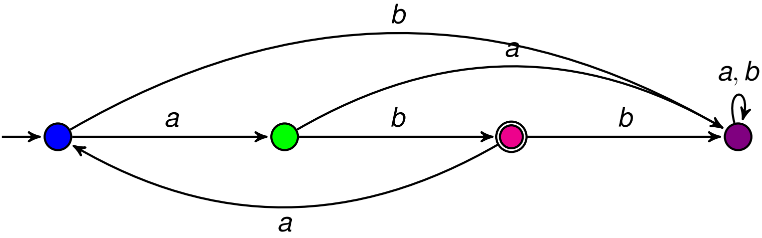
CS 228 : Logic in Computer Science

Krishna. S

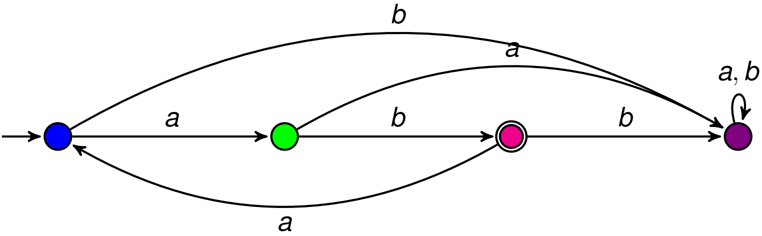
A Quick Recap

- We focus on FO over words : the signature has $<, S, Q_a, Q_b, \dots$. Remember you always have $=$ with you
- Satisfiability of FO over words : reduce FO satisfiability problem to another problem, using automata
- Reduce satisfiability of an FO formula to the emptiness question of an equivalent automata
- Recall how automata accept languages

A First Machine *A*

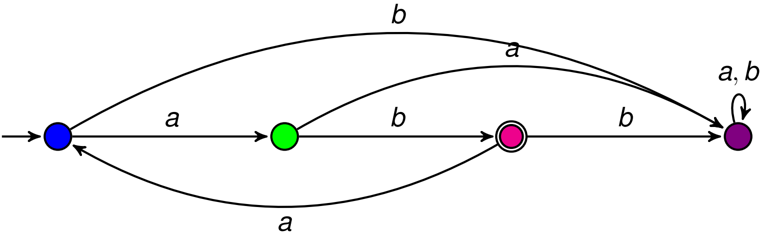


A First Machine A



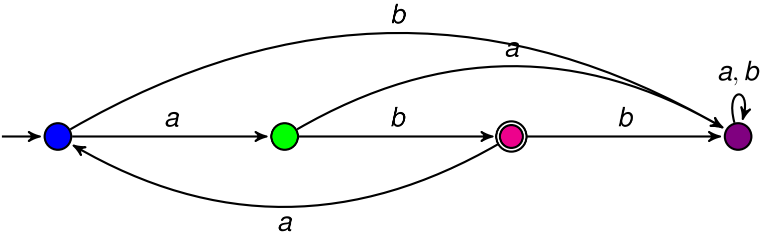
- A path from one state to another gives a word over $\Sigma = \{a, b, c\}$

A First Machine A



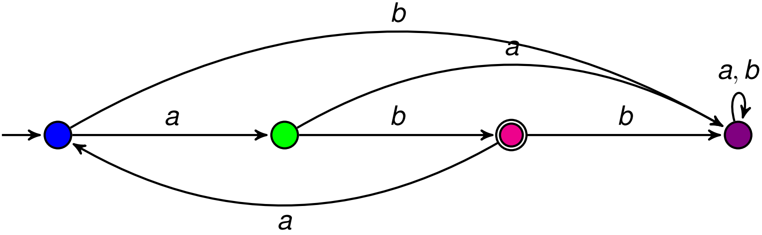
- A path from one state to another gives a word over $\Sigma = \{a, b, c\}$
- The machine accepts words along paths from an initial state to a final state

A First Machine A



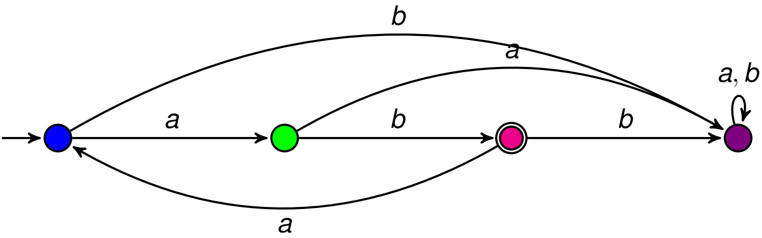
- A path from one state to another gives a word over $\Sigma = \{a, b, c\}$
- The machine accepts words along paths from an initial state to a final state
- The set of words accepted by the machine is called the language accepted by the machine

A First Machine A



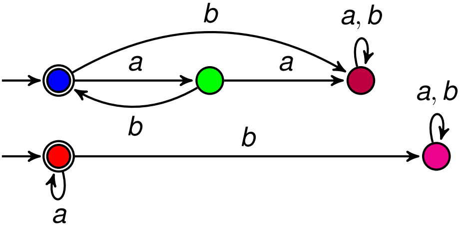
- What is the language L accepted by this machine, $L(A)$?

