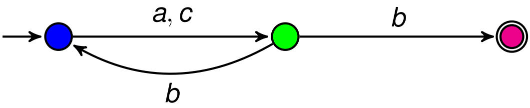
CS 228 : Logic in Computer Science

Krishna. S

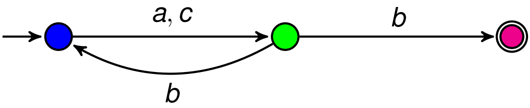
Recap : Idea for SAT checking

- Given FO formula φ over an alphabet Σ, construct an edge labeled graph G_φ : a graph whose edges are labeled by Σ.



Recap : Idea for SAT checking

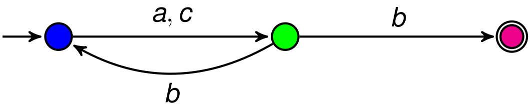
- Given FO formula φ over an alphabet Σ, construct an edge labeled graph G_φ : a graph whose edges are labeled by Σ.



- Each path in the graph gives rise to a word over Σ, obtained by reading off the labels on the edges

Recap : Idea for SAT checking

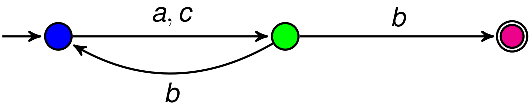
- Given FO formula φ over an alphabet Σ, construct an edge labeled graph G_φ : a graph whose edges are labeled by Σ.



- Each path in the graph gives rise to a word over Σ, obtained by reading off the labels on the edges
- G_φ has some special kinds of vertices
  - There is a unique vertex called the start vertex (blue vertex)
  - There are some vertices called good vertices (magenta vertex)

Recap : Idea for SAT checking

- Given FO formula φ over an alphabet Σ, construct an edge labeled graph G_φ : a graph whose edges are labeled by Σ.



- Each path in the graph gives rise to a word over Σ, obtained by reading off the labels on the edges
- G_φ has some special kinds of vertices
  - There is a unique vertex called the start vertex (blue vertex)
  - There are some vertices called good vertices (magenta vertex)
- Read off words on paths from the start vertex to any final vertex and call this set of words $L(G_\varphi)$
- Ensure that G_φ is constructed such that $L(\varphi) = L(G_\varphi)$.

Languages, Machines and Logic

A language $L \subseteq \Sigma^*$ is called regular iff there exists some DFA A such that $L = L(A)$.

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 $L = L(\varphi)$.

What we plan to show: L is FO-definable $\Rightarrow$ L is regular. Note that the converse is not true.

Deterministic Finite Automata

- Every state on every symbol goes to a unique state

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

Deterministic Finite Automata

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

Deterministic Finite Automata

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

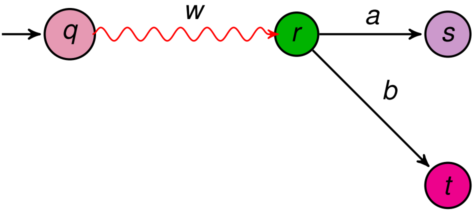
Deterministic Finite Automata

- 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
  - $w = aab$
  - $\delta(q, a) = q_1$, $\delta(\delta(q, a), a) = \delta(q_1, a) = q_2$,
    $\delta(\delta(\delta(q, a), a), b) = \delta(\delta(q_1, a), b) = \delta(q_2, b) = q_3$

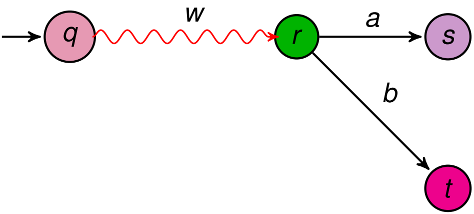
Deterministic Finite Automata

- 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 $\hat{\delta}(q_0, w) \notin F$

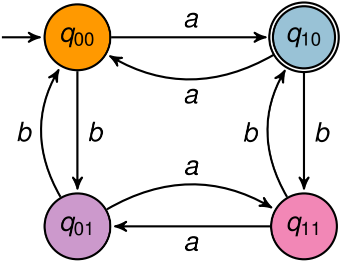
DFA Acceptance

- $w \in \Sigma^*$ 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

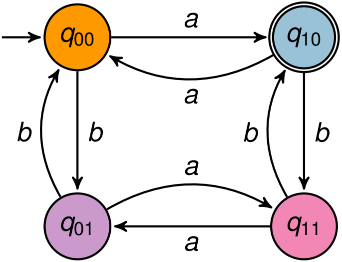
DFA Acceptance

- $w \in \Sigma^*$ 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
- $L(A) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$
- $\Sigma^* = L(A) \cup \overline{L(A)}$

