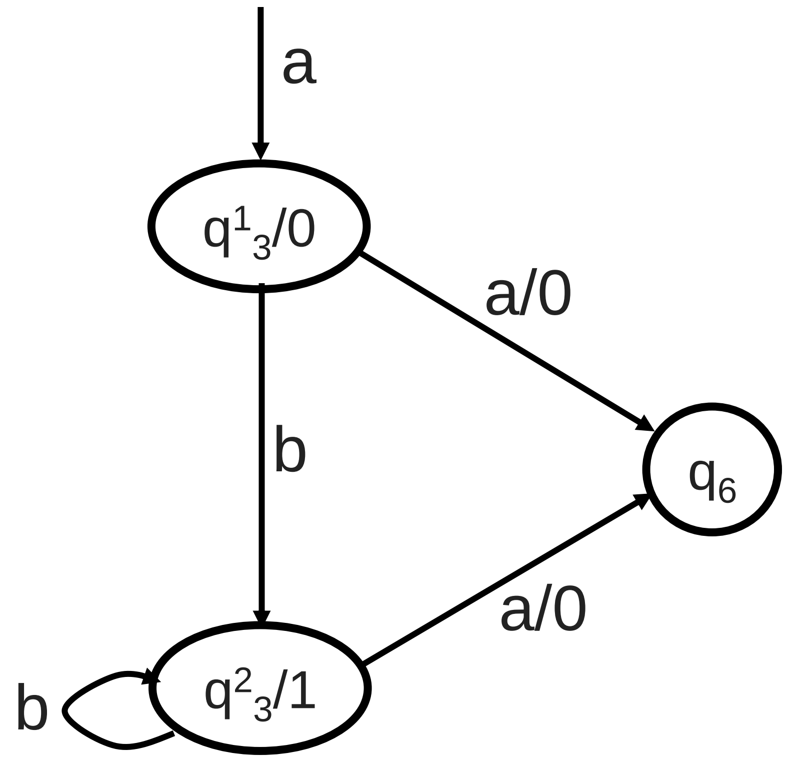


FINITE AUTOMATA WITH OUTPUT

We must consider

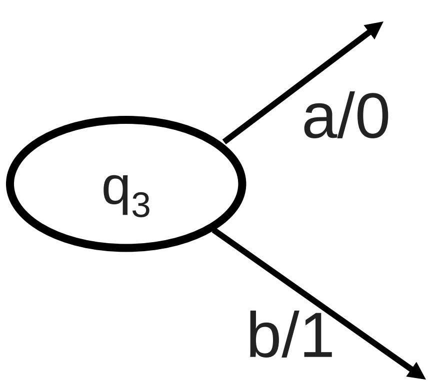
1. incoming edges
2. outgoing edges
3. loops