A First Machine A

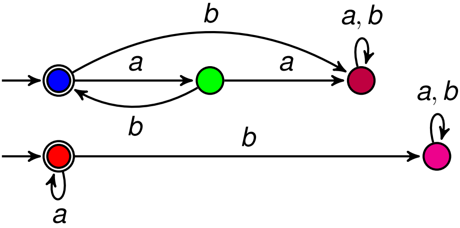


- What is the language L accepted by this machine, $L(A)$?
- Write an FO formula φ such that $L(\varphi) = L(A)$

A Second and a Third Machine B, C

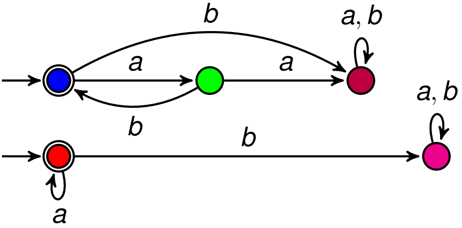


A Second and a Third Machine B, C



- What are $L(B)$, $L(C)$?

A Second and a Third Machine B, C



- What are $L(B), L(C)$?
- Give an FO formula φ such that $L(\varphi) = L(B) \cup L(C)$

Finite State Machines

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

Finite State Machines

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

- Q is a finite set of states

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

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

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

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
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function
- $q_0 \in Q$ is the initial state
- $F \subseteq Q$ is the set of final states

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
- $\delta : Q \times \Sigma \rightarrow Q$ 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 $L = L(A)$.

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

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*

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*

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

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*

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

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

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

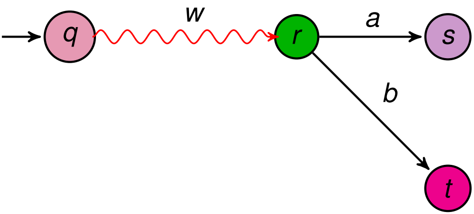
Deterministic Finite Automata

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

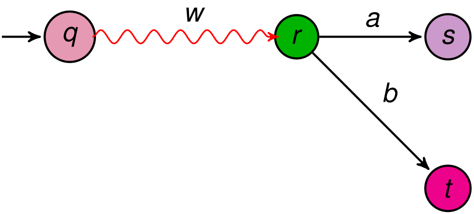
Deterministic Finite Automata

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

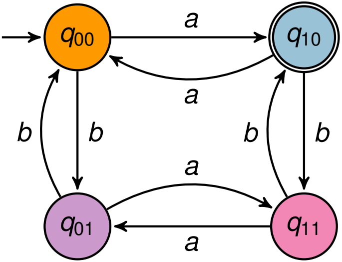
DFA Acceptance

- $w \in \Sigma^*$ 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

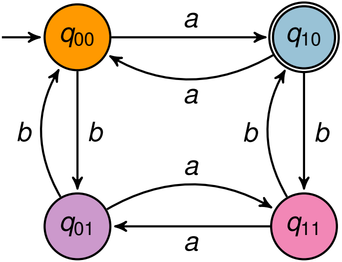
DFA Acceptance

- $w \in \Sigma^*$ 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

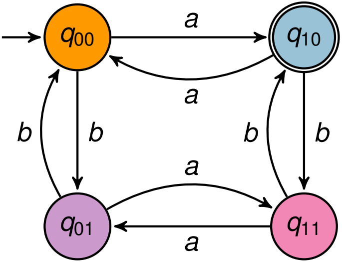


Language Acceptance : Proof



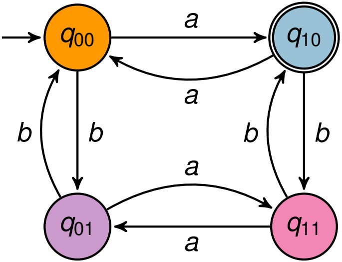
- $L = \{w \in \{a,b\}^* \mid |w|_a \text{ is odd and } |w|_b \text{ is even}\}$

Language Acceptance : Proof



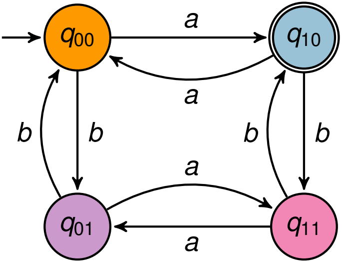
- $L = \{w \in \{a, b\}^* \mid |w|_a \text{ is odd and } |w|_b \text{ is even}\}$
- Show that for any $w \in \Sigma^*$,
  - $\hat{\delta}(q_{00}, w) = q_{ij}$ with $i, j \in \{0, 1\}$, parity of i same as $|w|_a$ and parity of j same as $|w|_b$

