Regular Expressions and DFAs

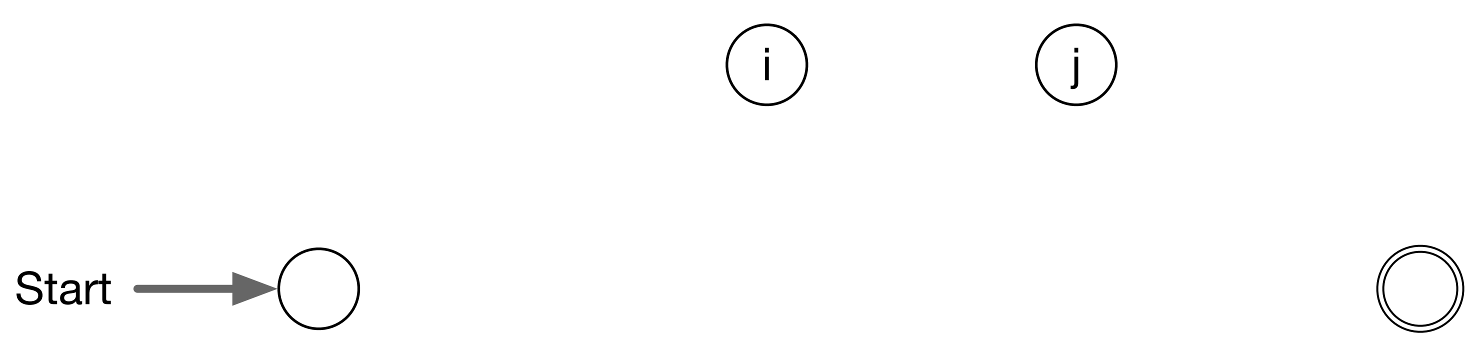
Thomas Schwarz, SJ

Regular Expressions and Deterministic Finite Automata

- Now, we need to show that every language accepted by a deterministic finite automaton is regular.
  - Given a DFA $M = (\{q_1, \ldots, q_n\}, \Sigma, \delta, q_1, F)$
  - Define $R^k_{i,j}$ as Set of strings that go from State i to State j without going through any state numbered higher than *k*
    - We can define $R^k_{i,j}$ by recursion, as we will show
      - $R^0_{i,i} = \{a \mid \delta(q_i, a) = q_i\} \cup \{\epsilon\}$
      - $R^0_{i,j} = \{a \mid \delta(q_i, a) = q_j\}$ if $i \neq j$
      - $R^k_{i,j} = R^{k-1}_{i,j} \; + \; R^{k-1}_{i,k} \cdot \left(R^{k-1}_{k,k}\right)^+ \cdot R^{k-1}_{k,j}$

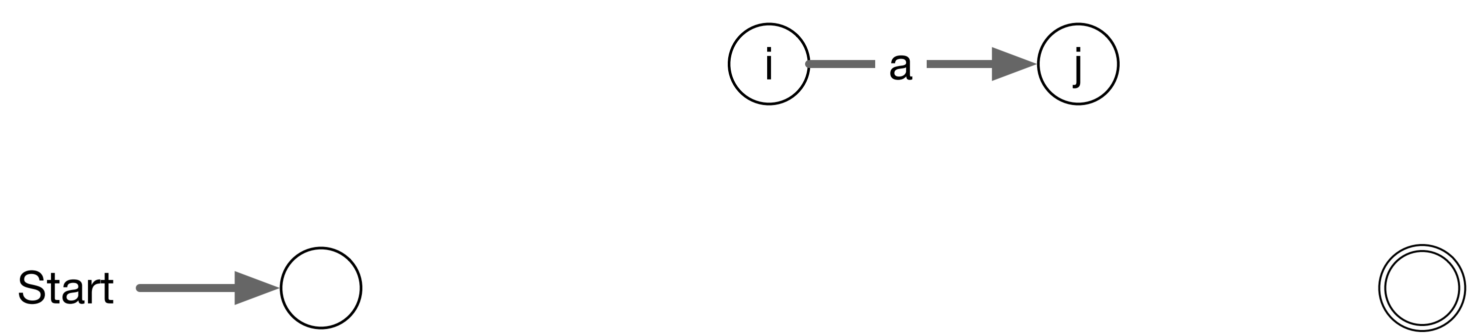
Regular Expressions and Deterministic Finite Automata

- Observation: $R^k_{i,j}$ is given by a regular expression
    - Proof by induction on *k*
        - Base: *k* = 0
            - First case $i \neq j$:
                - $R^0_{i,j}$ is the set of strings accepted by going from State i to State j without going through any other State
                - If there is no transition: $R^0_{i,j} = \emptyset$.



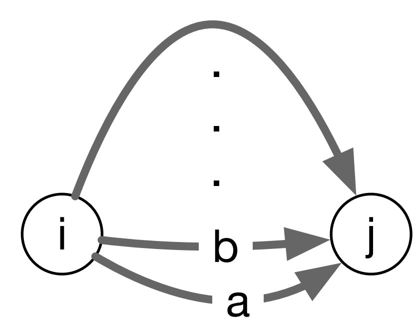
Regular Expressions and Deterministic Finite Automata

- Observation: $R_{i,j}^{k}$ is given by a regular expression
  - Proof by induction on *k*
    - Base: $k = 0$
      - First case $i \neq j$:
        - $R_{i,j}^{0}$ is the set of strings accepted by going from State i to State j without going through any other State
        - If there is a transition: $R_{i,j}^{0} = \mathbf{a}$.



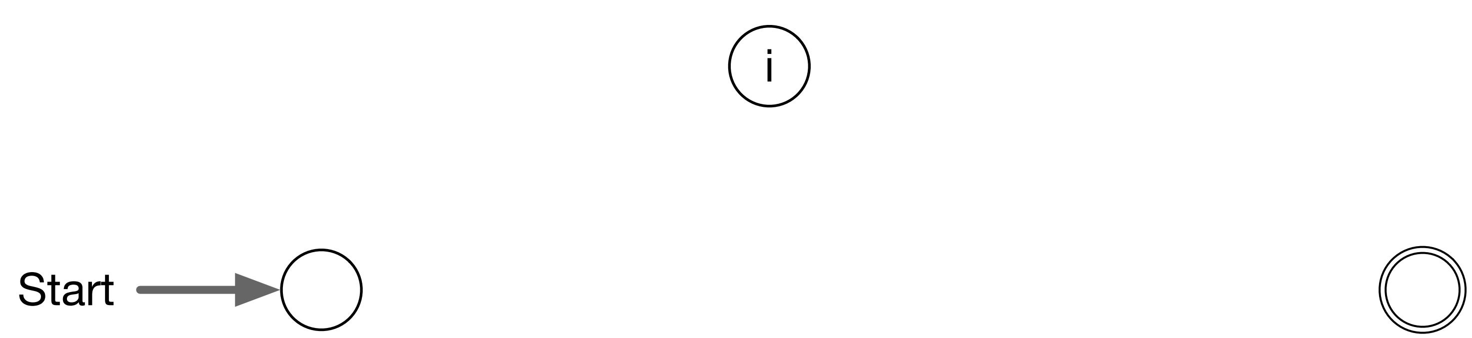
Regular Expressions and Deterministic Finite Automata

