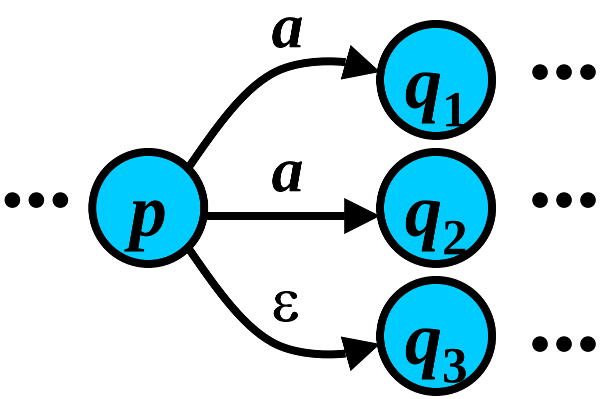
Part IV.
Variants of Finite Automata

Theory vs. Practice

a) Configuration: pax

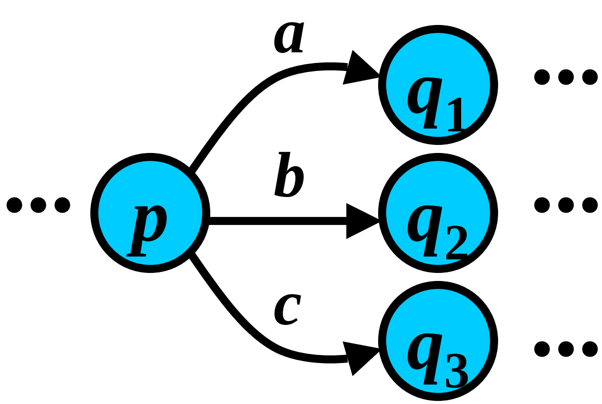


Next Configuration:
q_1x or q_2x or q_3ax **?**

Theory: ☺ × Practice: ☹

---

b) Configuration: pax



Next Configuration:
only q_1x

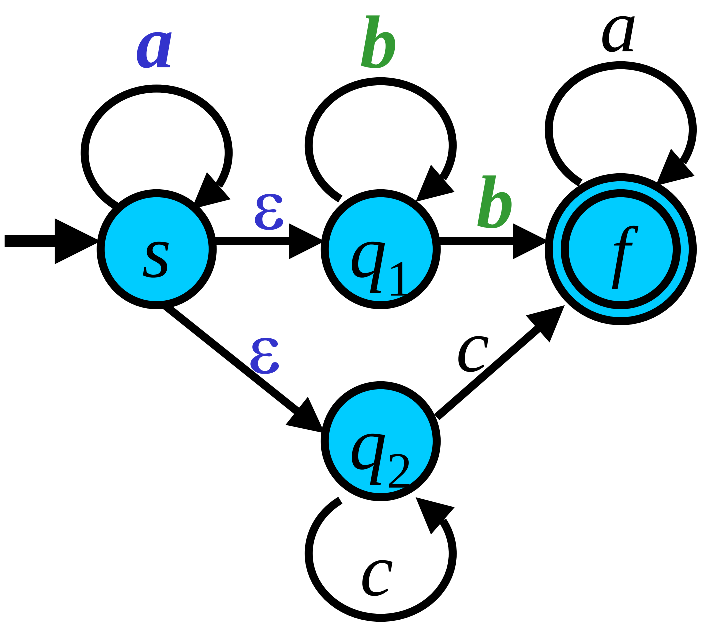
Theory: ☹ × Practice: ☺

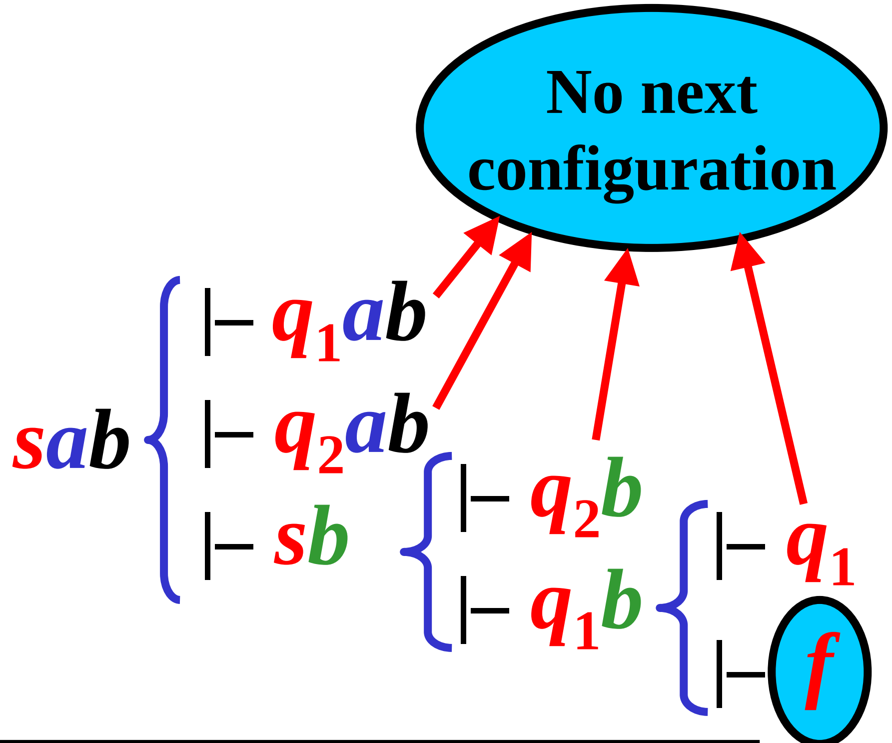
Use of FA in General

Simulation of all possible moves from every configuration.

Example:

FA M is defined as:





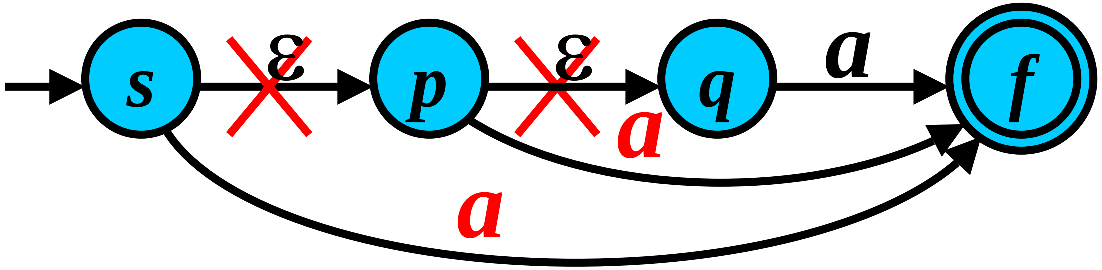
Question: $ab \in L(M)$ **?**

Answer: YES, $ab \in L(M)$ because $f \in F$.

From FA to DFA in Essence 1/2

Preference in practice: *Determinictic FA* (DFA) that makes no more than one move from every configuration.

