CS 228 : Logic in Computer Science

Krishna. S

Finite State Machines

A deterministic finite state automaton (DFA) $A = (Q, \Sigma, \delta, q_0, F)$

- Q is a finite set of states
- Σ is a finite alphabet
- $\delta : Q \times \Sigma \to Q$ is the transition function
- $q_0 \in Q$ is the initial state
- $F \subseteq Q$ is the set of final states
- $L(A)$=all words leading from q_0 to some $f \in F$

Languages, Machines and Logic

A language $L \subseteq \Sigma^*$ is called regular iff there exists some DFA A such that $L = L(A)$.
A language $L \subseteq \Sigma^*$ is called FO-definable iff there exists an FO formula φ such that $L = L(\varphi)$.

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Begins with *a*, ends with *b*, and has a pair of consecutive *a*'s
- Contains a *b* and ends with *aa*
- Contains *abb*
- There are two occurrences of *b* between which only *a*'s occur
- Right before the last position is an *a*
- Even length words

Is it Regular? Is it FO-definable?

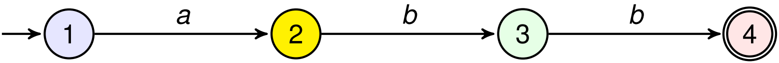
$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*

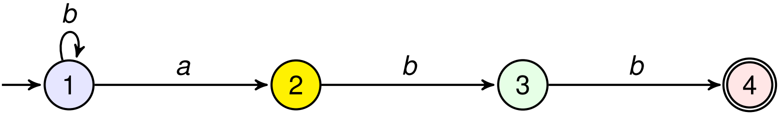


$$\exists x \exists y \exists z (Q_a(x) \wedge Q_b(y) \wedge Q_b(z) \wedge S(x, y) \wedge S(y, z))$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*

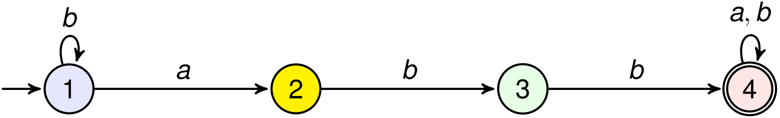


$$\exists x \exists y \exists z (Q_a(x) \wedge Q_b(y) \wedge Q_b(z) \wedge S(x, y) \wedge S(y, z))$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*

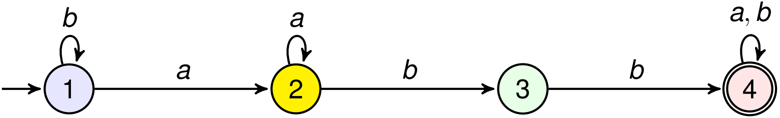


$$\exists x \exists y \exists z (Q_a(x) \wedge Q_b(y) \wedge Q_b(z) \wedge S(x, y) \wedge S(y, z))$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*

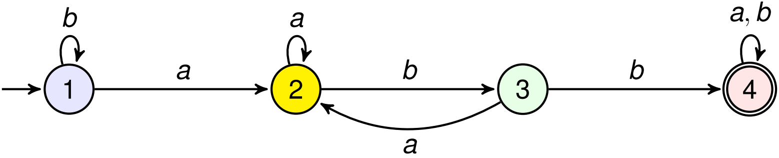


$$\exists x \exists y \exists z (Q_a(x) \wedge Q_b(y) \wedge Q_b(z) \wedge S(x, y) \wedge S(y, z))$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

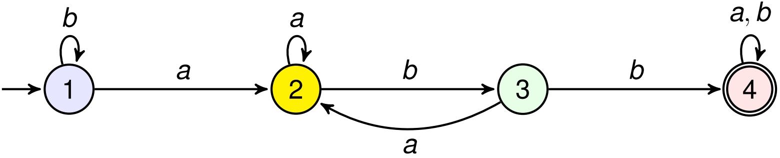
- Contains *abb*



Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Contains *abb*



$$\exists x \exists y \exists z (Q_a(x) \wedge Q_b(y) \wedge Q_b(z) \wedge S(x, y) \wedge S(y, z))$$

Is it Regular? Is it FO-definable?