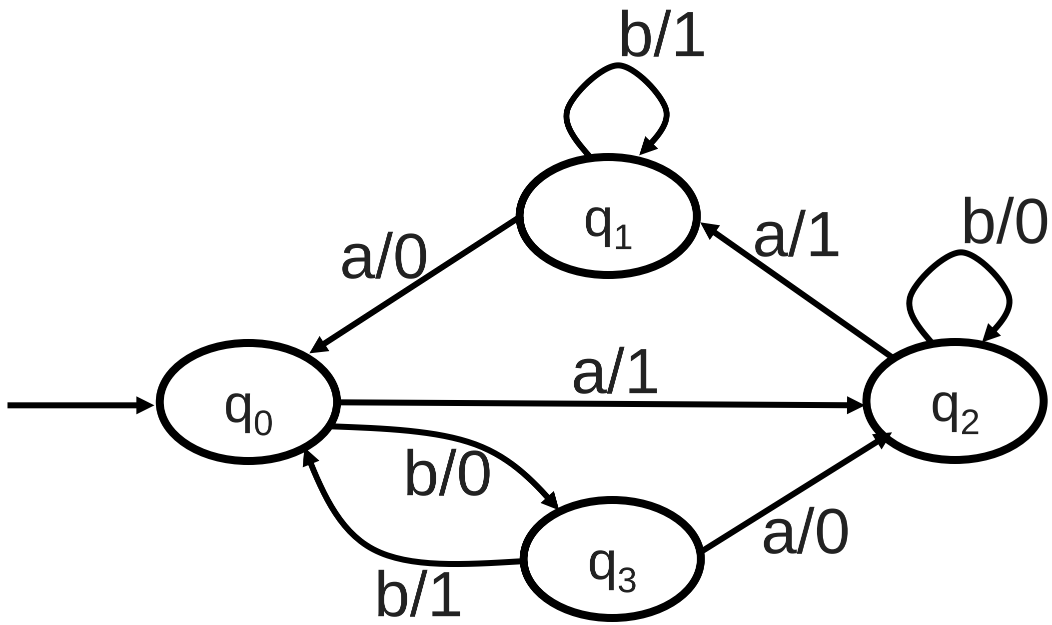
FINITE AUTOMATA WITH OUTPUT

If there is no entering arrow, choose any letter from Γ.

