State Minimization

Lin Chen

Email: Lin.Chen@ttu.edu

Grader: zulfi.khan@ttu.edu

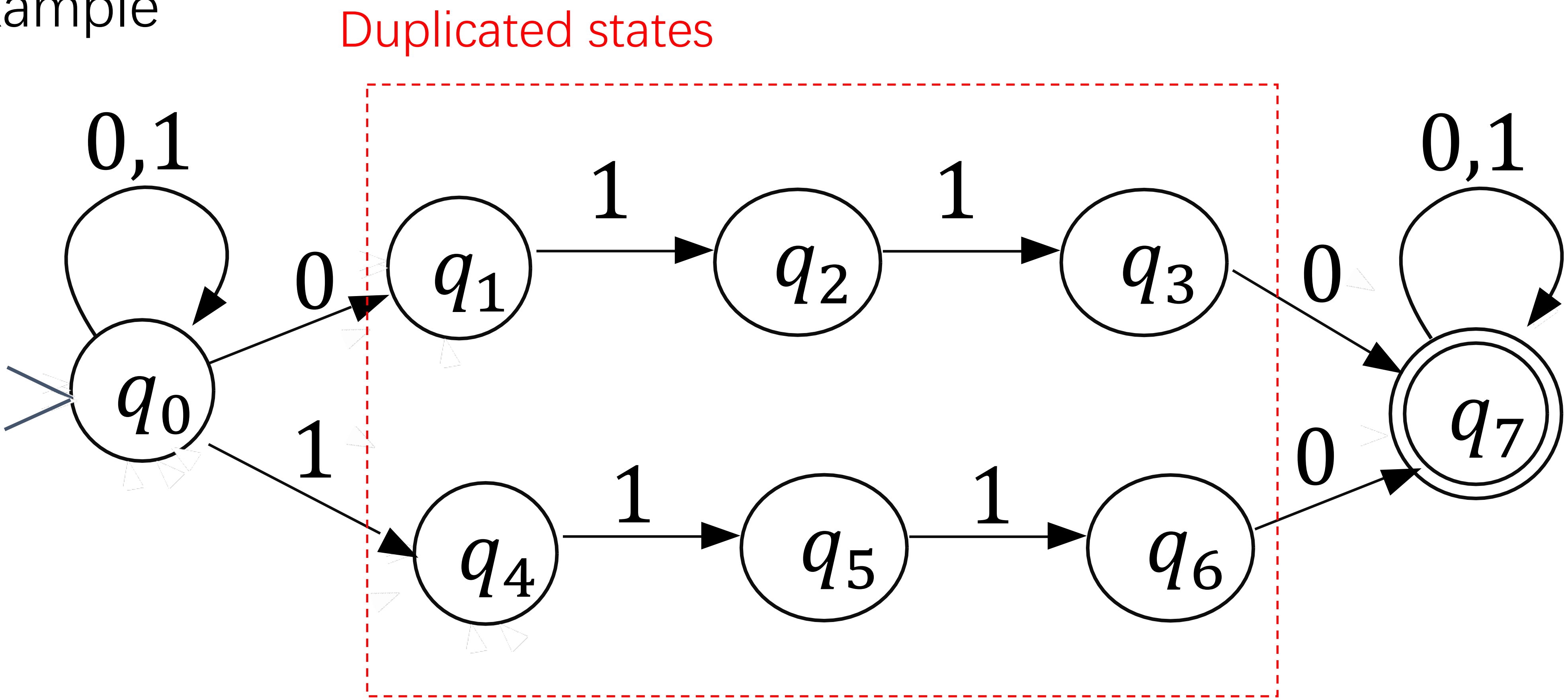
TEXAS TECH
UNIVERSITY

State Minimization

- There can be different DFAs accepting the same language
- Given regular language, we want the minimal DFA