Language Acceptance : Proof



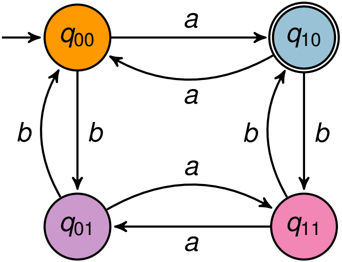
- Prove by induction on $|w|$

Language Acceptance : Proof



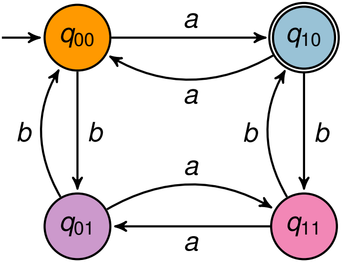
- Prove by induction on $|w|$
- Base case : For $|w| = \epsilon$, $\hat{\delta}(q_{00}, \epsilon) = q_{00}$

Language Acceptance : Proof



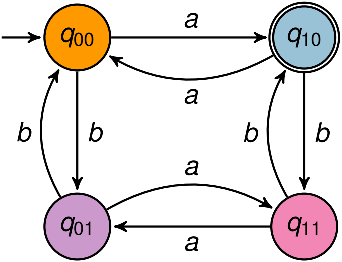
- Prove by induction on $|w|$
- Base case : For $|w| = \epsilon$, $\hat{\delta}(q_{00}, \epsilon) = q_{00}$
- Assume the claim for $x \in \Sigma^*$, and show it for $xc, c \in \{a, b\}$.

Language Acceptance : Proof



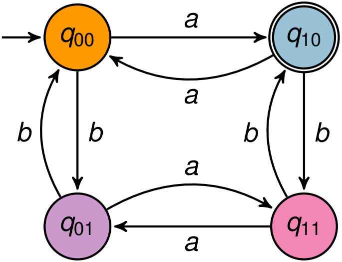
- $\hat{\delta}(q_{00}, xc) = \delta(\hat{\delta}(q_{00}, x), c)$

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



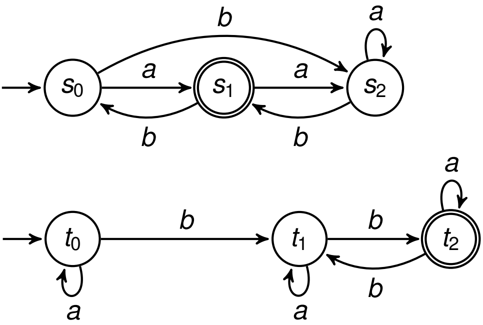
- Case Analysis : If $|x|_a$ odd and $|x|_b$ even, then $i = 1, j = 0$
  - $\delta(q_{10}, a) = q_{00}, \delta(q_{10}, b) = q_{11}$
  - $|xa|_a$ is even and $|xa|_b$ is even
  - $|xb|_a$ is odd and $|xb|_b$ is odd
- Other Cases : Similar
- $\hat{\delta}(q_{00}, x) = q_{10}$ iff $|x|_a$ odd and $|x|_b$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

Closure Properties : DFA

Closure under Complementation

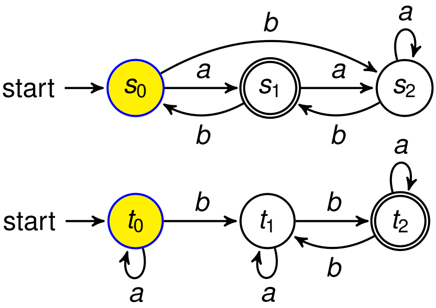
- If L is regular, so is $\overline{L}$
  - Let $A = (Q, q_0, \Sigma, \delta, F)$ be the DFA such that $L = L(A)$
  - For every $w \in L$, $\hat{\delta}(q_0, w) = f$ for some $f \in F$
  - For every $w \notin L$, $\hat{\delta}(q_0, w) = q$ for some $q \notin F$
  - Construct $\overline{A} = (Q, q_0,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