- Observation: $R^k_{i,j}$ is given by a regular expression
  - Proof by induction on *k*
    - Base: *k* = 0
      - First case $i \neq j$:
        - $R^0_{i,j}$ is the set of strings accepted by going from State i to State j without going through any other State
        - If there are more transitions: $R^0_{i,j} = \mathbf{a} + \mathbf{b} + \ldots.$



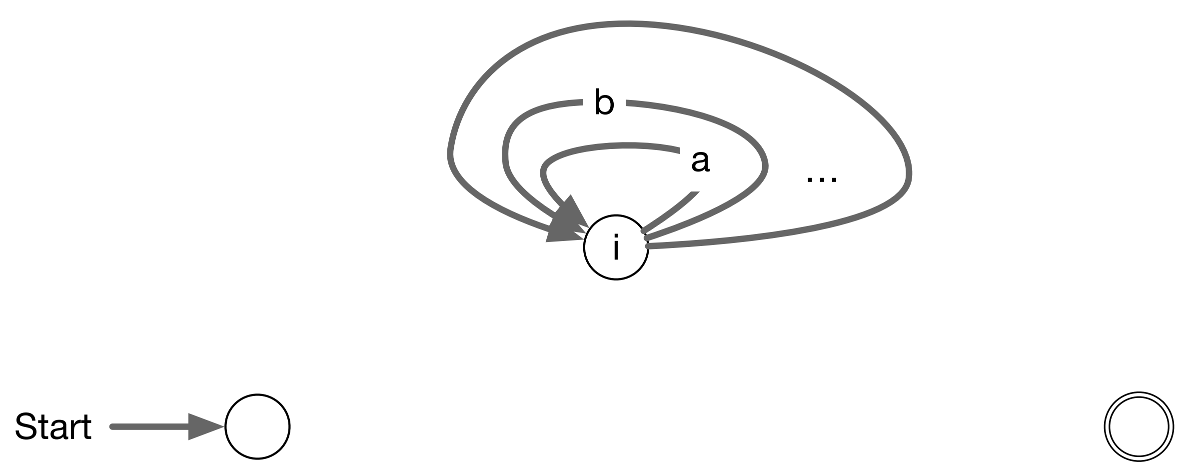
Regular Expressions and Deterministic Finite Automata

- Observation: $R^k_{i,j}$ is given by a regular expression
  - Proof by induction on *k*
    - Base: *k* = 0
      - Second case $i = j$:
        - $R^0_{i,i}$ is the set of strings accepted by going from State i to State j without going through any other State
        - If there are no transitions: $R^0_{i,i} = \epsilon$



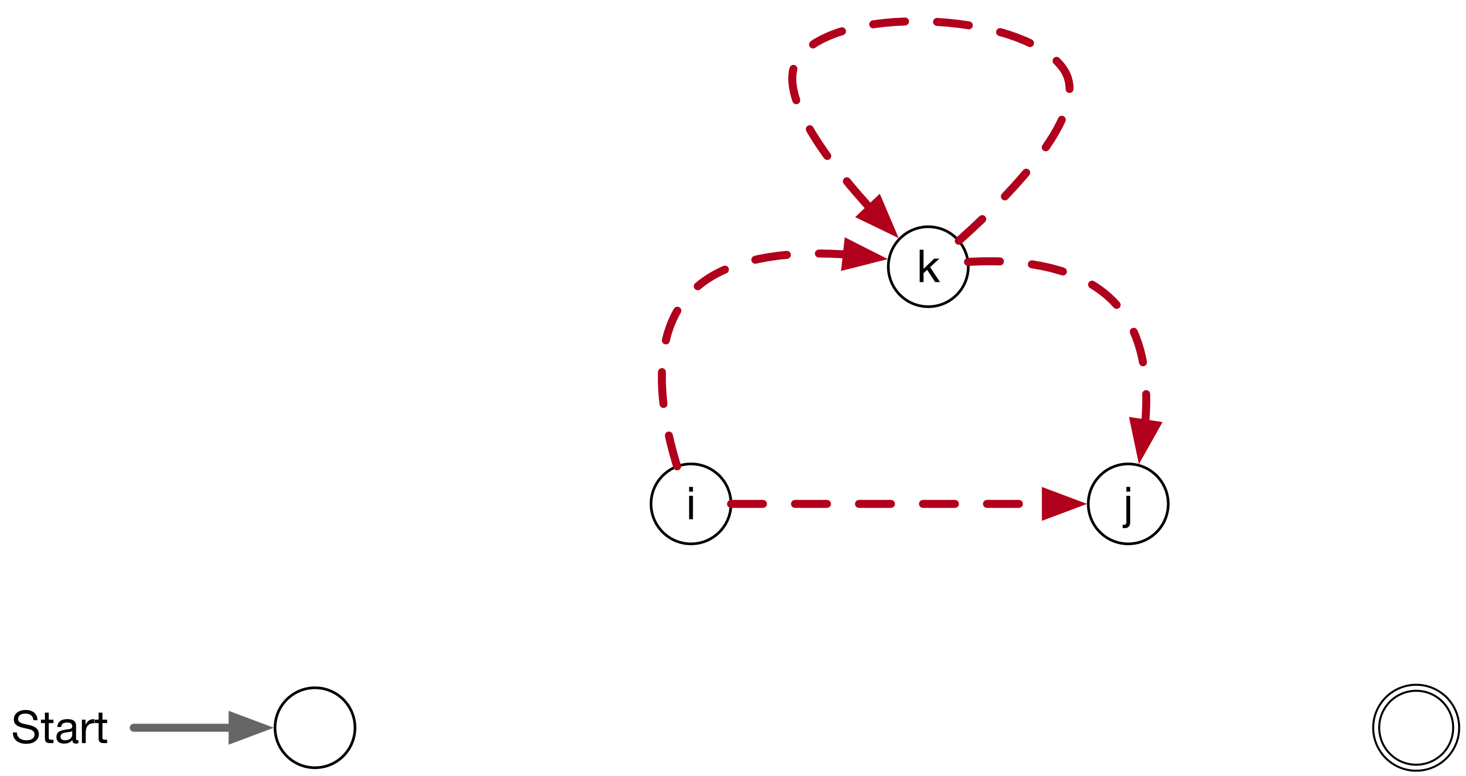
Regular Expressions and Deterministic Finite Automata

