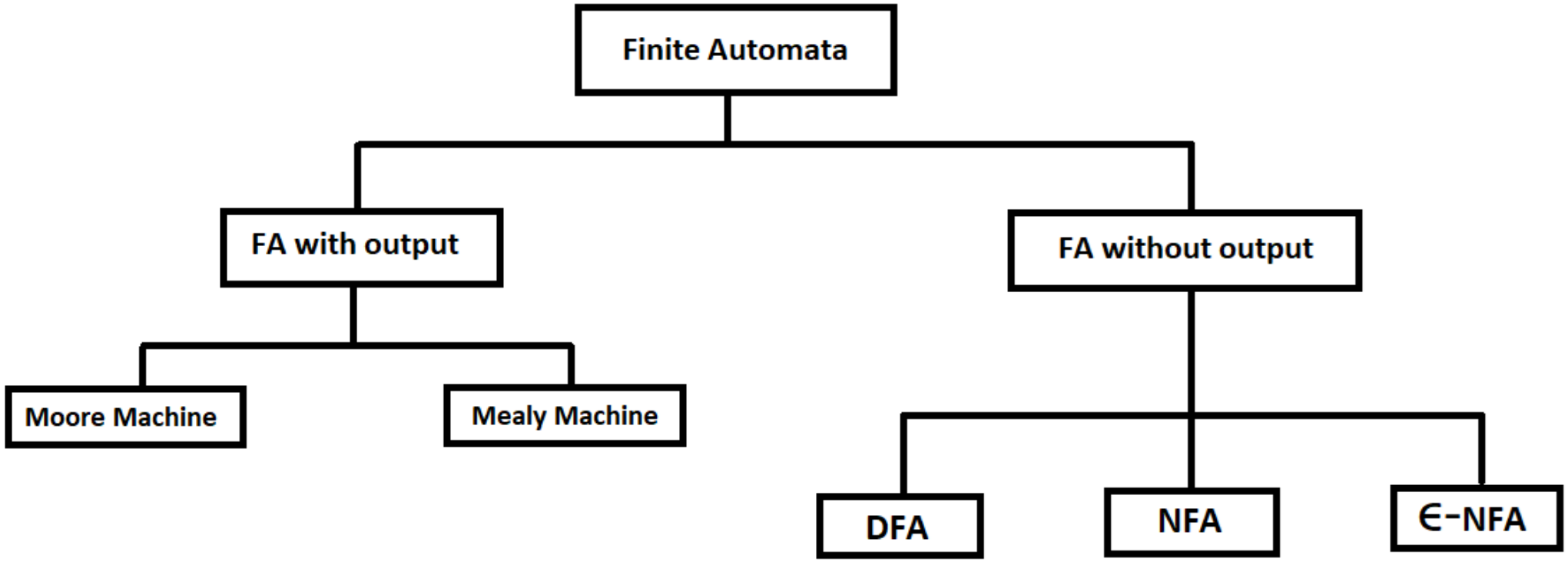
FINITE STATE MACHINE
(Finite Automata)

FINITE STATE MACHINE

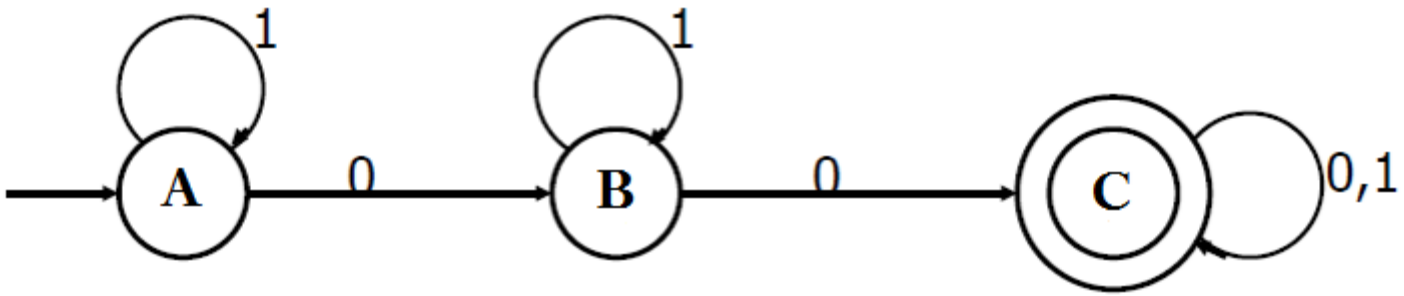


DETERMINISTIC FINITE AUTOMATA (DFA)

- it is a finite-state machine that accepts and rejects strings of symbols and only produces a unique computation (or run) of the automaton for each input string
- it is the simplest model of computation
- it has a very limited memory

DETERMINISTIC FINITE AUTOMATA (DFA)