State Minimization

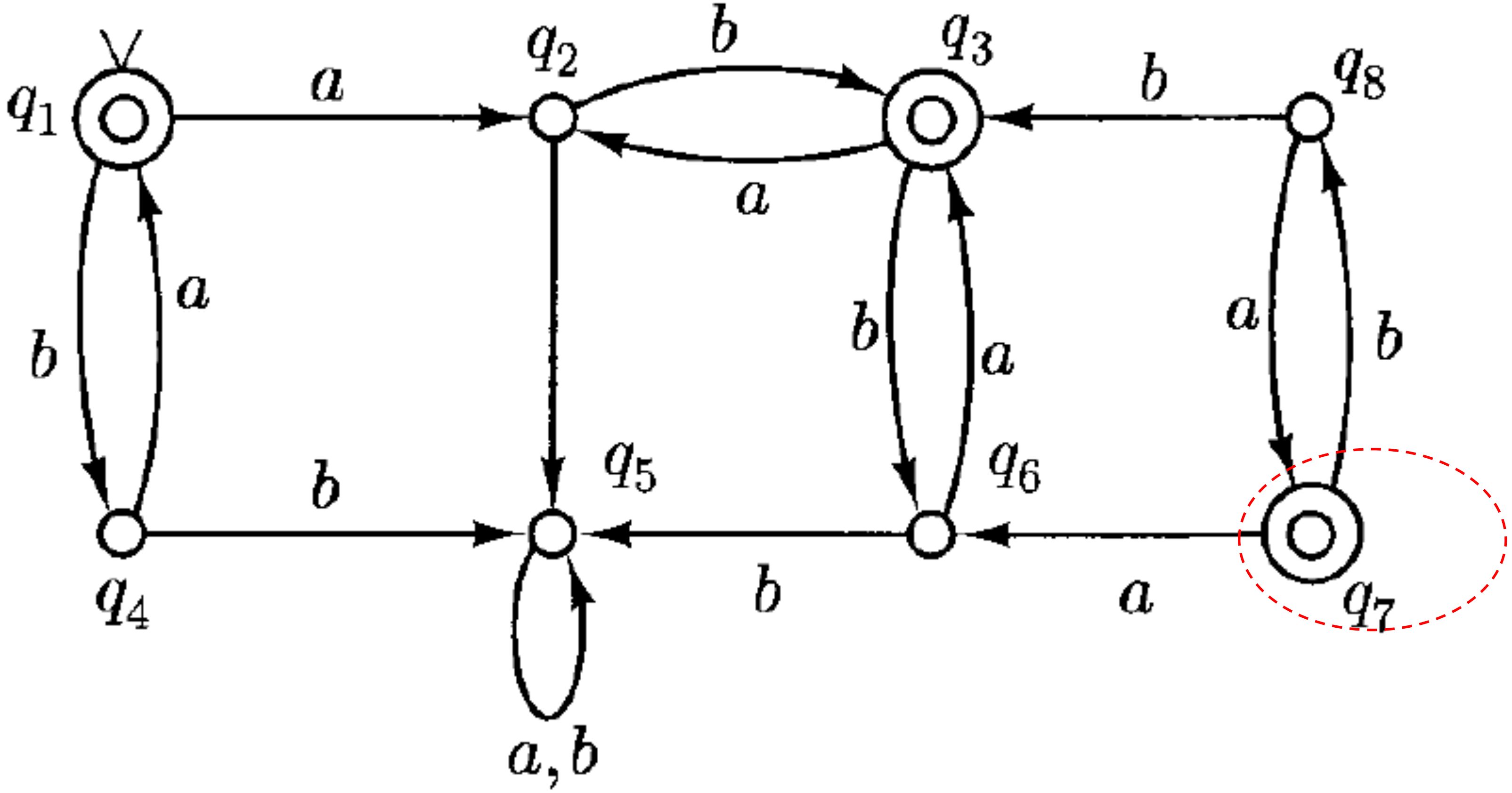
- Example



State Minimization

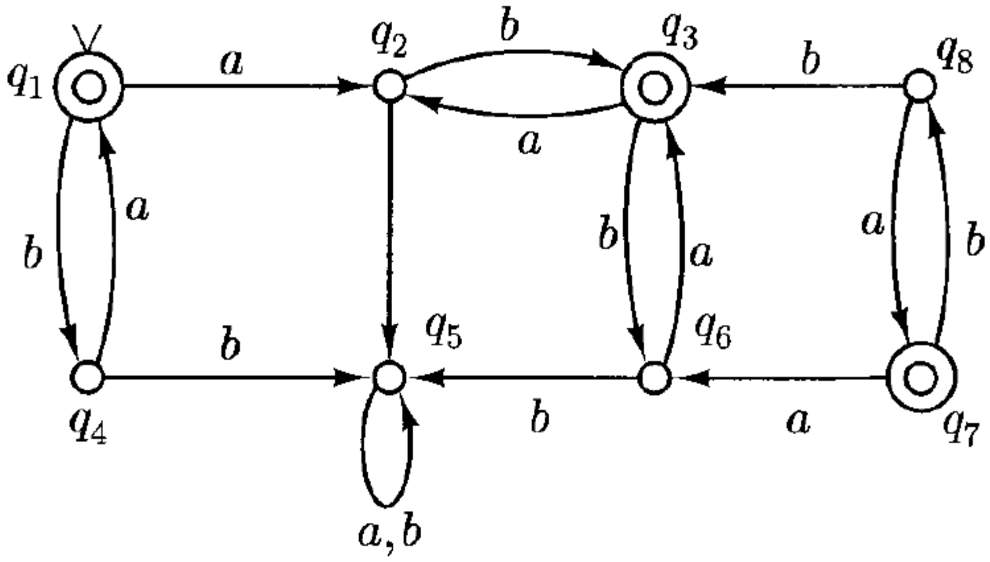
- Example

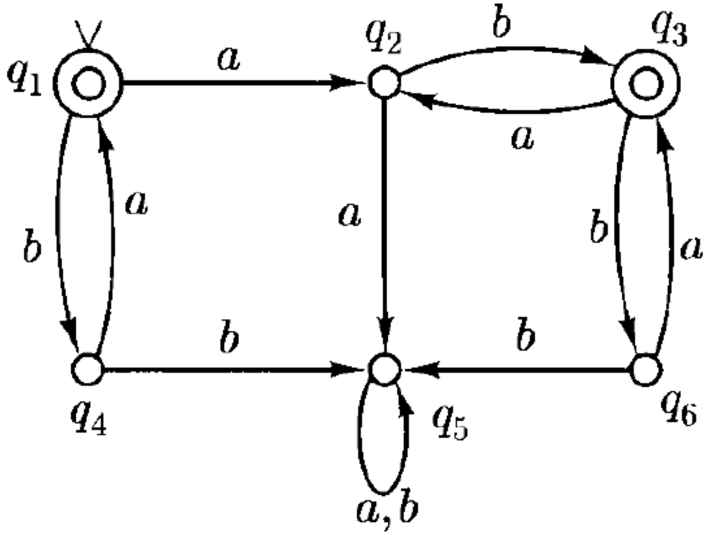
Unreachable states



State Minimization

- Redundant states unnecessarily complicate things
  - Remove unreachable states





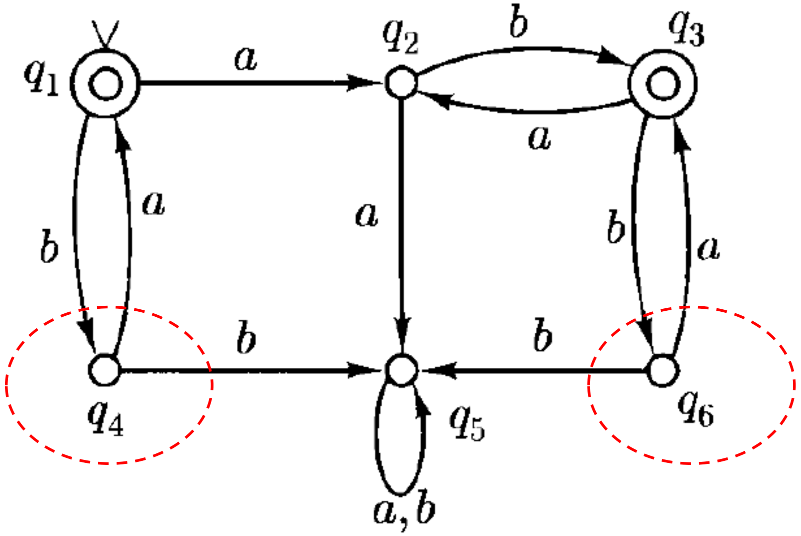
State Minimization

- Redundant states unnecessarily complicate things
  - Remove unreachable states
    - ⋆ Keep a set R, initially $\{s\}$, as the set of states reachable from s
    - ⋆ If any q_i can be reached by one directed edge from some state in R, add it into R
    - ⋆ If we cannot add any new state, R is the set of all reachable states

      Why?