- Observation: $R_{i,j}^{k}$ is given by a regular expression
  - Proof by induction on *k*
    - Base: *k* = 0
      - Second case $i = j$:
        - $R_{i,i}^{0}$ is the set of strings accepted by going from State i to State j without going through any other State
        - If there are self-transitions: $R_{i,i}^{0} = \epsilon + \mathbf{a} + \mathbf{b} + \dots$



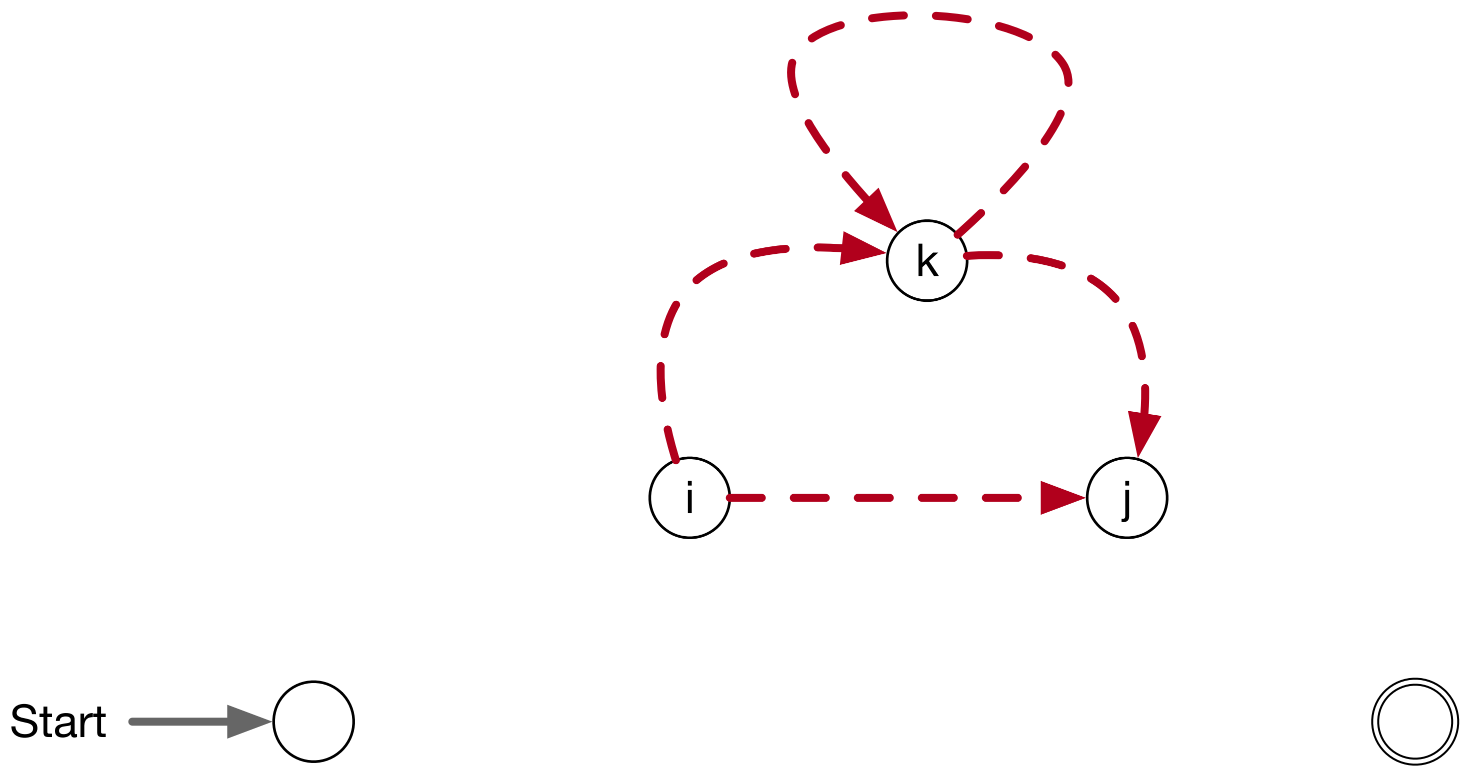
Regular Expressions and Deterministic Finite Automata

- Observation: $R_{i,j}^{k}$ is given by a regular expression
  - Proof by induction on *k*
  - Induction step: *k* —> *k*+1
    - How can we get from State *i* to State *j*



Regular Expressions and Deterministic Finite Automata

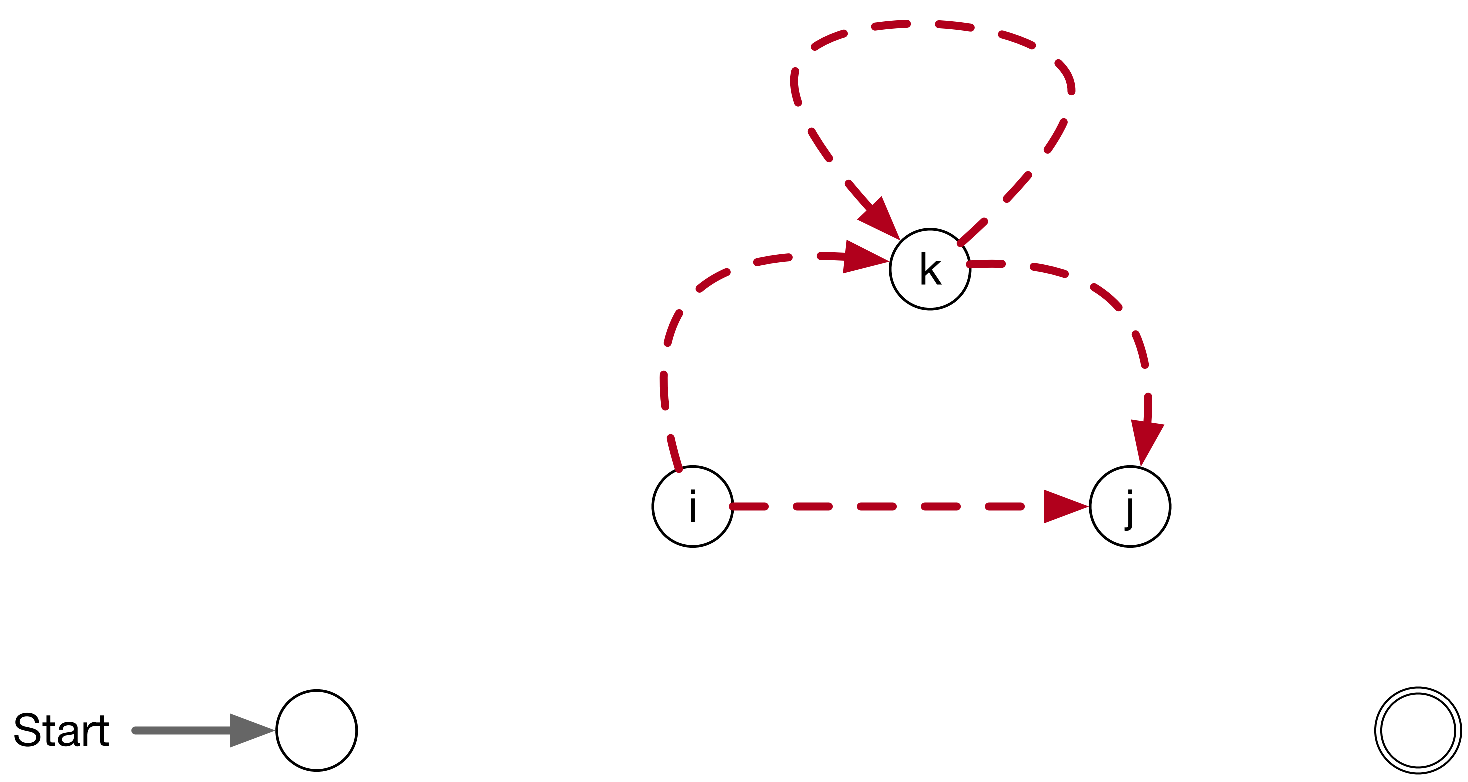
- How can we get from State *i* to State *j* ?