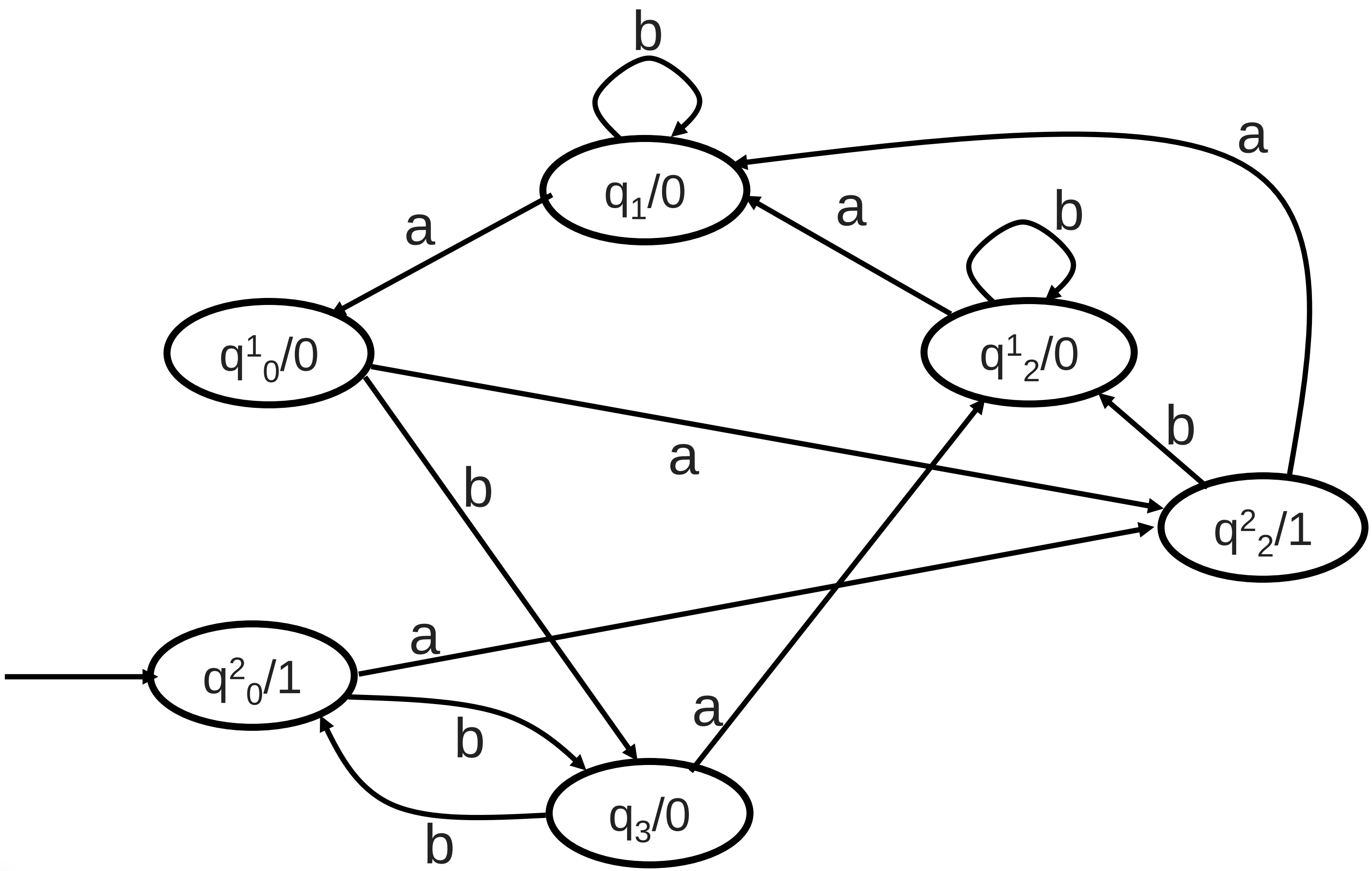


Choose any copy of q_0 as the start state.

FINITE AUTOMATA WITH OUTPUT

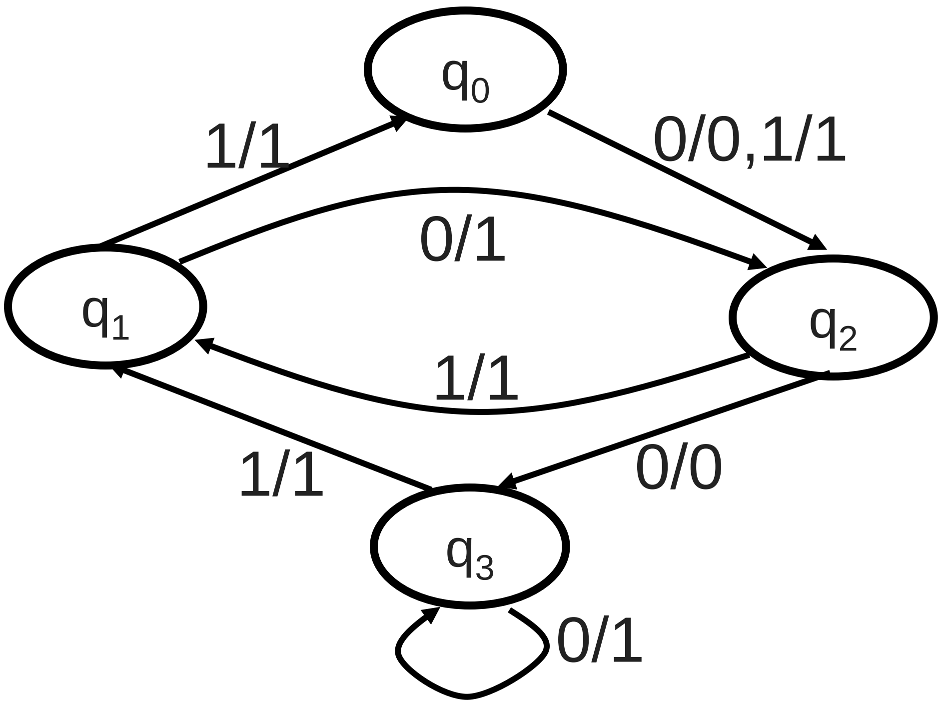


FINITE AUTOMATA WITH OUTPUT



FINITE AUTOMATA WITH OUTPUT

Old state	Input 0 new state	Input 0 output	Input 1 new state	Input 1 output
q_0	q_2	0	q_2	1
q_1	q_2	1	q_0	1
q_2	q_3	0	q_1	1
q_3	q_3	1	q_1	1



FINITE AUTOMATA WITH OUTPUT

Automata Summary

	Finite Automata	Tr. Graphs	Generalized Tr. Graphs	non-determ automata	Moore Machines	Mealy Machines
start states	1	≥ 1	≥ 1	1	1	1
final states	≥ 0	≥ 0	≥ 0	≥ 0	0	0
labels on arrows	letters of Σ	words of Σ^*	Reg. Expr. over Σ	letters of Σ	letters of Σ	i/o, $i \in \Sigma$, $o \in \Gamma$
# of trans. from each state	1 transition for each letter of Σ	≥ 0	≥ 0	≥ 0	1 transition for each letter of Σ	1 transition for each letter of Σ
Deterministic?	Yes	No	No	No	Yes	Yes
Output?	No	No	No	No	Yes	Yes