1) Gist: Removal of ε-moves

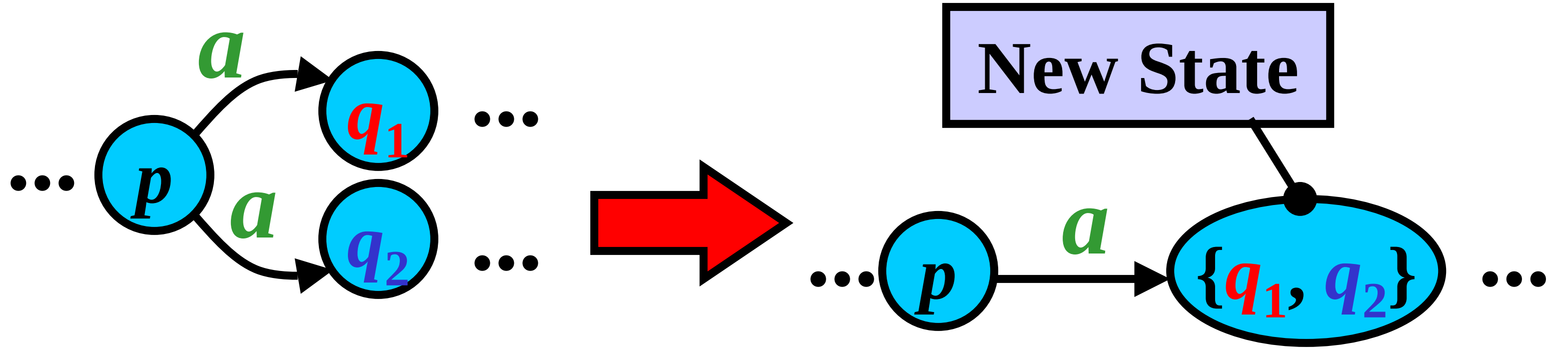


Definition: Let $M = (Q, \Sigma, R, s, F)$ be a FA. M is an ***ε-free finite automaton*** if for all rules $pa \rightarrow q \in R$, where $p, q \in Q$, holds

$$a \in \Sigma \; (a \neq \varepsilon)$$

From FA to DFA in Essence 2/2

2) Gist: Removal of nodeterminism



Definition: Let $M = (Q, \Sigma, R, s, F)$ be an **ε-free FA**. M is a ***deterministic finite automaton*** (DFA) if for each rule $pa \rightarrow q \in R$ it holds that $R - \{pa \rightarrow q\}$ contains no rule with the left-hand side equal to pa.

Theorem

- For every FA M, there is an equivalent DFA M_d.

Proof is based on these conversions:

Finite Automaton M

↓

ε-free Finite Automaton M' — $L(M) = L(M')$

↓

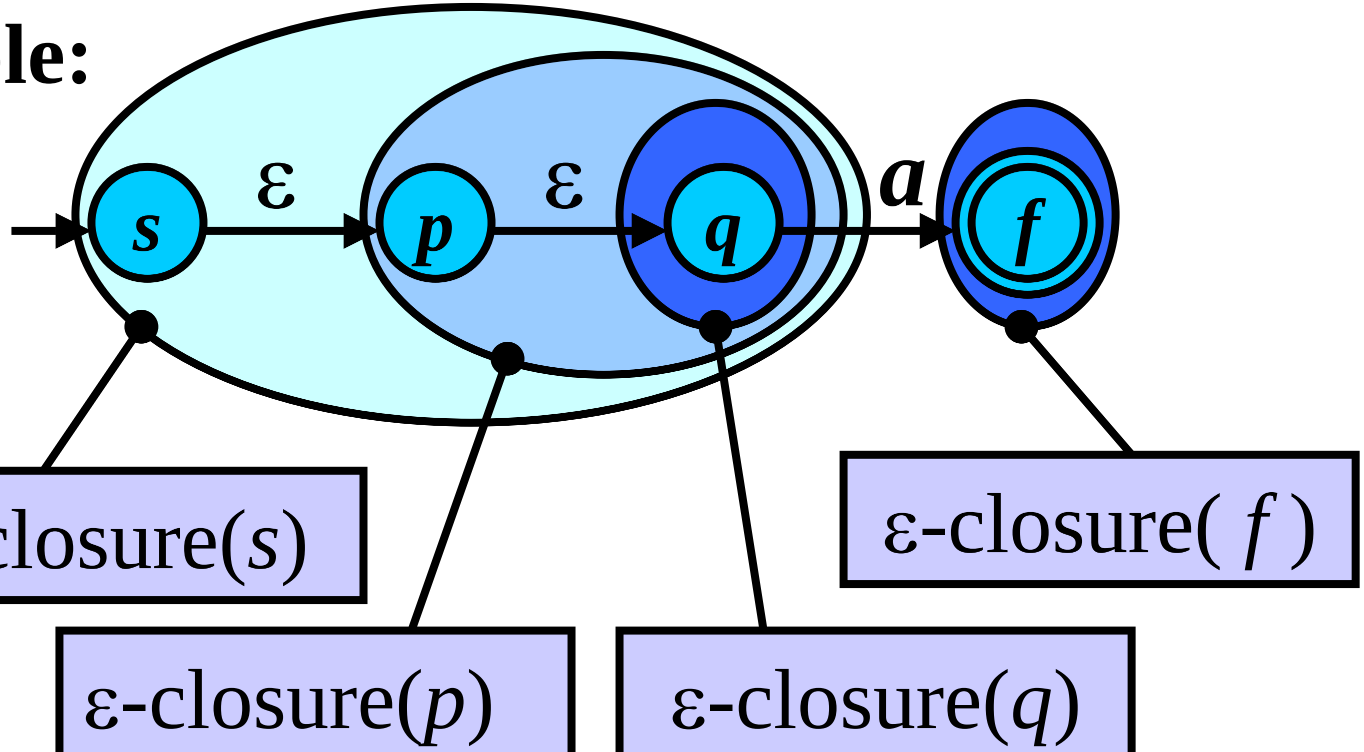
DFA M_d — $L(M') = L(M_d)$

ε-closure

Gist: *q* is in ε-closure(*p*) if FA can reach *q* from *p* without reading.

Definition: For every states $p \in Q$, we define a set *ε-closure*(p) as ε-closure(p) = $\{q: q \in Q, p \vdash^* q\}$

Example:



Algorithm: ε-closure

- **Input:** $M = (Q, \Sigma, R, s, F)$; $p \in Q$
- **Output:** ε-closure(p)

- **Method:**
- $i := 0$; $Q_0 := \{p\}$;
- **repeat**

  $i := i + 1$;

  $Q_i := Q_{i-1} \cup \{\, p' : p' \in Q,\ q \rightarrow p' \in R,$
  $q \in Q_{i-1} \,\}$;

until $Q_i = Q_{i-1}$;

- ε-closure(p) := Q_i.

ε-closure: Example

$M = (Q, \Sigma, R, s, F)$, where: $Q = \{s, p, q, f\}$, $\Sigma = \{a\}$,
$R = \{s \rightarrow p, p \rightarrow q, qa \rightarrow f\}$, $F = \{f\}$

Task: ε-closure(s)

$Q_0 = \{\mathbf{s}\}$

---

1) $\mathbf{s} \rightarrow p'; p' \in Q$: $\mathbf{s} \rightarrow \mathbf{p}$

$Q_1 = \{\mathbf{s}\} \cup \{\mathbf{p}\} = \{\mathbf{s}, \mathbf{p}\}$

---

2) $\mathbf{s} \rightarrow p'; p' \in Q$: $\mathbf{s} \rightarrow \mathbf{p}$
$\mathbf{p} \rightarrow p'; p' \in Q$: $\mathbf{p} \rightarrow \mathbf{q}$

$Q_2 = \{\mathbf{s}, \mathbf{p}\} \cup \{\mathbf{p}, \mathbf{q}\} = \{\mathbf{s}, \mathbf{p}, \mathbf{q}\}$