- Can go without touching *k*
- Can go to *k* without touching *k*, then zero, once, or many times from *k* to *k* without touching *k* in between, followed by going from *k* to *j* without touching *k*

Regular Expressions and Deterministic Finite Automata

- How can we get from State *i* to State *j* ?

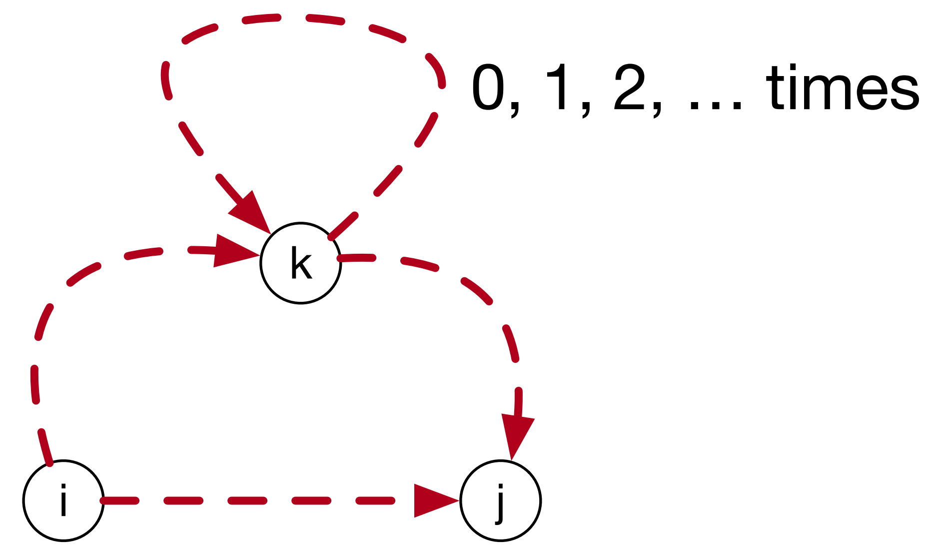


- We read off:

$$R^k_{i,j} = R^{k-1}_{i,j} + R^{k-1}_{i,k} \cdot R^{k-1}_{k,j} + R^{k-1}_{i,k} \cdot R^{k-1}_{k,k} \cdot R^{k-1}_{k,j} + R^{k-1}_{i,k} \cdot R^{k-1}_{k,k} \cdot R^{k-1}_{k,k} \cdot R^{k-1}_{k,j} + \ldots$$

Regular Expressions and Deterministic Finite Automata

- How can we get from State *i* to State *j* ?



- $R^{k}_{i,j} = R^{k-1}_{i,j} + R^{k-1}_{i,k} \cdot (R^{k-1}_{k,k})^{*} \cdot R^{k-1}_{k,j}$

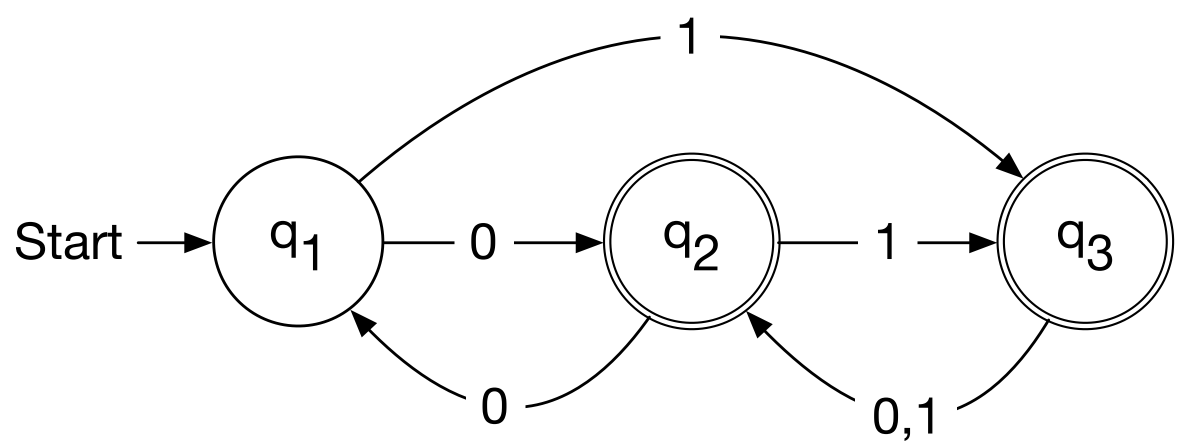
Regular Expressions and Deterministic Finite Automata

- It follows that the language accepted by a DFA is regular:
  - A string is accepted if it moves from the initial state to a final state

$$\mathscr{L}(M) = \cup_{q_j \in \mathscr{F}} R_{1,j}^{n+1} = \sum_{q_j \in \mathscr{F}} \mathbf{r}_{1,j}^{n+1}$$