State Minimization

- Redundant states unnecessarily complicate things
  - Remove unreachable states
    - Is this DFA minimal? What about duplicated states?



State Minimization

- We need a formal definition on what we call "duplicated".
- Let $L \subseteq \Sigma^*$ be a language and $x, y \in \Sigma^*$. x, y are equivalent with respect to L, denoted as $x \approx_L y$, if for all $z \in \Sigma^*$, $xz \in L$ if and only if $yz \in L$
  - $\approx_L$ is an equivalence relation. (why?)

State Minimization

- We need a formal definition on what we call "duplicated".
- Let $L \subseteq \Sigma^*$ be a language and $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$. $\boldsymbol{x}, \boldsymbol{y}$ are equivalent with respect to L, denoted as $x \approx_L y$, if for all $z \in \Sigma^*$, $xz \in L$ if and only if $yz \in L$

  - $\approx_L$ is an equivalence relation. (why?)

    Reflexive, Symmetric and Transitive

    - Reflexive: $x \approx_L x$
    - Symmetric: $x \approx_L y$, then $y \approx_L x$
    - Transitive: $x \approx_L y$, $y \approx_L z$, then $x \approx_L z$

State Minimization

- We need a formal definition on what we call "duplicated".
- Let $L \subseteq \Sigma^*$ be a language and $x, y \in \Sigma^*$. x, y are equivalent with respect to L, denoted as $x \approx_L y$, if for all $z \in \Sigma^*$, $xz \in L$ if and only if $yz \in L$
  - $\approx_L$ is an equivalence relation.
  - Denote by $[x]$ the equivalent class containing x

State Minimization

- Example: $L = (ab + ba)^*$, then
  - $[e] = L$
  - $[a] = La$
  - $[b] = Lb$
  - $[aa] = L(aa + bb)\Sigma^*$

State Minimization

- Example: $L = (ab + ba)^*$, then
  - $[e] = L$
  - $[a] = La$
  - $[b] = Lb$
  - $[aa] = L(aa + bb)\Sigma^*$

    $*$ For any $x \in [a]$, $xz \in L$ iff $az \in L$. $az \in L$ iff $z \in bL$. Thus, $xz \in L$ iff $z \in bL$.

State Minimization

- Example: $L = (ab + ba)^*$, then
  - $[e] = L$
  - $[a] = La$
  - $[b] = Lb$
  - $[aa] = L(aa + bb)\Sigma^*$
    - For any $x \in [a]$, $xz \in L$ iff $az \in L$. $az \in L$ iff $z \in bL$. Thus, $xz \in L$ iff $z \in bL$.
    - $[a] \subseteq La$: $xbL \subseteq L$, so $x \in La$

State Minimization

- Example: $L = (ab + ba)^*$, then
  - $[e] = L$
  - $[a] = La$
  - $[b] = Lb$
  - $[aa] = L(aa + bb)\Sigma^*$
    - For any $x \in [a]$, $xz \in L$ iff $az \in L$. $az \in L$ iff $z \in bL$. Thus, $xz \in L$ iff $z \in bL$.
    - $[a] \subseteq La$: $xbL \subseteq L$, so $x \in La$
    - $La \subseteq [a]$: $Laz \subseteq L$ iff $z \in bL$

State Minimization

- A DFA also defines an equivalence relation.
- Let $M = (K, \Sigma, \delta, s, F)$ be a DFA. Two strings x, y are called equivalent with respect to M, denoted $x \sim_M y$, if they both drive M from s to the same state, i.e., there exists q s.t. $(s, x) \vdash_M^* (q, e)$ and $(s, y) \vdash_M^* (q, e)$

State Minimization

- A DFA also defines an equivalence relation.
- Let $M = (K, \Sigma, \delta, s, F)$ be a DFA. Two strings x, y are called equivalent with respect to M, denoted $x \sim_M y$, if they both drive M from s to the same state, i.e., there exists q s.t. $(s, x) \vdash_M^* (q, e)$ and $(s, y) \vdash_M^* (q, e)$

  - $\sim_M$ is an equivalence relation. (why?)