Language Acceptance : Proof

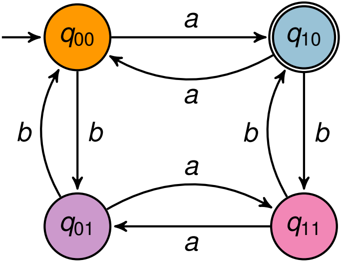


Language Acceptance : Proof



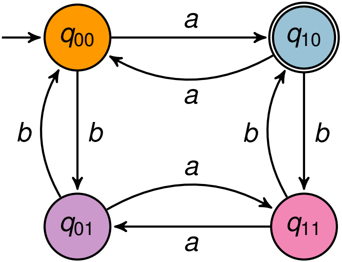
- $L = \{w \in \{a, b\}^* \mid |w|_a \text{ is odd and } |w|_b \text{ is even}\}$

Language Acceptance : Proof



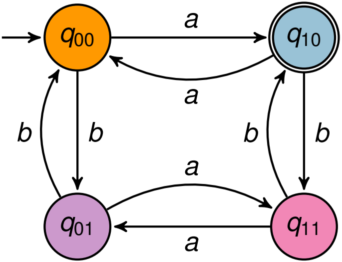
- $L = \{w \in \{a, b\}^* \mid |w|_a \text{ is odd and } |w|_b \text{ is even}\}$
- Show that for any $w \in \Sigma^*$,
  - $\hat{\delta}(q_{00}, w) = q_{ij}$ with $i, j \in \{0, 1\}$, parity of i same as $|w|_a$ and parity of j same as $|w|_b$

Language Acceptance : Proof



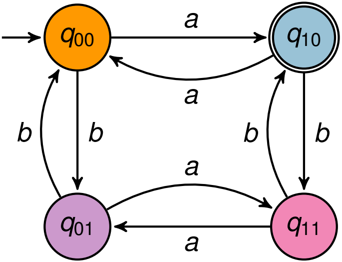
- Prove by induction on $|w|$

Language Acceptance : Proof



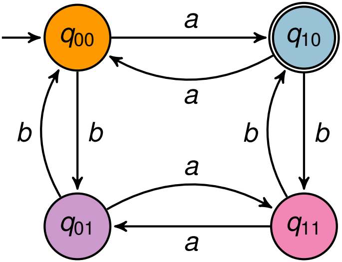
- Prove by induction on $|w|$
- Base case : For $|w| = \epsilon$, $\hat{\delta}(q_{00}, \epsilon) = q_{00}$

Language Acceptance : Proof



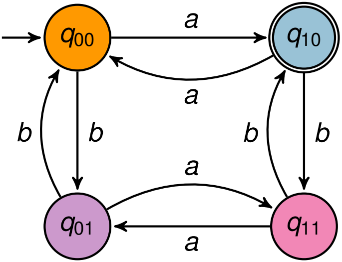
- Prove by induction on $|w|$
- Base case : For $|w| = \epsilon$, $\hat{\delta}(q_{00}, \epsilon) = q_{00}$
- Assume the claim for $x \in \Sigma^*$, and show it for $xc, c \in \{a, b\}$.

Language Acceptance : Proof



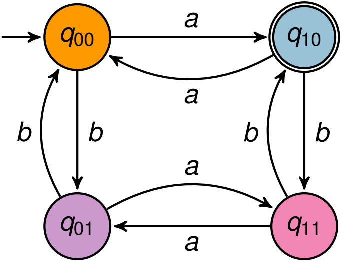
- $\hat{\delta}(q_{00}, xc) = \delta(\hat{\delta}(q_{00}, x), c)$

Language Acceptance : Proof



- $\hat{\delta}(q_{00}, xc) = \delta(\hat{\delta}(q_{00}, x), c)$
- By induction hypothesis, $\hat{\delta}(q_{00}, x) = q_{ij}$ iff
  - parity of i and $|x|_a$ are the same
  - parity of j and $|x|_b$ are the same