Regular Expressions and Deterministic Finite Automata

- Example:



$$r_{1,1}^{1} = \epsilon$$

$$r_{1,2}^{1} = \mathbf{0}$$

$$r_{1,3}^{0} = \mathbf{1}$$

$$r_{2,1}^{1} = \mathbf{0}$$

$$r_{2,2}^{1} = \epsilon$$

$$r_{2,3}^{1} = \mathbf{1}$$

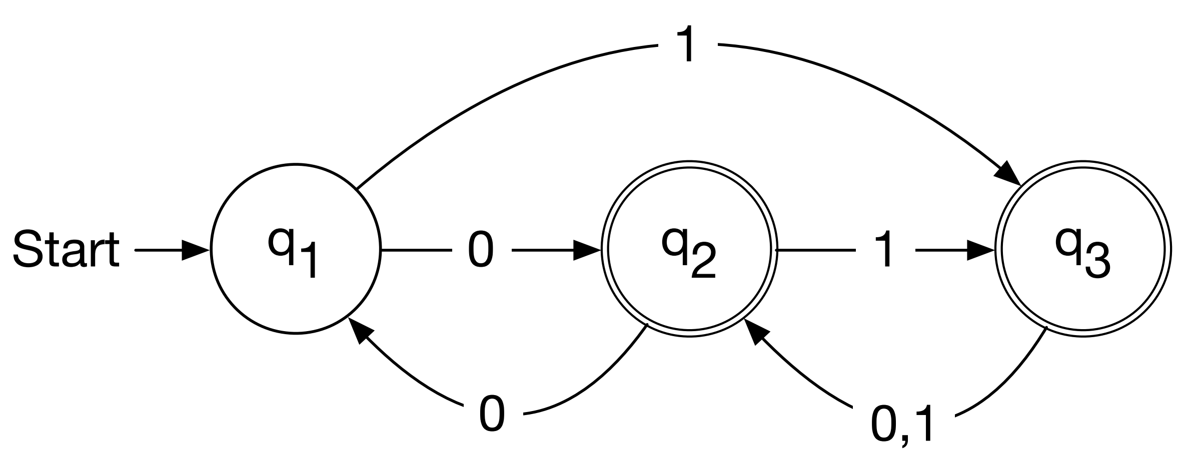
$$r_{3,1}^{1} = \emptyset$$

$$r_{3,2}^{1} = \mathbf{0} + \mathbf{1}$$

$$r_{3,3}^{1} = \emptyset$$

Regular Expressions and Deterministic Finite Automata

- Example:



$$r_{1,1}^{2} = \epsilon$$

$$r_{1,2}^{1} = \mathbf{0}$$

$$r_{1,3}^{2} = \mathbf{1}$$

$$r_{2,1}^{2} = \mathbf{0}$$

$$r_{2,2}^{2} = \epsilon + \mathbf{0}\epsilon^{*}\mathbf{0} = \epsilon + \mathbf{00}$$

$$r_{2,3}^{2} = \mathbf{1} + \mathbf{0}\epsilon^{*}\mathbf{1} = \mathbf{1} + \mathbf{01}$$

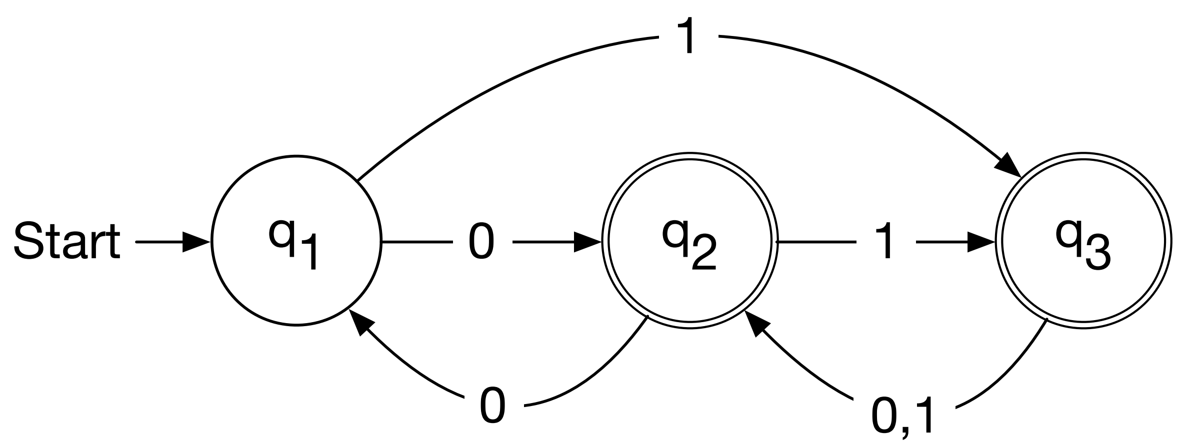
$$r_{3,1}^{2} = \emptyset$$

$$r_{3,2}^{2} = \mathbf{0} + \mathbf{1}$$

$$r_{3,3}^{2} = \epsilon$$

Regular Expressions and Deterministic Finite Automata

- Example:



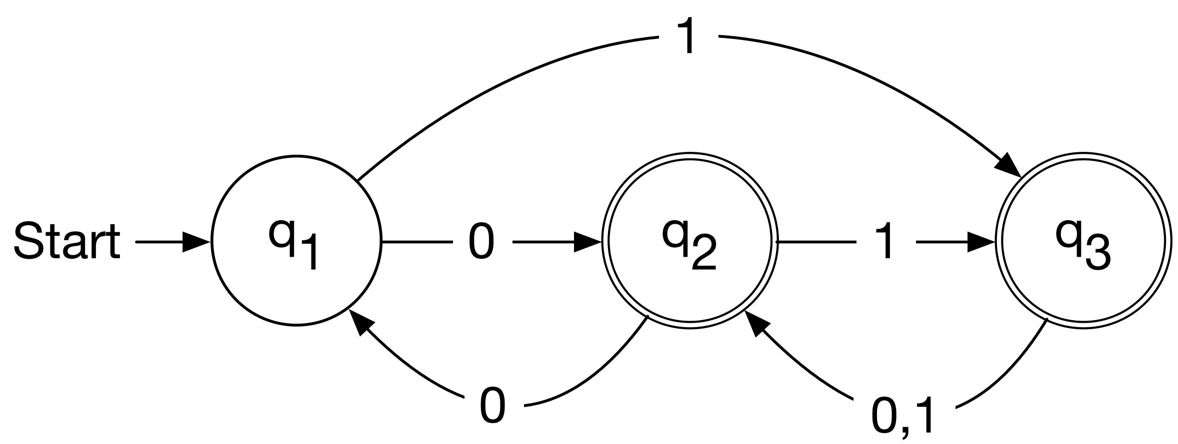
$$r_{1,1}^3 = r_{1,1}^2 + r_{1,2}^3 {r_{2,2}^3}^* r_{2,1}^2 = \epsilon + \mathbf{0}(\epsilon + \mathbf{00})^*\mathbf{0} = \epsilon + \mathbf{0}(\mathbf{00})^*\mathbf{0} = \epsilon + (\mathbf{00})^*\mathbf{0} = \mathbf{00}^*$$

$$r_{1,2}^3 = r_{1,2}^2 + r_{1,2}^2(r_{2,2}^2)^* r_{2,2}^2 = \mathbf{0} + \mathbf{0}(\mathbf{00})^*\epsilon = \mathbf{0}(\mathbf{00})^*$$

$$r_{1,3}^3 = r_{1,3}^2 + r_{1,2}^2(r_{2,2}^2)^* r_{2,3}^2 = \mathbf{1} + \mathbf{0}(\epsilon + \mathbf{00})^*\mathbf{1} = \mathbf{1} + \mathbf{0}(\mathbf{00})^*\mathbf{1}$$