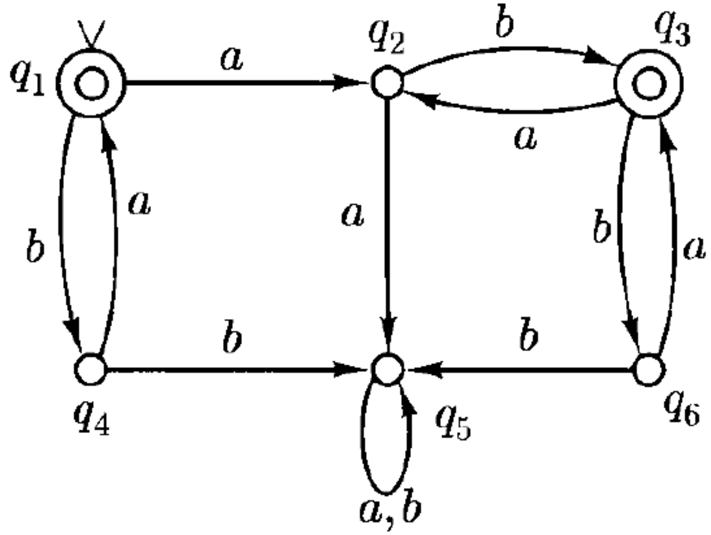
State Minimization

- A DFA also defines an equivalence relation.
- Let $M = (K, \Sigma, \delta, s, F)$ be a DFA. Two strings x, y are called equivalent with respect to M, denoted $x \sim_M y$, if they both drive M from s to the same state, i.e., there exists q s.t. $(s, x) \vdash_M^* (q, e)$ and $(s, y) \vdash_M^* (q, e)$
  - $\sim_M$ is an equivalence relation. (why?)
  - The equivalence class corresponding to a state q is denoted as E_q

All the strings ends at (q, e)

State Minimization

- Example: Let $L = (ab + ba)^*$ and $M =$
  - $E_{q_1} = (ba)^*$
  - $E_{q_2} = La + a$
  - $E_{q_3} = abL$
  - $E_{q_4} = b(ab)^*$
  - $E_{q_5} = L(bb + aa)\Sigma^*$
  - $E_{q_6} = abLb$



State Minimization

Theorem: For any DFA, M, and the language it accepts, $L(M)$, if $x \sim_M y$, then $x \approx_{L(M)} y$.

State Minimization

Theorem: For any DFA, M, and the language it accepts, $L(M)$, if $x \sim_M y$, then $x \approx_{L(M)} y$.

- Let $q(x), q(y) \in K$ be such that $(s, x) \vdash_M^* (q(x), e)$ and $(s, x) \vdash_M^* (q(y), e)$
- $x \sim_M y$ means $q(x) = q(y)$
- For any z, $wz \in L(M)$ iff $(q(w), z) \vdash_M^* (f, e)$ for some $f \in F$ $(w = x, y)$
- $xz \in L(M)$ iff $yz \in L(M)$
- $x \approx_{L(M)} y$

State Minimization

Theorem: For any DFA, M, and the language it accepts, $L(M)$, if $x \sim_M y$, then $x \approx_{L(M)} y$.

Corollary: Any DFA that accepts a regular language $L \in \Sigma^*$ contains at least τ states, where τ is the number of equivalence classes of L.