Closure under Intersection



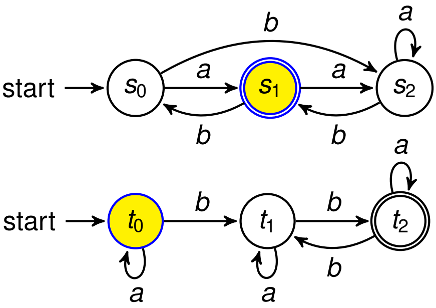
Closure under Intersection

- *aaab*



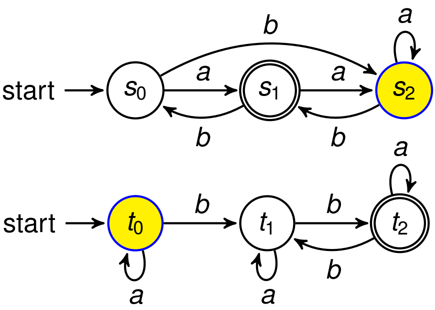
Closure under Intersection

- *a**aab*



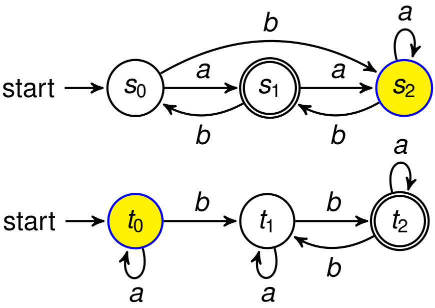
Closure under Intersection

- *aaab*



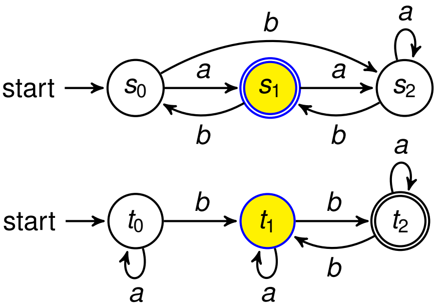
Closure under Intersection

- $aa\color{red}{a}b$



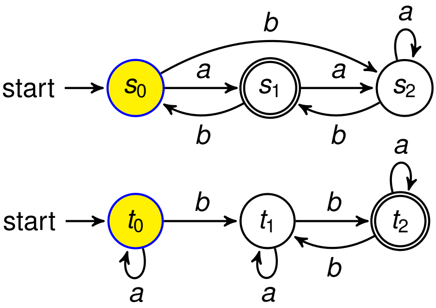
Closure under Intersection

- *aaab*



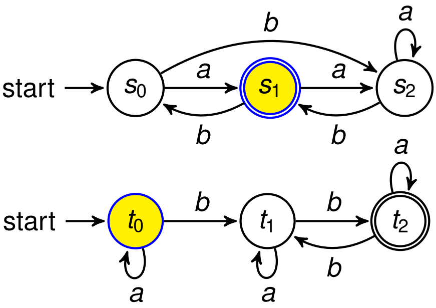
Closure under Intersection

- *aabba*



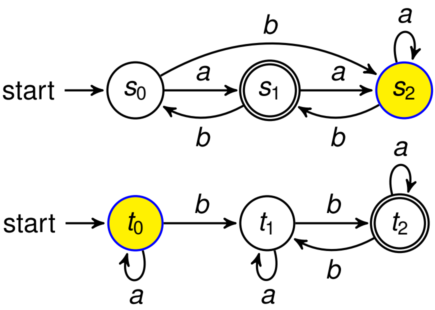
Closure under Intersection

- *aabba*



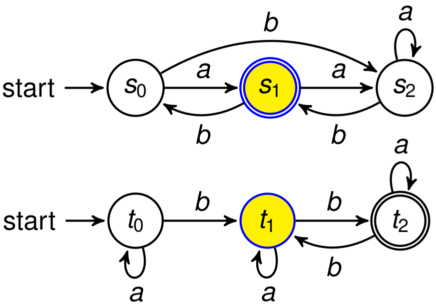
Closure under Intersection

- *aabba*



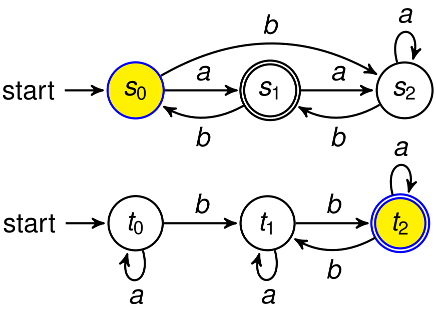
Closure under Intersection

- *aabba*



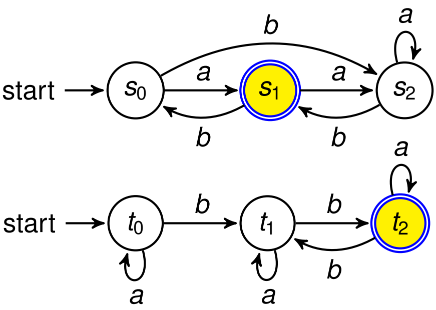
Closure under Intersection

- *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 $x \in L(A_2)$