---

3) $\mathbf{s} \rightarrow p'; p' \in Q$: $\mathbf{s} \rightarrow \mathbf{p}$
$\mathbf{p} \rightarrow p'; p' \in Q$: $\mathbf{p} \rightarrow \mathbf{q}$
$\mathbf{q} \rightarrow p'; p' \in Q$: **none**

$Q_3 = \{\mathbf{s}, \mathbf{p}, \mathbf{q}\} \cup \{\mathbf{p}, \mathbf{q}\} = \{\mathbf{s}, \mathbf{p}, \mathbf{q}\} = Q_2 =$ **ε-closure(*s*)**

Algorithm: FA to ε-free FA

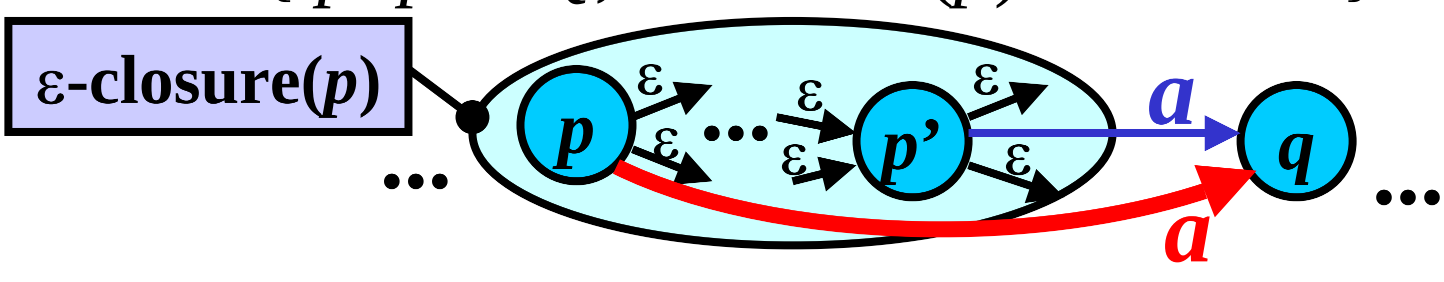
Gist: Skip all ε-moves

- **Input:** FA $M = (Q, \Sigma, R, s, F)$
- **Output:** ε-free FA $M' = (Q, \Sigma, R', s, F')$

- **Method:**
- $R' := \emptyset$;
- **for all** $p \in Q$ **do**

$$R' := R' \cup \{ pa \rightarrow q: p'a \rightarrow q \in R, a \in \Sigma, p' \in \varepsilon\text{-closure}(p), q \in Q \};$$

- $F' := \{ p: p \in Q, \varepsilon\text{-closure}(p) \cap F \neq \emptyset \}$.



FA to ε-free FA: Example 1/3

$M = (Q, \Sigma, R, s, F)$, where:

$Q = \{s, q_1, q_2, f\}$; $\Sigma = \{a, b, c\}$;

$R = \{sa \rightarrow s, s \rightarrow q_1, q_1b \rightarrow q_1, q_1b \rightarrow f, s \rightarrow q_2, q_2c \rightarrow q_2, q_2c \rightarrow f, fa \rightarrow f\}$; $F = \{f\}$

---

1) for $p = \mathbf{s}$: ε-closure($\mathbf{s}$) = $\{\mathbf{s}, \mathbf{q_1}, \mathbf{q_2}\}$

A. $\mathbf{s}d \rightarrow q'$, $d \in \Sigma$, $q' \in Q$: $\mathbf{sa} \rightarrow \mathbf{s}$

B. $\mathbf{q_1}d \rightarrow q'$, $d \in \Sigma$, $q' \in Q$: $\mathbf{q_1b} \rightarrow \mathbf{q_1}$, $\mathbf{q_1b} \rightarrow \mathbf{f}$

C. $\mathbf{q_2}d \rightarrow q'$, $d \in \Sigma$, $q' \in Q$: $\mathbf{q_2c} \rightarrow \mathbf{q_2}$, $\mathbf{q_2c} \rightarrow \mathbf{f}$

$R' = \emptyset \cup \{\mathbf{sa} \rightarrow \mathbf{s}, \mathbf{sb} \rightarrow \mathbf{q_1}, \mathbf{sb} \rightarrow \mathbf{f}, \mathbf{sc} \rightarrow \mathbf{q_2}, \mathbf{sc} \rightarrow \mathbf{f}\}$

FA to ε-free FA: Example 2/3

2) for $p = q_1$: ε-closure(q_1) = {q_1}

A. $q_1 d \rightarrow q'$; $d \in \Sigma$; $q' \in Q$: $q_1 b \rightarrow q_1$, $q_1 b \rightarrow f$

$R' = R' \cup \{q_1 b \rightarrow q_1, q_1 b \rightarrow f\}$

---

3) for $p = q_2$: ε-closure(q_2) = {q_2}

A. $q_2 d \rightarrow q'$; $d \in \Sigma$; $q' \in Q$: $q_2 c \rightarrow q_2$, $q_2 c \rightarrow f$

$R' = R' \cup \{q_2 c \rightarrow q_2, q_2 c \rightarrow f\}$

---

4) for $p = f$: ε-closure(f) = {f}

A. $fd \rightarrow q'$; $d \in \Sigma$; $q' \in Q$: $fa \rightarrow f$

$R' = R' \cup \{fa \rightarrow f\}$

$R' = \{sa \rightarrow s, sb \rightarrow q_1, sb \rightarrow f, sc \rightarrow q_2, sc \rightarrow f,$
$q_1 b \rightarrow q_1, q_1 b \rightarrow f, q_2 c \rightarrow q_2, q_2 c \rightarrow f, fa \rightarrow f\}$

FA to ε-free FA: Example 3/3

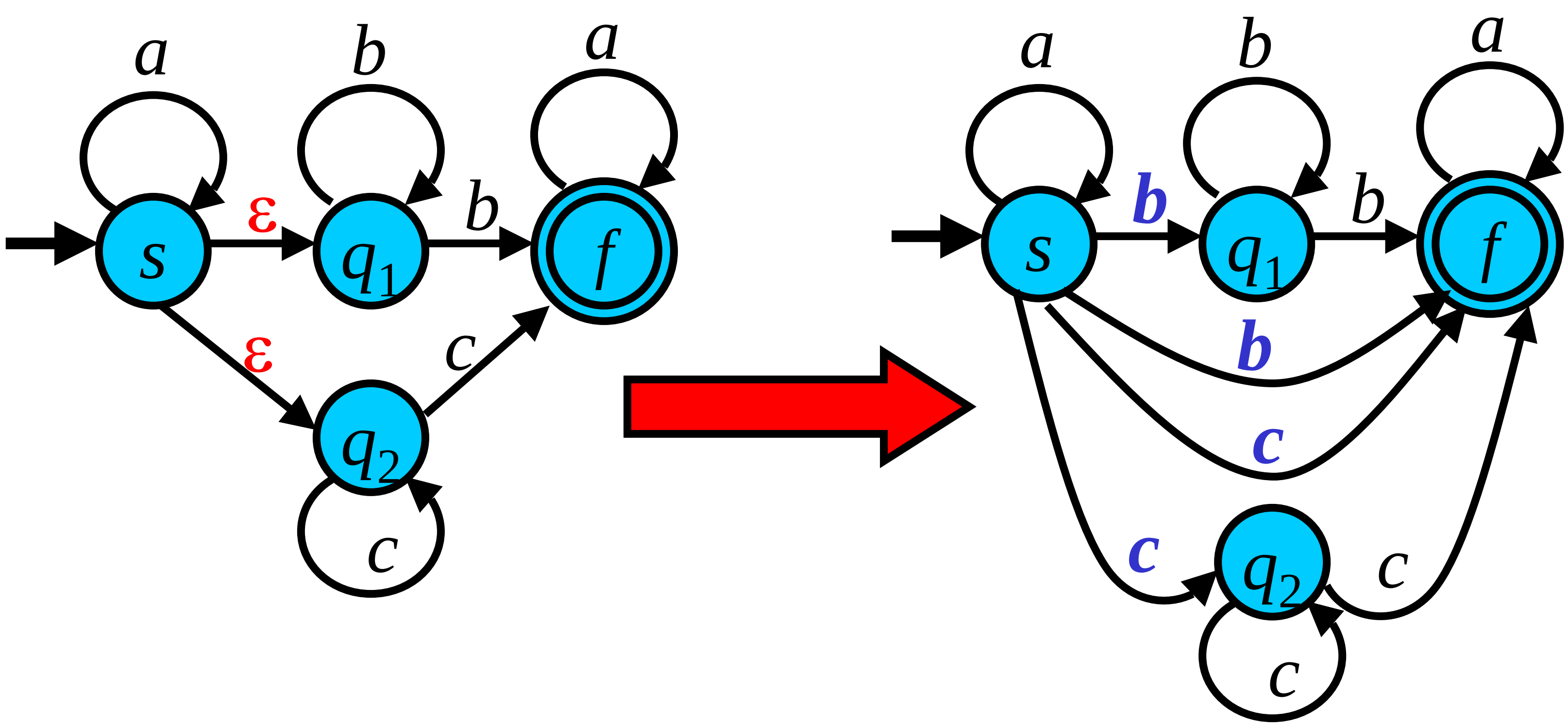
$\varepsilon\text{-closure}(s) \cap F = \{s, q_1, q_2\} \cap \{f\} = \emptyset$