Language Acceptance : Proof



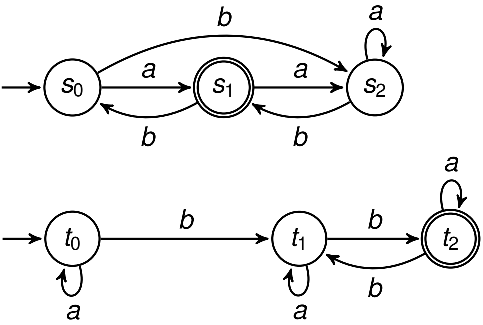
- Case Analysis : If $|x|_a$ odd and $|x|_b$ even, then $i = 1, j = 0$
  - $\delta(q_{10}, a) = q_{00}, \delta(q_{10}, b) = q_{11}$
  - $|xa|_a$ is even and $|xa|_b$ is even
  - $|xb|_a$ is odd and $|xb|_b$ is odd
- Other Cases : Similar
- $\hat{\delta}(q_{00}, x) = q_{10}$ iff $|x|_a$ odd and $|x|_b$ even

Closure Properties : DFA

Closure under Complementation

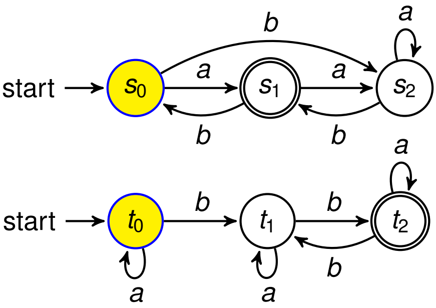
- If L is regular, so is $\overline{L}$
  - Let $A = (Q, q_0, \Sigma, \delta, F)$ be the DFA such that $L = L(A)$
  - For every $w \in L$, $\hat{\delta}(q_0, w) = f$ for some $f \in F$
  - For every $w \notin L$, $\hat{\delta}(q_0, w) = q$ for some $q \notin F$
  - Construct $\overline{A} = (Q, q_0, \Sigma, \delta, Q - F)$
    - $w \in L(\overline{A})$ iff $\hat{\delta}(q_0, w) \in Q - F$ iff $w \notin L(A)$
    - $L(\overline{A}) = \overline{L(A)}$