Regular Expressions and Deterministic Finite Automata

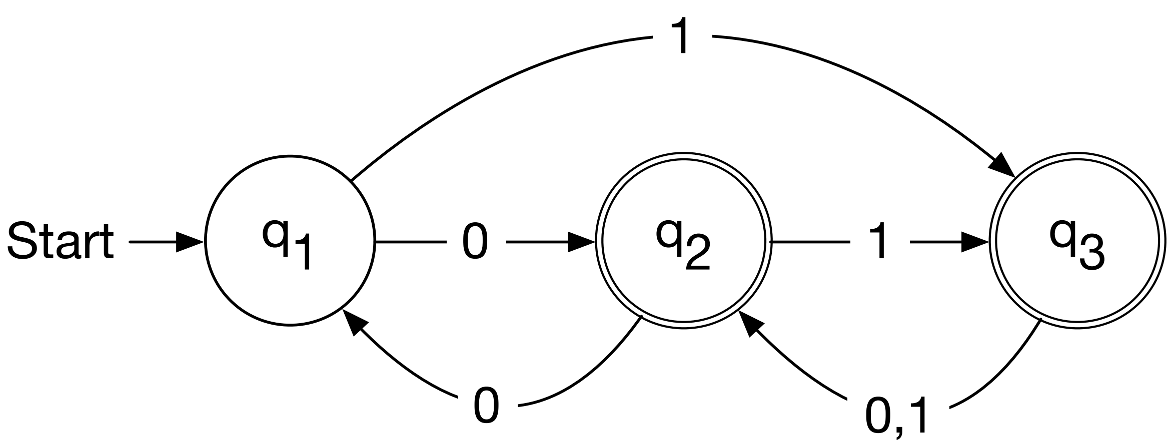
- Example:



$$r_{2,1}^{3} = r_{2,1}^{2} + r_{2,2}^{2}(r_{2,2}^{2})^{*}r_{2,1}^{2} = \mathbf{0} + (\mathbf{00})^{*}(\mathbf{00})^{*}\mathbf{0} = (\mathbf{00})^{*}\mathbf{0}$$

Regular Expressions and Deterministic Finite Automata

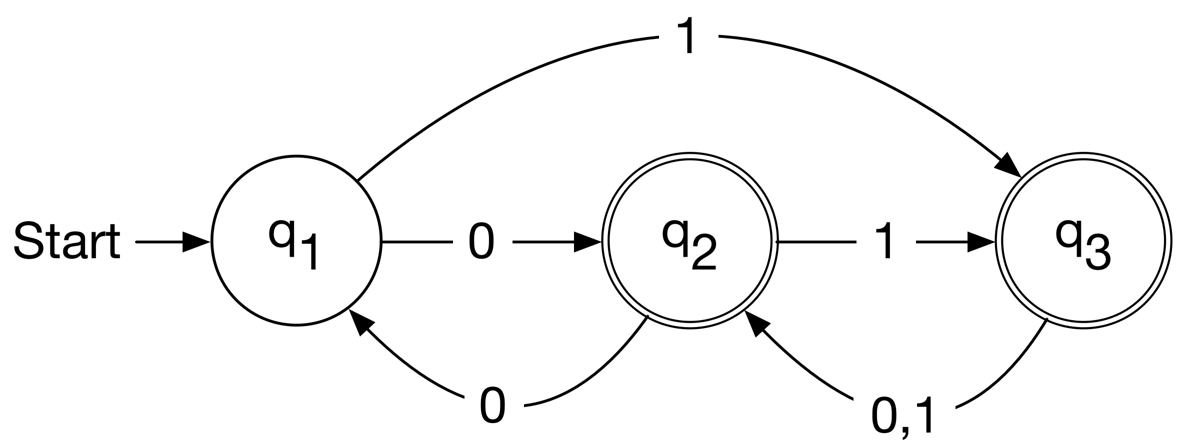
- Example:



$$r^3_{2,2} = r^2_{2,2} + r^2_{2,2}(r^2_{2,2})^* r^2_{2,2} = (\mathbf{00})^*$$

Regular Expressions and Deterministic Finite Automata

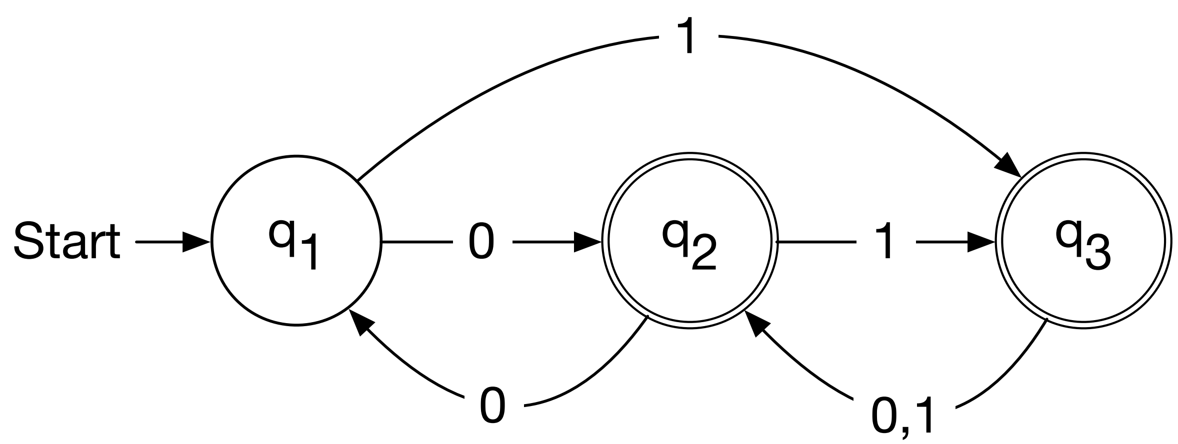
- Example:



$$r^3_{2,3} = r^2_{2,3} + r^2_{2,2}(r^2_{2,2})^* r^2_{2,3} = (\mathbf{1} + \mathbf{01}) + \mathbf{00}(\mathbf{00})^*(\mathbf{1} + \mathbf{01}) = = \mathbf{1} + \mathbf{01} + (\mathbf{00})^+\mathbf{1} + (\mathbf{00})^+\mathbf{01}$$