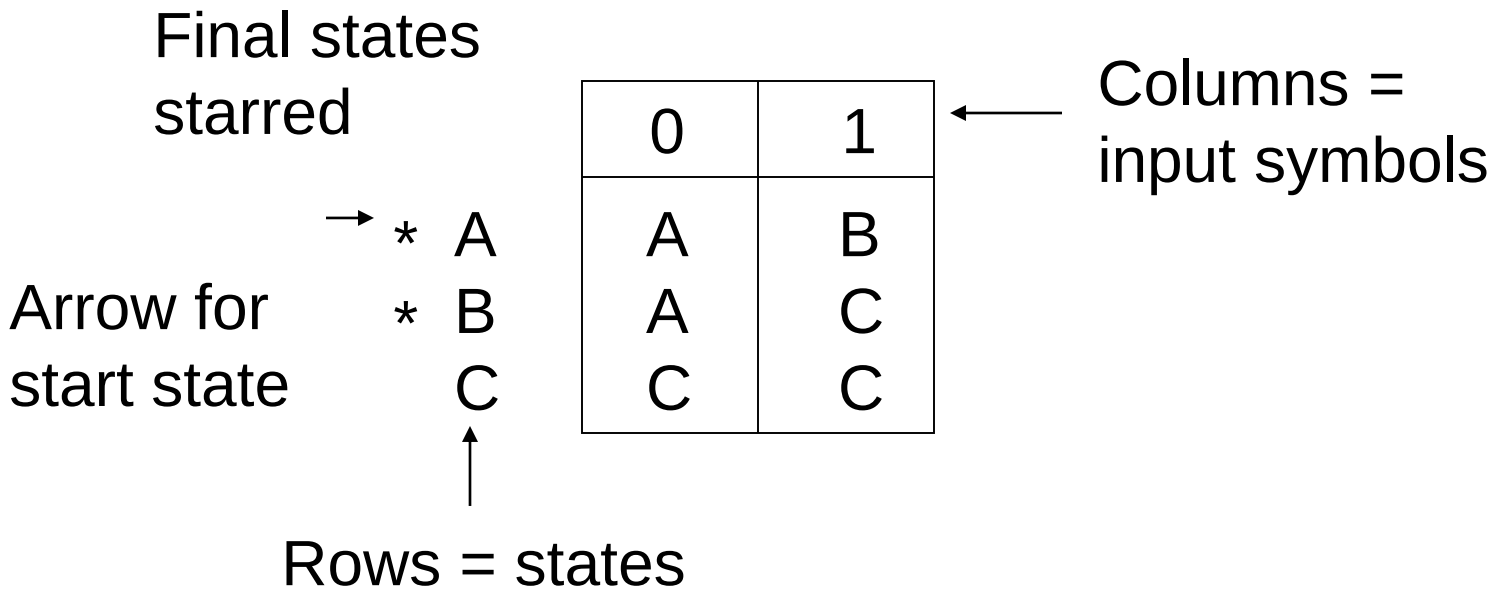
- DFAs are easiest to present pictorially: Such a graph is called a state transition diagram



FORMAL DEFINITION

- A deterministic finite automaton M is a 5-tuple, $(Q, \Sigma, \delta, q_0, F)$, consisting of:
  - Q = set of all states
  - Σ = input symbols
  - δ = transition function $(Q \times \Sigma \rightarrow Q)$
  - q0 = start state / initial state
  - F = set of final states

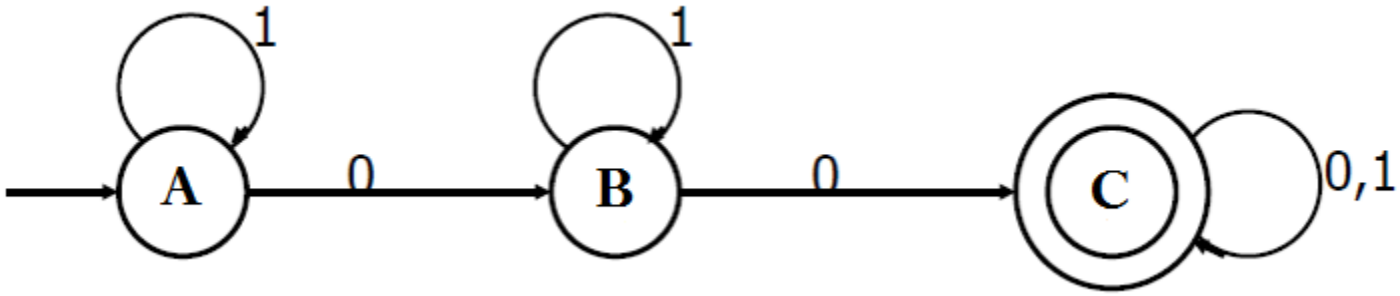
ALTERNATIVE REPRESENTATION: TRANSITION TABLE



	0	1
→ * A	A	B
* B	A	C
C	C	C

EXAMPLE 1

- **From the given DFA diagram, describe a DFA accepting the language by:**
  - a.) determining the 5-tuple
  - b.) building a DFA transition table



- **5-tuple**
  - Q = {A,B,C}
  - Σ = {0,1}
  - δ = {A × 0 → B; A × 1 → A; B × 0 → C; B × 1 → B; C × 0 → C; C × 1 → C}
  - q0 = A
  - F = C

- **Transition Table**

		0	1
→	A	B	A
	B	C	B
*	C	C	C

EXAMPLE 2

- **For each of the following languages, describe a DFA accepting the language by drawing a DFA diagram**
  - L1 = set of all strings over {0,1} that starts with 0
  - L2 = set of all strings over {0,1} of length 2