Closure under Intersection



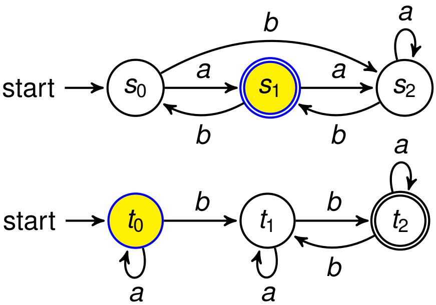
Closure under Intersection

- *aaab*



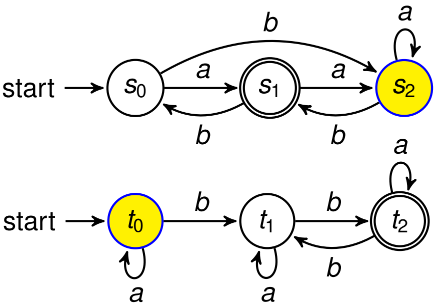
Closure under Intersection

- *aaab*



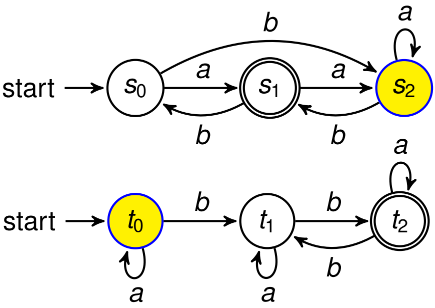
Closure under Intersection

- *aaab*



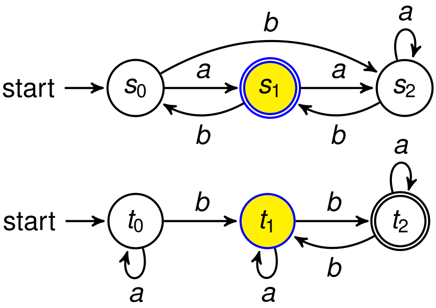
Closure under Intersection

- *aaab*



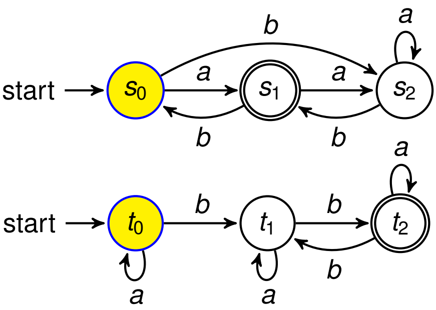
Closure under Intersection

- *aaab*



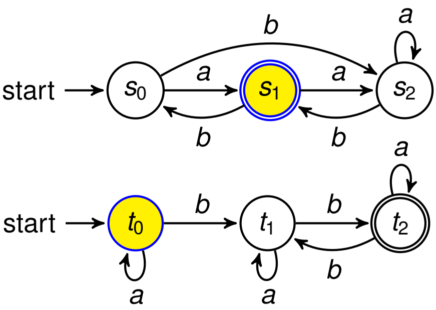
Closure under Intersection

- *aabba*



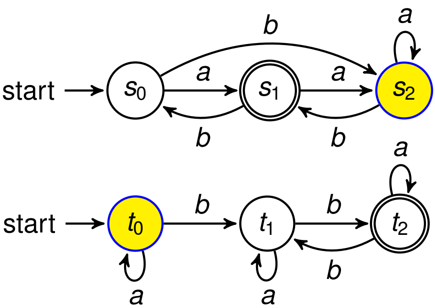
Closure under Intersection

- *aabba*



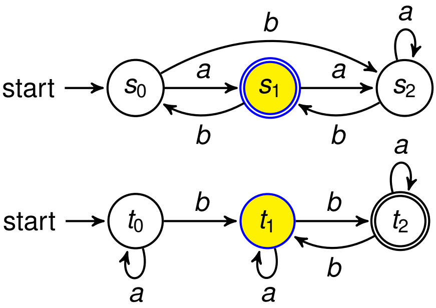
Closure under Intersection

- *aabba*



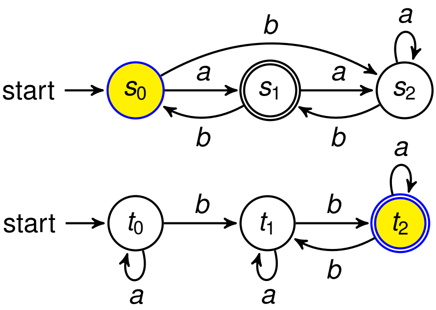
Closure under Intersection

- *aabba*



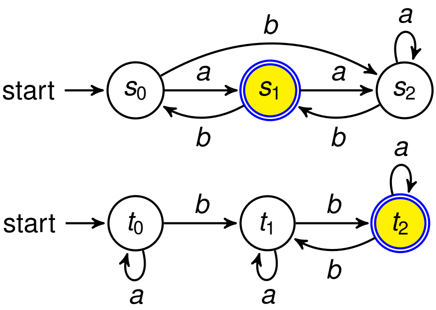
Closure under Intersection

- *aabba*



Closure under Intersection

- *aabba*



Closure under Intersection

- $A_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$
- $A_2 = (Q_2, \Sigma, \delta_2, s_0, F_2)$
- $A = (Q_1 \times Q_2, \Sigma, \delta, (q_0, s_0), F)$,
  - $\delta((q, s), a) = (\delta_1(q, a), \delta_2(s, a))$
  - $F = F_1 \times F_2$

Closure under Intersection

- $A_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$
- $A_2 = (Q_2, \Sigma, \delta_2, s_0, F_2)$
- $A = (Q_1 \times Q_2, \Sigma, \delta, (q_0, s_0), F)$,
  - $\delta((q, s), a) = (\delta_1(q, a), \delta_2(s, a))$
  - $F = F_1 \times F_2$
- Show that for all $x \in \Sigma^*$, $\hat{\delta}((p, q), x) = (\hat{\delta_1}(p, x), \hat{\delta_2}(q, x))$

$x \in L(A)$ iff $\hat{\delta}((q_0, s_0), x) \in F$

Closure under Intersection

- $A_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$
- $A_2 = (Q_2, \Sigma, \delta_2, s_0, F_2)$
- $A = (Q_1 \times Q_2, \Sigma, \delta, (q_0, s_0), F)$,
  - $\delta((q, s), a) = (\delta_1(q, a), \delta_2(s, a))$
  - $F = F_1 \times F_2$
- Show that for all $x \in \Sigma^*$, $\hat{\delta}((p, q), x) = (\hat{\delta_1}(p, x), \hat{\delta_2}(q, x))$

$x \in L(A)$ iff $\hat{\delta}((q_0, s_0), x) \in F$ iff $(\hat{\delta_1}(q_0, x), \hat{\delta_2}(s_0, x)) \in F_1 \times F_2$