Regular Expressions and Deterministic Finite Automata

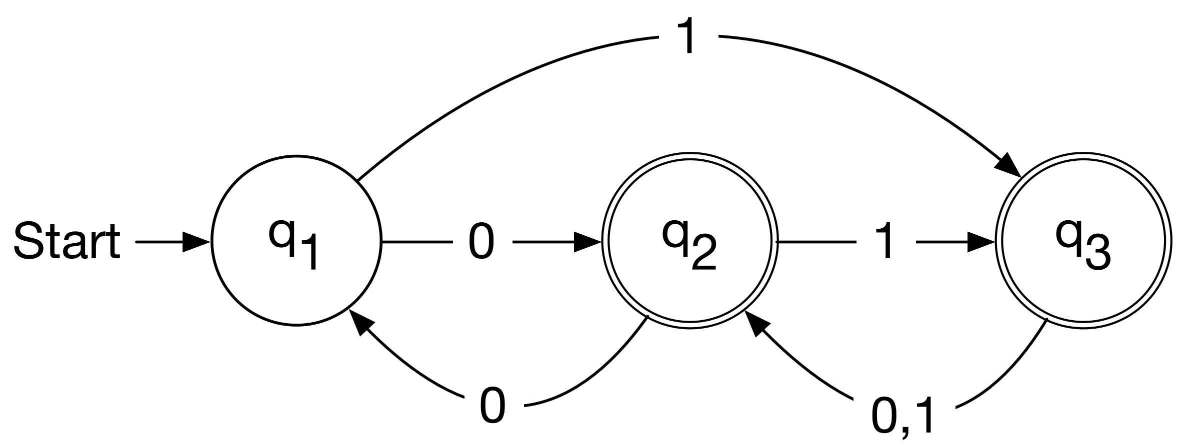
- Example:



$$r^3_{3,1} = r^2_{3,1} + r^2_{3,2}(r^2_{2,2})^* r^2_{2,1} = \varnothing + (\mathbf{0} + \mathbf{1})(\mathbf{00})^*\mathbf{0} = (\mathbf{00})^+ + \mathbf{1}(\mathbf{00})^*\mathbf{0}$$

Regular Expressions and Deterministic Finite Automata

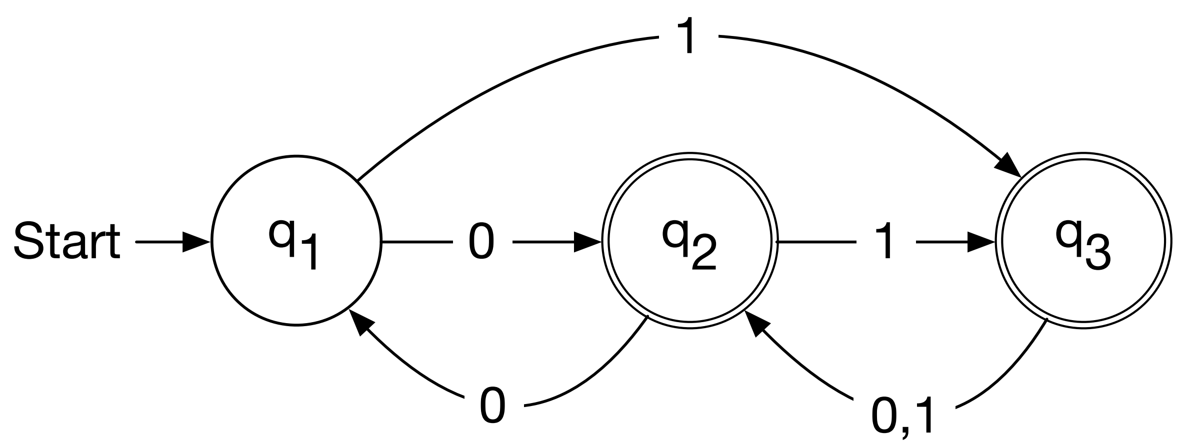
- Example:



$$r^3_{3,2} = r^2_{3,2} + r^2_{3,2}(r^2_{2,2})^* r^2_{2,2} = (\mathbf{0} + \mathbf{1}) + (\mathbf{0} + \mathbf{1})(\mathbf{00})^*\mathbf{00} = \mathbf{0}(\mathbf{00})^* + \mathbf{1}(\mathbf{00})^*$$

Regular Expressions and Deterministic Finite Automata

- Example:



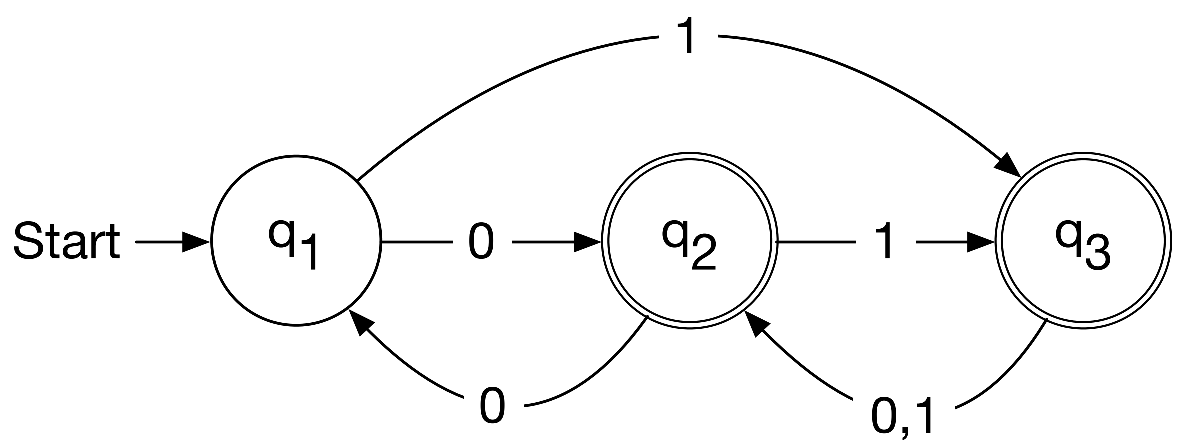
$$r^3_{3,3} = r^2_{3,3} + r^2_{3,2}(r^2_{2,2})^* r^2_{2,3} = \epsilon + (\mathbf{0} + \mathbf{1})(\mathbf{00})^*(\mathbf{1} + \mathbf{01})$$

$$= \epsilon + \mathbf{0}(\mathbf{00})^*\mathbf{1} + \mathbf{1}(\mathbf{00})^*\mathbf{1} + \mathbf{0}(\mathbf{00})^*\mathbf{01} + \mathbf{1}(\mathbf{00})^*\mathbf{01}$$

$$= \epsilon + \mathbf{0}(\mathbf{00})^*\mathbf{1} + \mathbf{1}(\mathbf{00})^*\mathbf{1} + (\mathbf{00})^+\mathbf{1} + \mathbf{1}(\mathbf{00})^*\mathbf{01}$$