$\varepsilon\text{-closure}(q_1) \cap F = \{q_1\} \cap \{f\} = \emptyset$

$\varepsilon\text{-closure}(q_2) \cap F = \{q_2\} \cap \{f\} = \emptyset$

$\varepsilon\text{-closure}(f) \cap F = \{f\} \cap \{f\} = \{f\} \neq \emptyset$

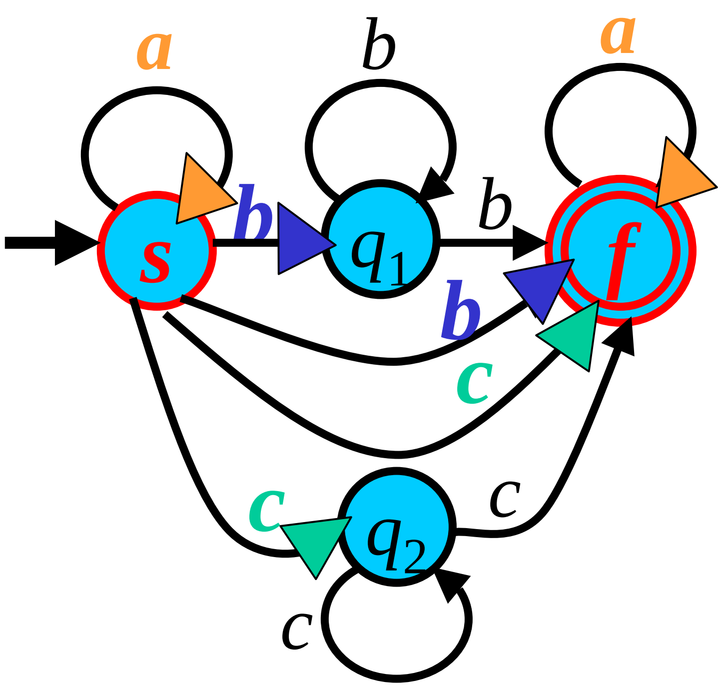
$\Rightarrow F' = \{f\}$



Algorithm: ε-free FA to DFA 1/2

Gist: In DFA, make states from all subsets of states in ε-free FA and move between them so that all possible states of ε-free FA are <u>simultaneously</u> simulated.

Illustration:

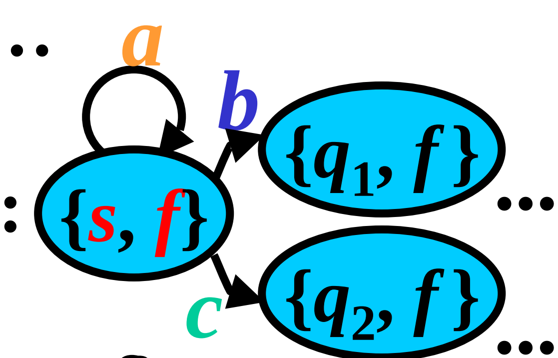


$Q_{DFA} = \{\{s\}, \{q_1\}, \{q_2\}, \{f\}, \{s,q_1\}, \{s,q_2\}, \{s, f\}, \{q_1,q_2\}, \{q_1, f\}, \{q_2, f\}, \{s,q_1,q_2\}, \{s,q_1, f\}, \{s,q_2, f\}, \{q_1,q_2, f\}, \{s,q_1,q_2, f\}\}$

For state $\{s\}$: …

⋮

For state $\{s, f\}$:



⋮

For state $\{s,q_1,q_2, f\}$: …

Algorithm: ε-free FA to DFA 2/2

- **Input:** ε-free FA: $M = (Q, \Sigma, R, s, F)$
- **Output:** DFA: $M_d = (Q_d, \Sigma, R_d, s_d, F_d)$

- **Method:**
- $Q_d := \{Q': Q' \subseteq Q, Q' \neq \emptyset\}$; $R_d := \emptyset$;
- **for each** $Q' \in Q_d$, **and** $a \in \Sigma$ **do begin**

  $Q'' := \{q: p \in Q', pa \rightarrow q \in R\}$;

  if $Q'' \neq \emptyset$ **then** $R_d := R_d \cup \{Q'a \rightarrow Q''\}$;

  end
- $s_d := \{s\}$;
- $F_d := \{F': F' \in Q_d, F' \cap F \neq \emptyset\}$.

ε-free FA to DFA: Example 1/5

$M = (Q, \Sigma, R, s, F)$, where:

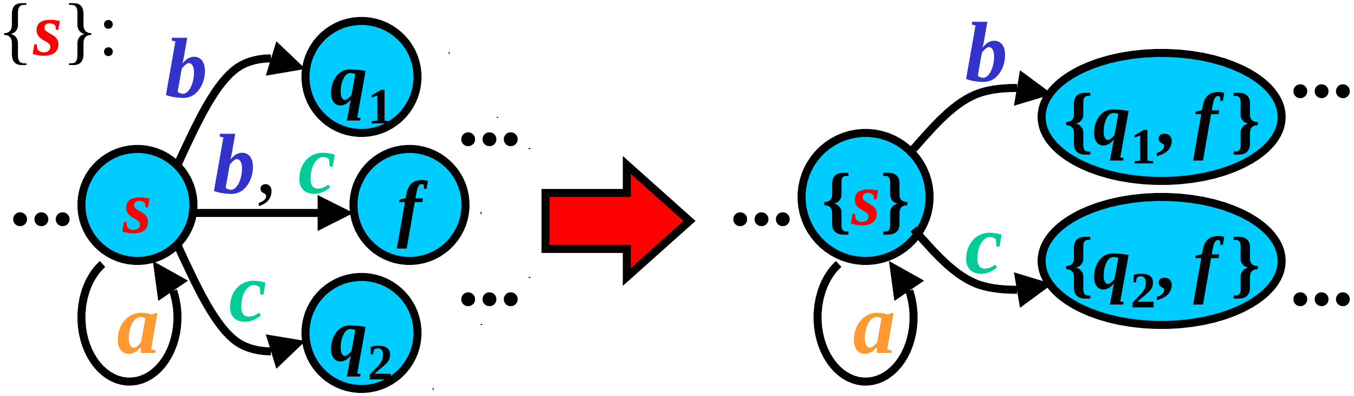
$Q = \{s, q_1, q_2, f\}$; $\Sigma = \{a, b, c\}$; $F = \{f\}$