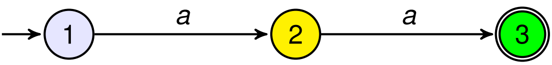
$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Right before the last position is an a :

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Right before the last position is an a :
  Examples : $ab, babbaa, bbab$
  Non examples : ba, bb, aba

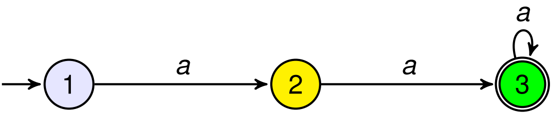


$$\exists x [Q_a(x) \wedge \exists y (S(x, y) \wedge \forall z (z \leqslant y))]$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Right before the last position is an a :
  Examples : $ab, babbaa, bbab$
  Non examples : ba, bb, aba

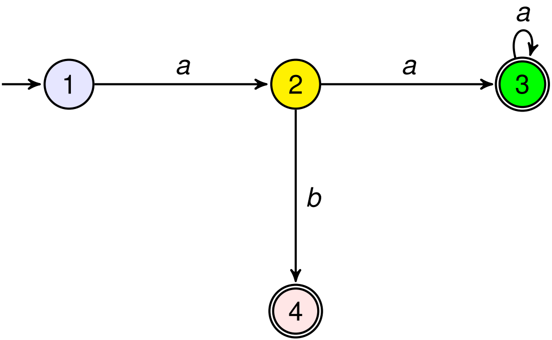


$$\exists x[Q_a(x) \wedge \exists y(S(x, y) \wedge \forall z(z \leqslant y))]$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Right before the last position is an a :
  Examples : ab, $babbaa$, $bbab$
  Non examples : ba, bb, aba

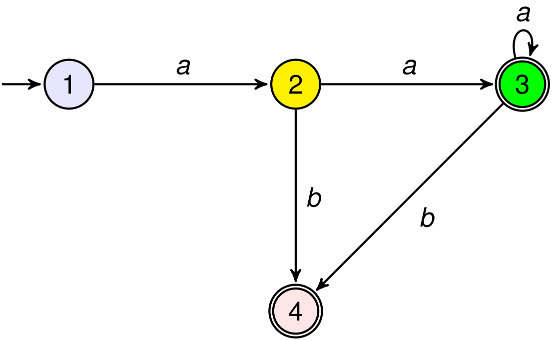


$$\exists x[Q_a(x) \wedge \exists y(S(x, y) \wedge \forall z(z \leqslant y))]$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Right before the last position is an a :
  Examples : ab, $babbaa$, $bbab$
  Non examples : ba, bb, aba

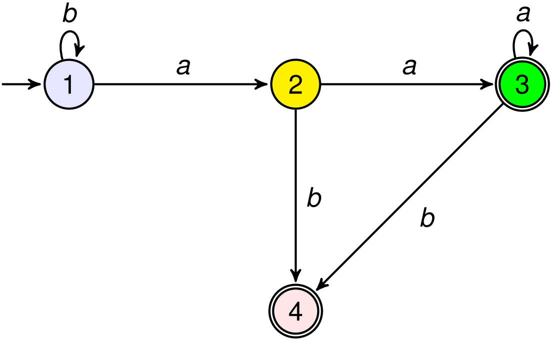


$$\exists x[Q_a(x) \wedge \exists y(S(x, y) \wedge \forall z(z \leqslant y))]$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Right before the last position is an a :
  Examples : ab, $babbaa$, $bbab$
  Non examples : ba, bb, aba

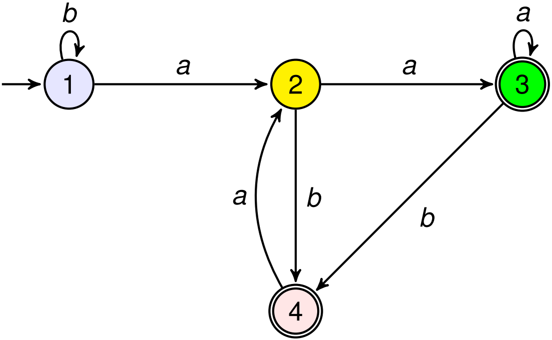


$$\exists x[Q_a(x) \wedge \exists y(S(x,y) \wedge \forall z(z \leqslant y))]$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Right before the last position is an a :
  Examples : ab, $babbaa$, $bbab$
  Non examples : ba, bb, aba

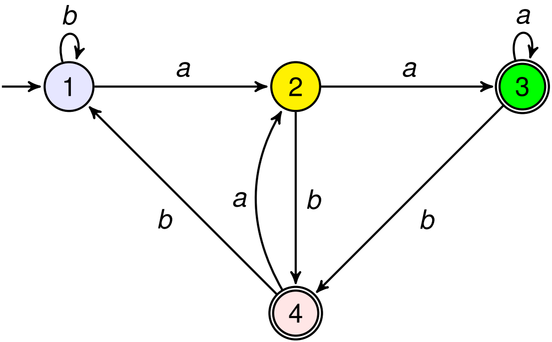


$$\exists x[Q_a(x) \wedge \exists y(S(x, y) \wedge \forall z(z \leqslant y))]$$

Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

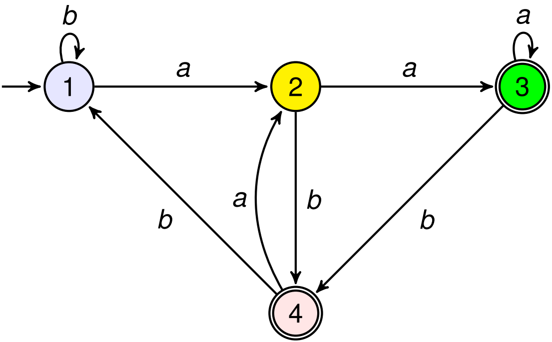
- Right before the last position is an a :
  Examples : ab, $babbaa$, $bbab$
  Non examples : ba, bb, aba



Is it Regular? Is it FO-definable?