Regular Expressions and Deterministic Finite Automata

- Example:



$$\mathscr{L}(M) = r^4_{1,2} + r^4_{1,3} = r^3_{1,2} + r^3_{1,3}(r^3_{3,3})^* r^3_{3,2} + r^3_{1,3} + r^3_{1,3}(r^3_{3,3})^* r_{3,3}$$

$$= \mathbf{0(00)^* + (1 + 0(00)^*1)\left(\epsilon + 0(00)^*1 + 1(00)^*1 + (00)^+1 + 1(00)^*01\right)^* 0(00)^* + 1(00)^*}$$

$$\mathbf{+1 + 0(00)^*1 + \left(1 + 0(00)^*1\right) \left(\epsilon + 0(00)^*1 + 1(00)^*1 + (00)^+1 + 1(00)^*01\right)^* 0(00)^* + 1(00)^*}$$

Finite Automata with Output

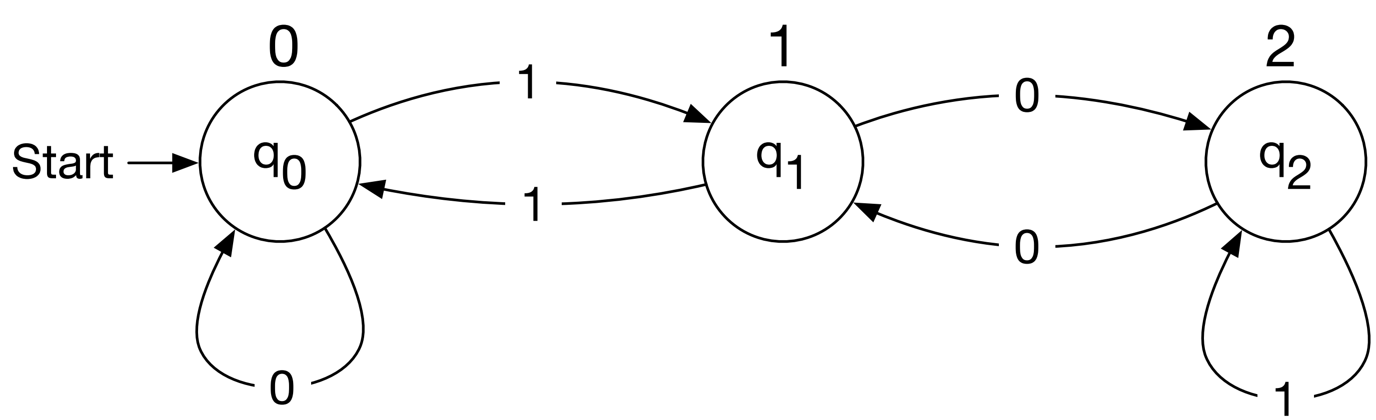
- Moore machines
  - Whenever the machine is in state *i* it outputs a symbol depending on the state

    - Example:
      - A Moore machine that calculates the remainder modulo 3 of a binary number
      - To derive the formula, consider

$$\begin{aligned} a.x \pmod 3 &\equiv 2a + x \pmod 3 \\ &\equiv \Big(2a \pmod 3\Big) + \Big(x \pmod 3\Big) \\ &\equiv 2\Big(a \pmod 3\Big) + \Big(x \pmod 3\Big) \end{aligned}$$

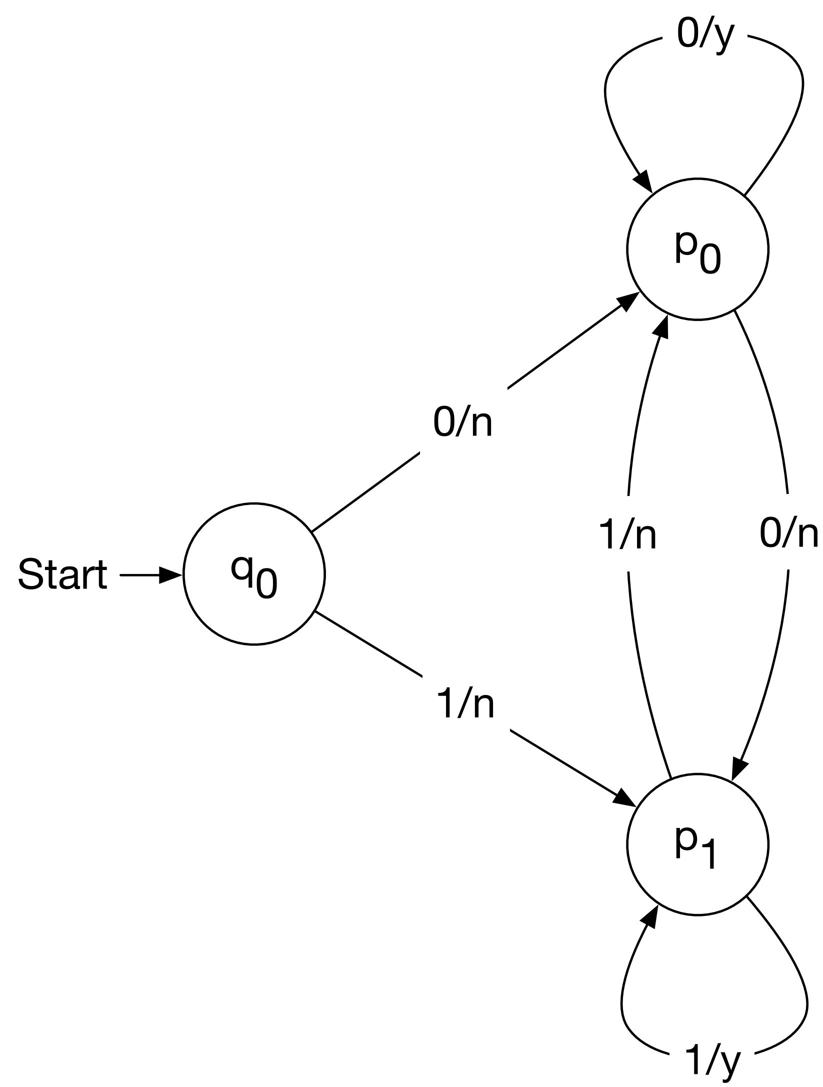
Finite Automata with Output

$a \pmod 3$	$x \pmod 3$	$a.x \pmod 3$
0	0	0
0	1	1
1	0	2
1	1	0
2	0	1
2	1	2



Finite Automata with Output

- Mealy Machines
  - Output depends on the current state and the transition



Finite Automata with Output

- It can be shown that Mealy and Moore machines are equivalent