Closure under Intersection

- $A_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$
- $A_2 = (Q_2, \Sigma, \delta_2, s_0, F_2)$
- $A = (Q_1 \times Q_2, \Sigma, \delta, (q_0, s_0), F)$,
  - $\delta((q, s), a) = (\delta_1(q, a), \delta_2(s, a))$
  - $F = F_1 \times F_2$
- Show that for all $x \in \Sigma^*$, $\hat{\delta}((p, q), x) = (\hat{\delta}_1(p, x), \hat{\delta}_2(q, x))$

$x \in L(A)$ iff $\hat{\delta}((q_0, s_0), x) \in F$ iff $(\hat{\delta}_1(q_0, x), \hat{\delta}_2(s_0, x)) \in F_1 \times F_2$ iff $\hat{\delta}_1(q_0, x) \in F_1$ and $\hat{\delta}_2(s_0, x) \in F_2$

Closure under Intersection

- $A_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$
- $A_2 = (Q_2, \Sigma, \delta_2, s_0, F_2)$
- $A = (Q_1 \times Q_2, \Sigma, \delta, (q_0, s_0), F)$,
  - $\delta((q,s),a) = (\delta_1(q,a), \delta_2(s,a))$
  - $F = F_1 \times F_2$
- Show that for all $x \in \Sigma^*$, $\hat{\delta}((p,q),x) = (\hat{\delta}_1(p,x), \hat{\delta}_2(q,x))$

$x \in L(A)$ iff $\hat{\delta}((q_0,s_0),x) \in F$ iff $(\hat{\delta}_1(q_0,x), \hat{\delta}_2(s_0,x)) \in F_1 \times F_2$ iff $\hat{\delta}_1(q_0,x) \in F_1$ and $\hat{\delta}_2(s_0,x) \in F_2$ iff $x \in L(A_1)$ and $x \in L(A_2)$

Closure under Union

- $A_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$
- $A_2 = (Q_2, \Sigma, \delta_2, s_0, F_2)$
- $A = (Q_1 \times Q_2, \Sigma, \delta, (q_0, s_0), F)$,
  - $\delta((q, s), a) = (\delta_1(q, a), \delta_2(s, a))$

Closure under Union

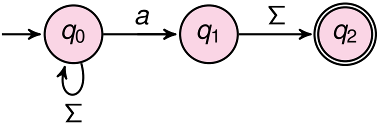
- $A_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$
- $A_2 = (Q_2, \Sigma, \delta_2, s_0, F_2)$
- $A = (Q_1 \times Q_2, \Sigma, \delta, (q_0, s_0), F)$,
  - $\delta((q, s), a) = (\delta_1(q, a), \delta_2(s, a))$
  - $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$
- Show that for all $x \in \Sigma^*$, $\hat{\delta}((p, q), x) = (\hat{\delta}_1(p, x), \hat{\delta}_2(q, x))$

$x \in L(A)$ iff $x \in L(A_1)$ or $x \in L(A_2)$

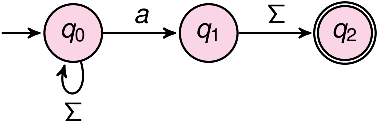
Moving on to Non-determinism

- We looked at DFA
- Showed closure under union, intersection and complementation
- Before we examine closure under concatenation, we look at a more relaxed model, which is as good as a DFA

The Comfort of Non-determinism

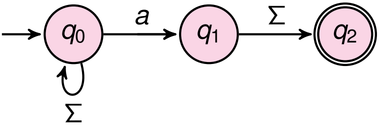


The Comfort of Non-determinism



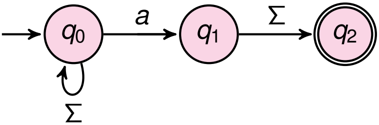
- Assume we relax the condition on transitions, and allow
  - $\delta : Q \times \Sigma \to 2^Q$
  - $\delta(q_0, a) = \{q_0, q_1\}, \delta(q_2, a) = \emptyset$

The Comfort of Non-determinism

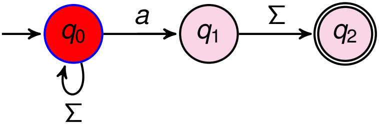


- Assume we relax the condition on transitions, and allow
  - $\delta : Q \times \Sigma \to 2^Q$
  - $\delta(q_0, a) = \{q_0, q_1\}, \delta(q_2, a) = \emptyset$
  - Is *aabb* accepted?