$R = \{sa \rightarrow s, sb \rightarrow q_1, sb \rightarrow f, sc \rightarrow q_2, sc \rightarrow f,$
$q_1b \rightarrow q_1, q_1b \rightarrow f, q_2c \rightarrow q_2, q_2c \rightarrow f, fa \rightarrow f\}$;

$Q_d = \{\{s\}, \{s,q_1\}, \{s,q_1,q_2\}, \{s,q_1,f\}, \{s,q_1,q_2,f\}, \{s,q_2\}, \{s,q_2,f\},$
$\{s,f\}, \{q_1\}, \{q_1,q_2\}, \{q_1,f\}, \{q_1,q_2,f\}, \{q_2\}, \{q_2,f\}, \{f\}\}$

---

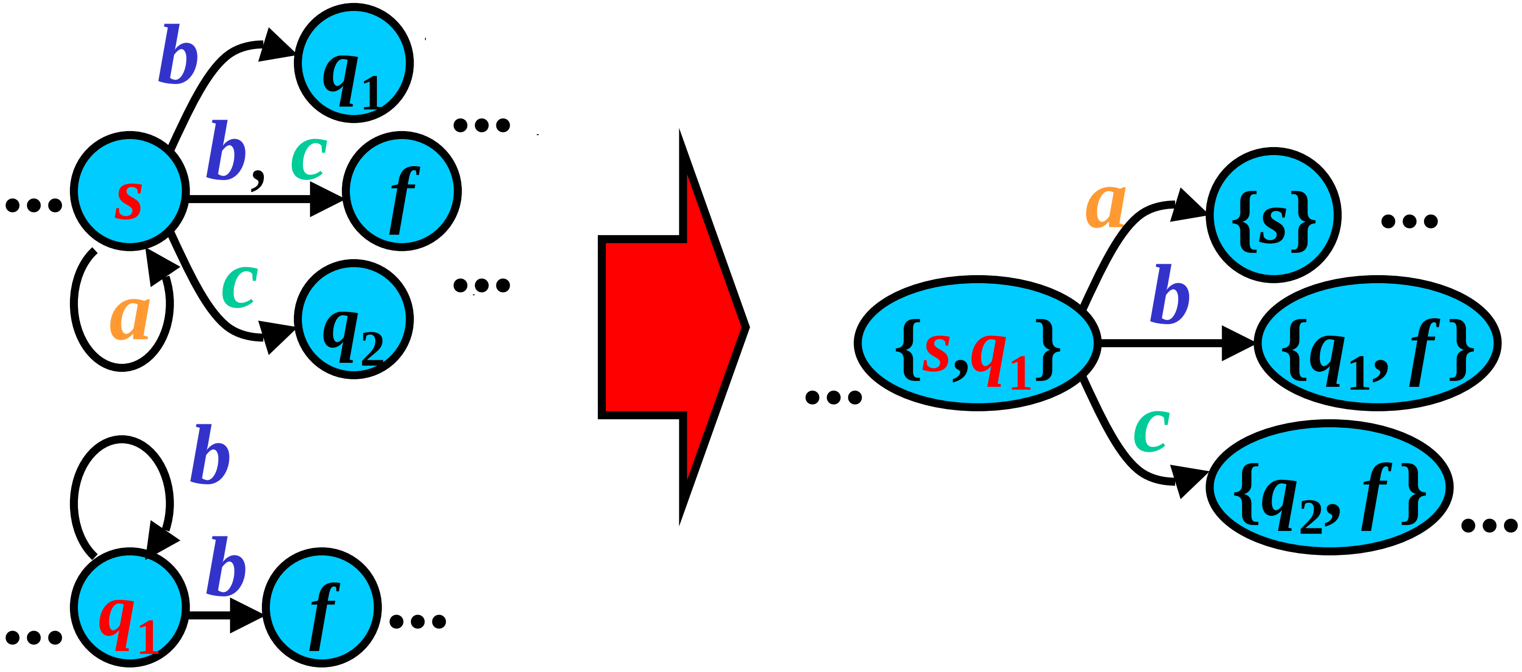
for $Q' = \{s\}$:



$R_d = \emptyset \cup \{\{s\}a \rightarrow \{s\}, \{s\}b \rightarrow \{q_1, f\}, \{s\}c \rightarrow \{q_2, f\}\}$

ε-free FA to DFA: Example 2/5

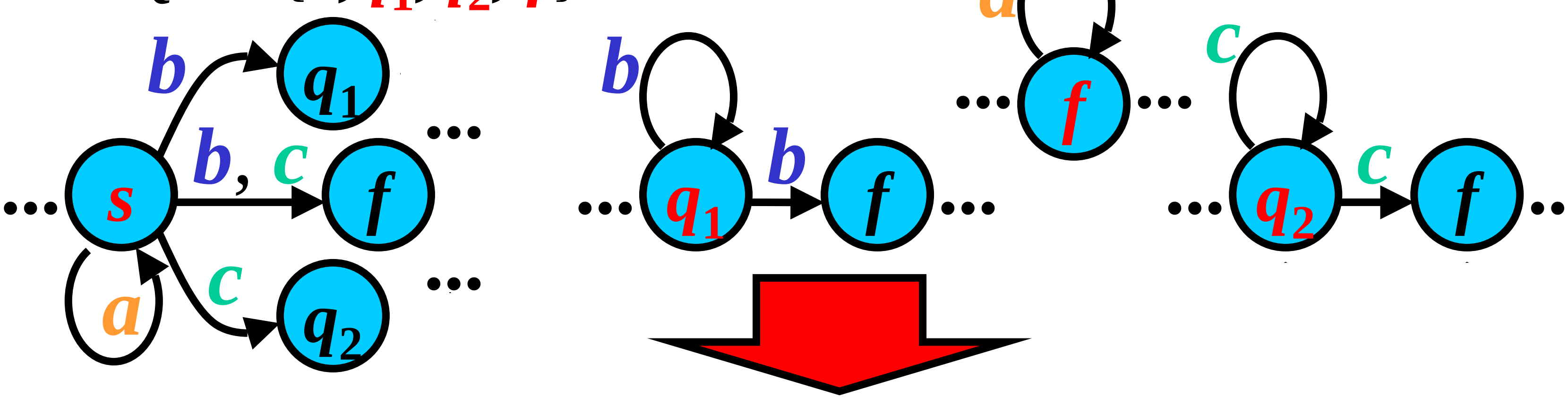
for Q' = {s, q_1}:

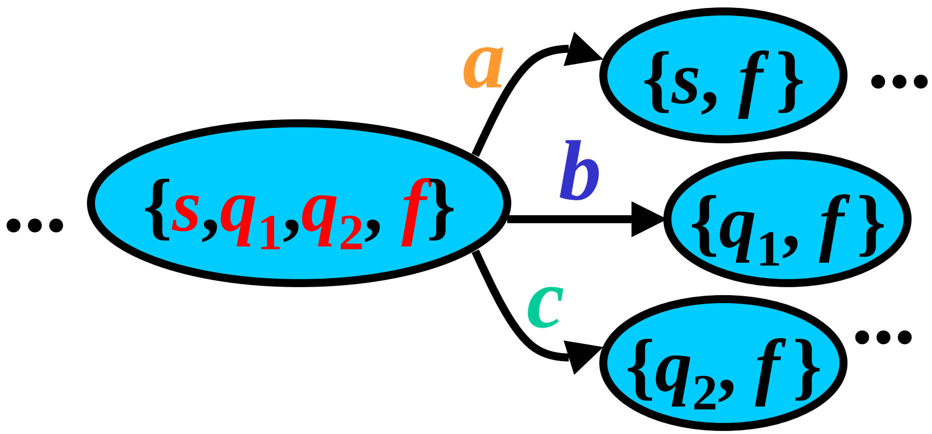


$R_d = R_d \cup \{\{s,q_1\}a \rightarrow \{s\}, \{s,q_1\}b \rightarrow \{q_1, f\}, \{s,q_1\}c \rightarrow \{q_2, f\}\}$

ε-free FA to DFA: Example 3/5

for $Q' = \{s, q_1, q_2, f\}$:





Construct the other states analogically…

$R_d = R_d \cup \{\{s, q_1, q_2, f\}a \rightarrow \{s, f\}, \{s, q_1, q_2, f\}b \rightarrow \{q_1, f\}, \{s, q_1, q_2, f\}c \rightarrow \{q_2, f\}\}$

ε-free FA to DFA: Example 4/5

Final states: $F_d := \{F' : F' \in Q_d, F' \cap F \neq \emptyset\}$

for $F = \{f\}$:

$\{s\} \cap \{f\} = \emptyset \Rightarrow \{s\} \notin F_d$

$\{s, q_1\} \cap \{f\} = \emptyset \Rightarrow \{s, q_1\} \notin F_d$

$\{s, q_1, q_2\} \cap \{f\} = \emptyset \Rightarrow \{s, q_1, q_2\} \notin F_d$

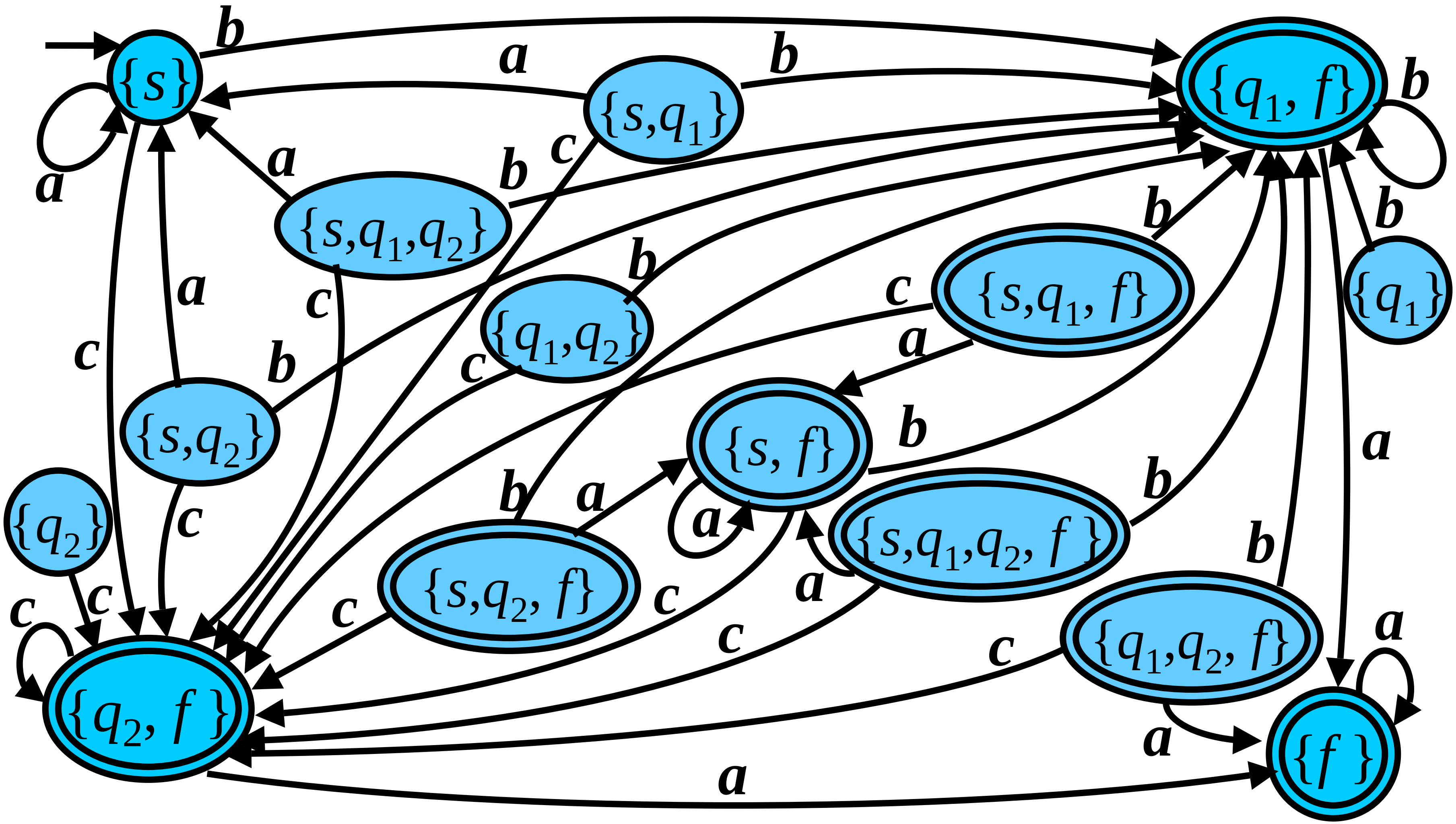
$\{s, q_1, f\} \cap \{f\} = \{f\} \neq \emptyset \Rightarrow \{s, q_1, f\} \in F_d$

$\{s, q_1, q_2, f\} \cap \{f\} = \{f\} \neq \emptyset \Rightarrow \{s, q_1, q_2, f\} \in F_d$

⋮

$F_d = \{\{s, q_1, f\}, \{s, q_1, q_2, f\}, \{s, q_2, f\}, \{s, f\}, \{q_1, f\}, \{q_1, q_2, f\}, \{q_2, f\}, \{f\}\}$

ε-free FA to DFA: Example 5/5



Question: Can we make DFA smaller?

Answer: YES

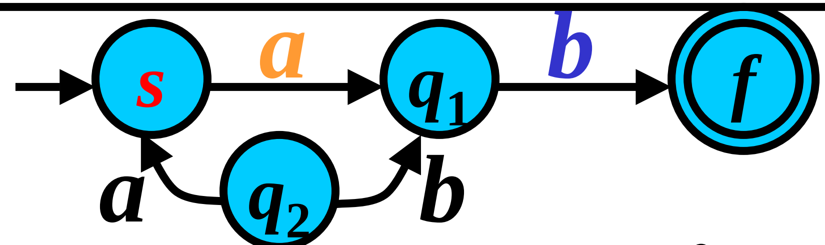
Accessible States

Gist: State *q* is *accessible* if a string takes DFA from *s* (the start state) to *q*.