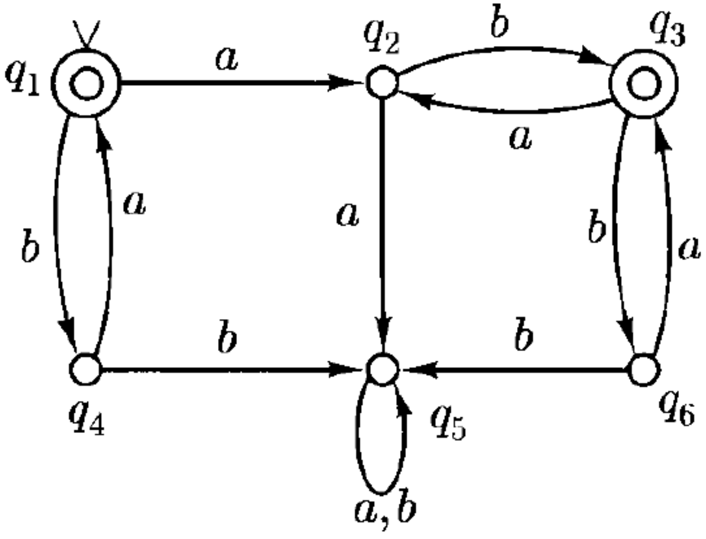
State Minimization

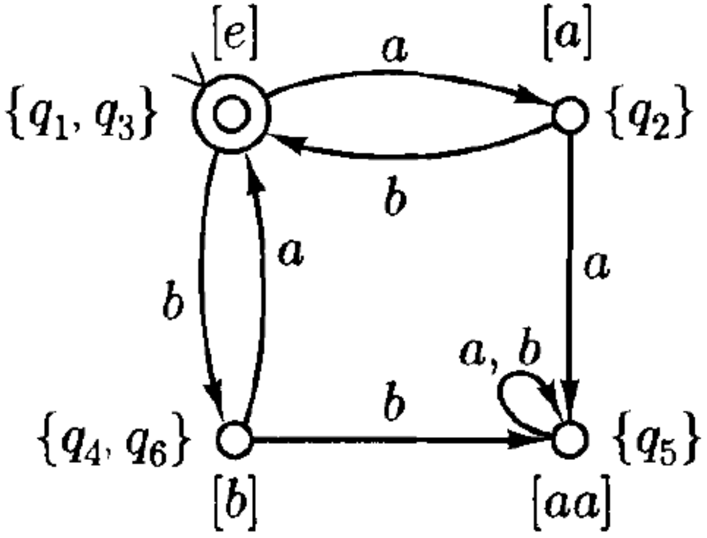
Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

State Minimization

Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

- Example:





State Minimization

Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

- $K = \{[x]: x \in \Sigma^*\}$
- $s = [e]$
- $F = \{[x]: x \in L\}$
- for any $[x] \in K$ and any $a \in \Sigma$, $\delta([x], a) = [xa]$

State Minimization

Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

- $K = \{[x] : x \in \Sigma^*\}$
- $s = [e]$
- $F = \{[x] : x \in L\}$
- for any $[x] \in K$ and any $a \in \Sigma$, $\delta([x], a) = [xa]$

K is finite, why?

State Minimization

Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

- $K = \{[x]: x \in \Sigma^*\}$
- $s = [e]$
- $F = \{[x]: x \in L\}$
- for any $[x] \in K$ and any $a \in \Sigma$, $\delta([x], a) = [xa]$

K is finite, why?

- It is a regular language, there must be some DFA accepting it. By corollary this DFA contains at least $|K|$ states.

State Minimization

Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

- $K = \{[x]: x \in \Sigma^*\}$
- $s = [e]$
- $F = \{[x]: x \in L\}$
- for any $[x] \in K$ and any $a \in \Sigma$, $\delta([x], a) = [xa]$

δ is a function, why?

State Minimization

Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

- $K = \{[x]: x \in \Sigma^*\}$
- $s = [e]$
- $F = \{[x]: x \in L\}$
- for any $[x] \in K$ and any $a \in \Sigma$, $\delta([x], a) = [xa]$

δ is a function, why?

- For any $x' \approx_L x$, $xb \in L$ iff $x'b \in L$ for every $b \in \Sigma^*$. Hence $xab \in L$ iff $x'ab \in L$ for every $b \in \Sigma^*$. Hence $[xa] = [x'a]$.

State Minimization

Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

- $K = \{[x]: x \in \Sigma^*\}$
- $s = [e]$
- $F = \{[x]: x \in L\}$
- for any $[x] \in K$ and any $a \in \Sigma$, $\delta([x], a) = [xa]$

We have indeed constructed a DFA. It remains to show $L = L(M)$

$$([x], y) \vdash_M^* ([xy], e)$$

Induction on the length of y

State Minimization

Theorem: [Myhill-Nerode Theorem]: Let $L \subseteq \Sigma^*$ be a regular language. Then there is a DFA with exactly τ states.

- $K = \{[x]: x \in \Sigma^*\}$
- $s = [e]$
- $F = \{[x]: x \in L\}$
- for any $[x] \in K$ and any $a \in \Sigma$, $\delta([x], a) = [xa]$

We have indeed constructed a DFA. It remains to show $L = L(M)$

$$([x], y) \vdash_M^* ([xy], e)$$

$x \in L(M)$ iff $([e], x) \vdash_M^* (q, e)$ for some $q \in F$, i.e., for some $q = [y]$ where $y \in L$.
Meanwhile $q = [ex] = [x]$, whereas $[x] = [y]$, $x \in L$.