The Comfort of Non-determinism

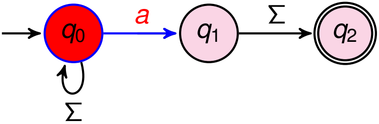


- Assume we relax the condition on transitions, and allow
  - $\delta : Q \times \Sigma \to 2^Q$
  - $\delta(q_0, a) = \{q_0, q_1\}, \delta(q_2, a) = \emptyset$
  - Is *aabb* accepted?



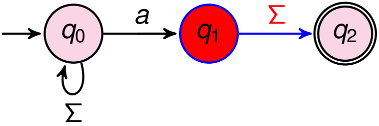
One run of *aabb*

Is *aabb* accepted?



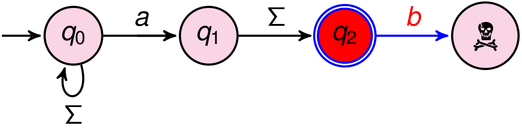
One run of *aabb*

Is *aabb* accepted?



One run of *aabb*

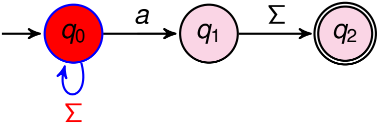
Is *aabb* accepted?



- A non-accepting run for *aabb*

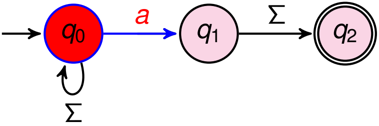
Another run of *aabb*

Is *aabb* accepted?



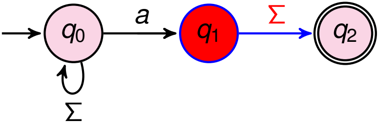
Another run of *aabb*

Is *aabb* accepted?



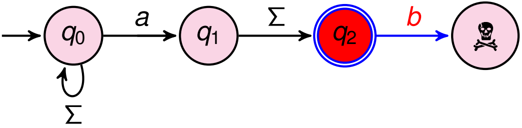
Another run of *aabb*

Is *aabb* accepted?



Another run of *aabb*

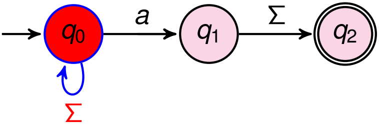
Is *aabb* accepted?



- A non-accepting run for *aabb*

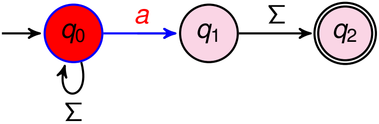
A run of *aaab*

Is *aaab* accepted?



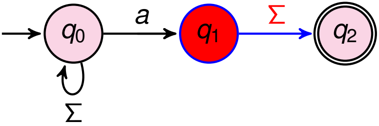
A run of *aaab*

Is *aaab* accepted?



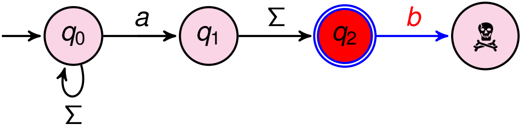
A run of *aaab*

Is *aaab* accepted?



A run of *aaab*

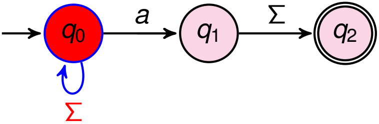
Is *aaab* accepted?



- A non-accepting run for *aaab*

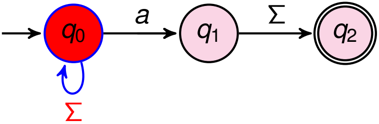
Another run of *aaab*

Is *aaab* accepted?



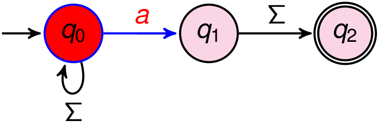
Another run of *aaab*

Is *aaab* accepted?



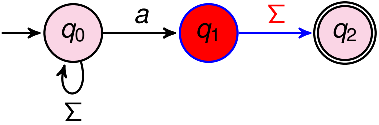
Another run of *aaab*

Is *aaab* accepted?



Another run of *aaab*

Is *aaab* accepted?



- An accepting run for *aaab*

Nondeterministic Finite Automata(NFA)

- $N = (Q, \Sigma, \delta, Q_0, F)$
  - Q is a finite set of states
  - $Q_0 \subseteq Q$ is the set of initial states
  - $\delta : Q \times \Sigma \rightarrow 2^Q$ is the transition function
  - $F \subseteq Q$ is the set of final states
- Acceptance condition : A word w is accepted iff it has atleast one accepting path