Definition: Let $M = (Q, \Sigma, R, s, F)$ be an FA.
A state $q \in Q$ is *accessible* if there exists $w \in \Sigma^*$ such that $sw \vdash^* q$; otherwise, q is *inaccessible*.

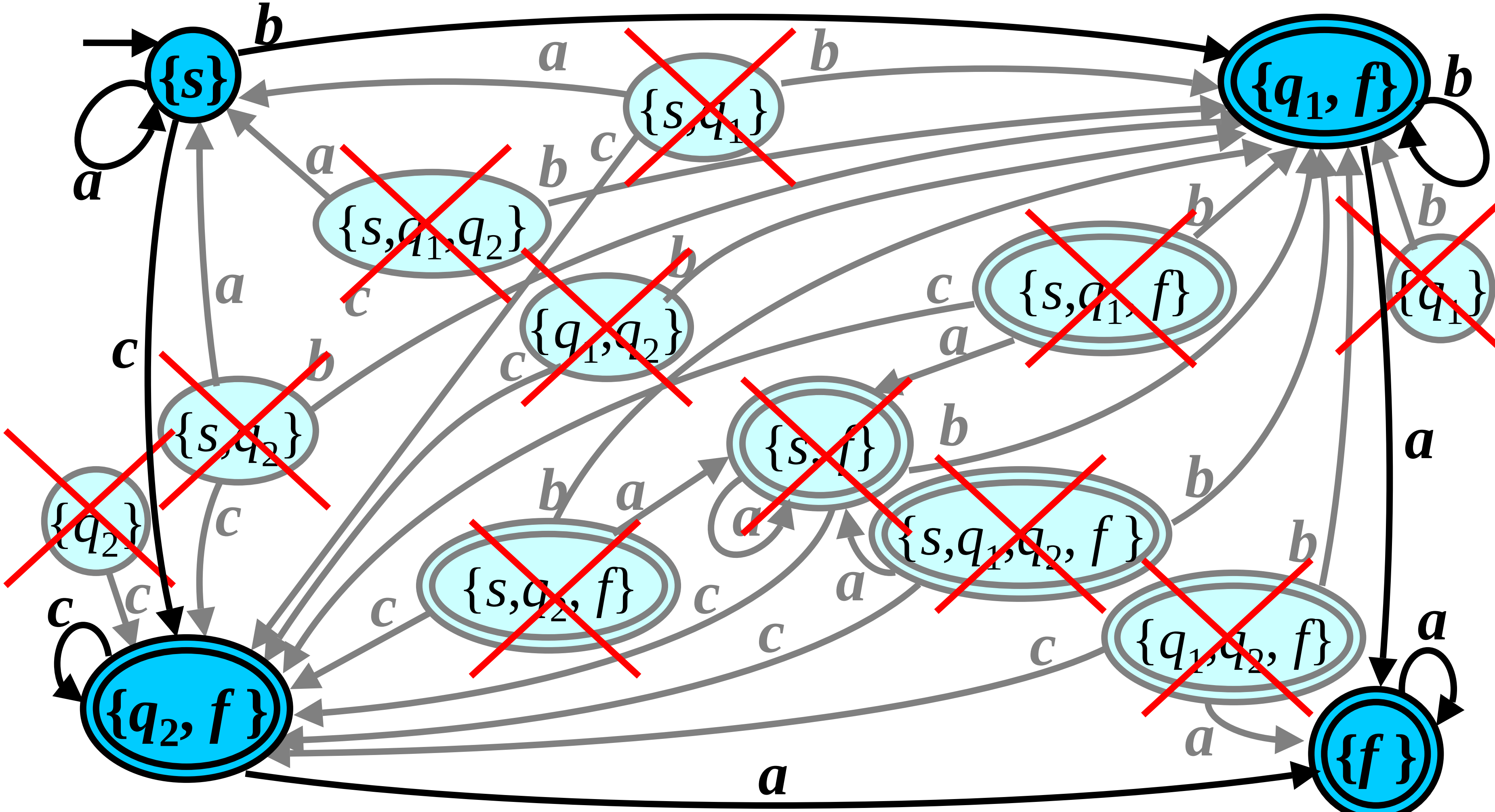
Note: Each inaccesible state can be removed from FA

Example:



State s - accesible: $w = \varepsilon$: $s \vdash^0 s$
State q_1 - accesible: $w = a$: $sa \vdash q_1$
State f - accesible: $w = ab$: $sab \vdash q_1 b \vdash f$
State q_2 - **inaccessible** (there is no $w \in \Sigma^*$ such that $sw \vdash^* q_2$)

Previous Example

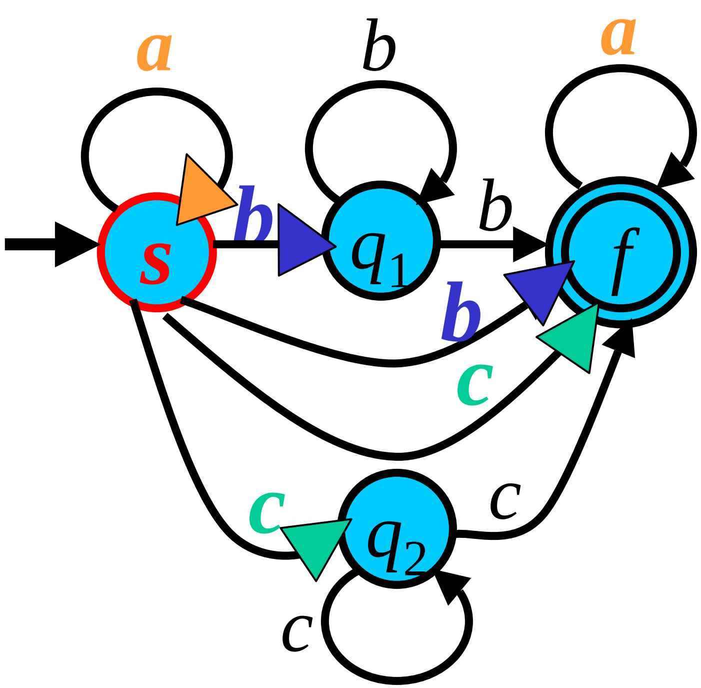


Many inaccessible states

Algorithm II: ε-free FA to DFA 1/2

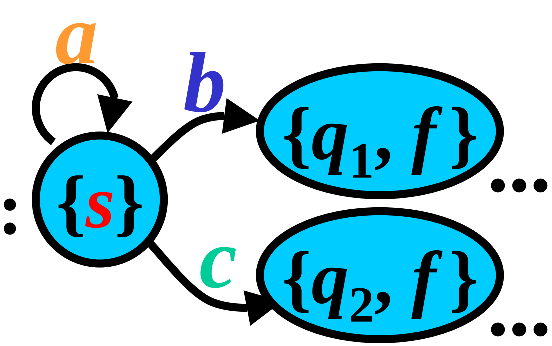
Gist: Analogy to the previous algorithm except that only sets of accessible states are introduced.