$\Sigma = \{a, b\}$. Consider the following languages $L \subseteq \Sigma^*$:

- Right before the last position is an a :
  Examples : $ab, babbaa, bbab$
  Non examples : ba, bb, aba



$$\exists x[Q_a(x) \wedge \exists y(S(x, y) \wedge \forall z(z \leqslant y))]$$

Deterministic Finite Automata

- Every state on every symbol goes to a unique state
  - $\delta : Q \times \Sigma \to Q$ is a transition function

Any letter you put on the state changes the state.

- Given a string $w \in \Sigma^*$ and a state $q \in Q$, iteratively apply δ
  - $w = aab$
  - $\delta(q, a) = q_1$,

Deterministic Finite Automata

- Every state on every symbol goes to a unique state
  - $\delta : Q \times \Sigma \to Q$ is a transition function
- Given a string $w \in \Sigma^*$ and a state $q \in Q$, iteratively apply δ
  - $w = aab$
  - $\delta(q, a) = q_1$, $\delta(\delta(q, a), a) = \delta(q_1, a) = q_2$,

Deterministic Finite Automata

- Every state on every symbol goes to a unique state
  - $\delta : Q \times \Sigma \to Q$ is a transition function
- Given a string $w \in \Sigma^*$ and a state $q \in Q$, iteratively apply δ
  - $w = aab$
  - $\delta(q, a) = q_1$, $\delta(\delta(q, a), a) = \delta(q_1, a) = q_2$,
    $\delta(\delta(\delta(q, a), a), b) =$

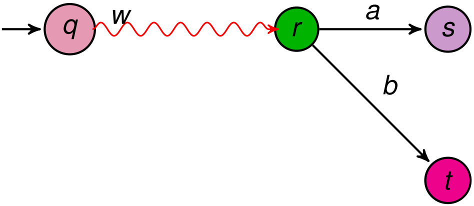
Deterministic Finite Automata

- Every state on every symbol goes to a unique state
  - $\delta : Q \times \Sigma \to Q$ is a transition function
- Given a string $w \in \Sigma^*$ and a state $q \in Q$, iteratively apply δ
  - $w = aab$
  - $\delta(q, a) = q_1$, $\delta(\delta(q, a), a) = \delta(q_1, a) = q_2$,
    $\delta(\delta(\delta(q, a), a), b) = \delta(\delta(q_1, a), b) =$

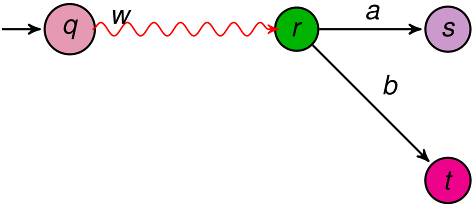
Deterministic Finite Automata

- Every state on every symbol goes to a unique state
  - $\delta : Q \times \Sigma \to Q$ is a transition function
- Given a string $w \in \Sigma^*$ and a state $q \in Q$, iteratively apply δ
  - $w = aab$
  - $\delta(q, a) = q_1$, $\delta(\delta(q, a), a) = \delta(q_1, a) = q_2$, $\delta(\delta(\delta(q, a), a), b) = \delta(\delta(q_1, a), b) = \delta(q_2, b) = q_3$
  - $\hat{\delta} : Q \times \Sigma^* \to Q$ extension of δ to strings
    - $\hat{\delta}(q, \epsilon) = q$
    - $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$

DFA : Transition Function on Words



DFA : Transition Function on Words



- $\hat{\delta}(q, wa) = s = \delta(\hat{\delta}(q, w), a) = \delta(r, a)$
- $\hat{\delta}(q, wb) = t = \delta(\hat{\delta}(q, w), b) = \delta(r, b)$

DFA Acceptance

- $w \in \Sigma^*$ is accepted iff $\hat{\delta}(q_0, w) \in F$
- $w \in \Sigma^*$ is rejected iff $\hat{\delta}(q_0, w) \notin F$
- Any string $w \in \Sigma^*$ is either accepted or rejected by a DFA A
- $L(A) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$
- $\Sigma^* = L(A) \cup \overline{L(A)}$

DFA States

- Each state is a bucket holding infinitely many words
- Thus we have good and bad buckets
- The buckets partition Σ^*
- Good buckets determine the language accepted by the DFA
- Words that land in bad buckets are not accepted by the DFA