Illustration:



$Q_{DFA} = \{\{s\}\}$

For state {*s*}:



Add new states $\{q_1, f\}$, $\{q_2, f\}$ to Q_{DFA}

For state $\{q_1, f\}$: …

For state $\{q_2, f\}$: …

Add new states …

⋮

Algorithm II: ε-free FA to DFA 2/2

- **Input:** ε-*free* FA: $M = (Q, \Sigma, R, s, F)$
- **Output:** DFA: $M_d = (Q_d, \Sigma, R_d, s_d, F_d)$
  without any inaccessible states

- **Method:**
- $s_d := \{s\}$; $Q_{new} := \{s_d\}$; $R_d = \emptyset$; $Q_d := \emptyset$; $F_d := \emptyset$;
- **repeat**
  let $Q' \in Q_{new}$; $Q_{new} := Q_{new} - \{Q'\}$; $Q_d := Q_d \cup \{Q'\}$;
  for each $a \in \Sigma$ **do begin**
    $Q'' := \{q: p \in Q', pa \rightarrow q \in R\}$;
    if $Q'' \neq \emptyset$ **then** $R_d := R_d \cup \{Q'a \rightarrow Q''\}$;
    if $Q'' \notin Q_d \cup \{\emptyset\}$ **then** $Q_{new} := Q_{new} \cup \{Q''\}$
  end;
  if $Q' \cap F \neq \emptyset$ **then** $F_d := F_d \cup \{Q'\}$

until $Q_{new} = \emptyset$.

ε-free FA to DFA: Example 1/3

$M = (Q, \Sigma, R, s, F)$, where:

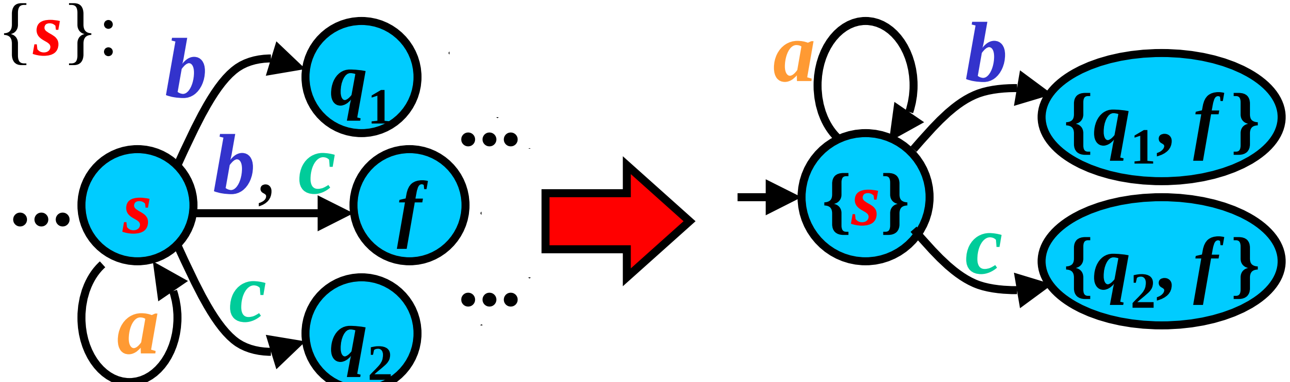
$Q = \{s, q_1, q_2, f\}$; $\Sigma = \{a, b, c\}$; $F = \{f\}$

$R = \{sa \rightarrow s, sb \rightarrow q_1, sb \rightarrow f, sc \rightarrow q_2, sc \rightarrow f,$
$q_1b \rightarrow q_1, q_1b \rightarrow f, q_2c \rightarrow q_2, q_2c \rightarrow f, fa \rightarrow f\}$;

$Q_{new} = \{\{s\}\}$; $R_d = \emptyset$; $Q_d = \emptyset$; $F_d = \emptyset$

---

for $Q' = \{s\}$:

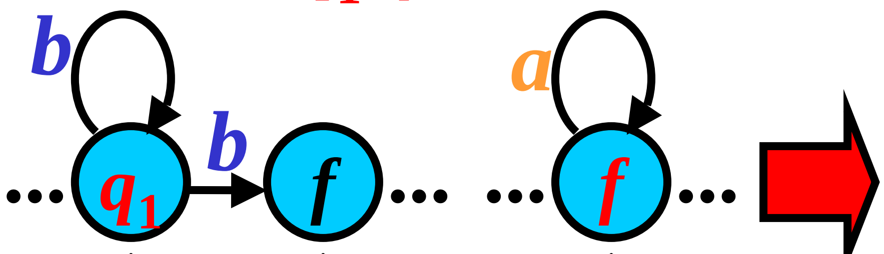


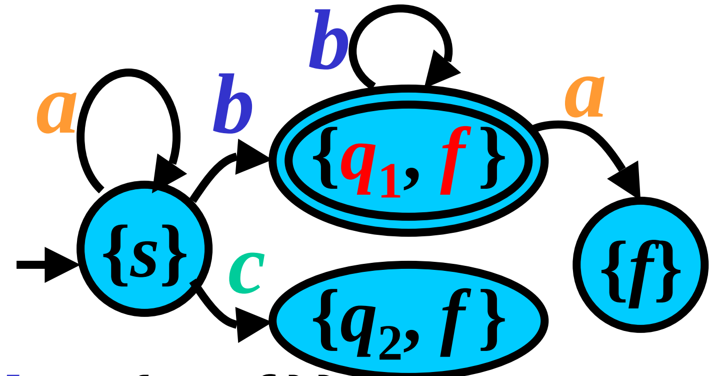
$R_d := \emptyset \cup \{\{s\}a \rightarrow \{s\}, \{s\}b \rightarrow \{q_1, f\}, \{s\}c \rightarrow \{q_2, f\}\}$

$Q_{new} = \{\{q_1, f\}, \{q_2, f\}\}$, $Q_d = \emptyset \cup \{\{s\}\}$, $F_d = \emptyset$

ε-free FA to DFA: Example 2/3

for $Q' = \{q_1, f\}$:



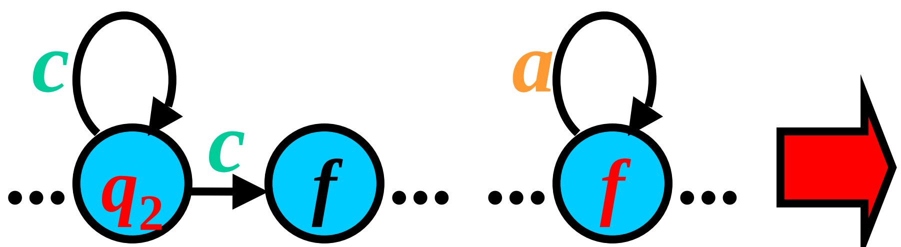


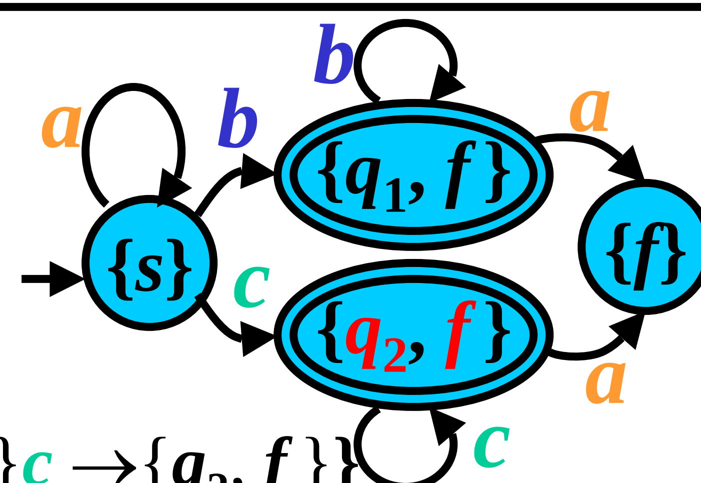
$R_d := R_d \cup \{\{q_1, f\}a \to \{f\}, \{q_1, f\}b \to \{q_1, f\}\}$

$Q_{new} = \{\{q_2, f\}, \{f\}\}, Q_d = Q_d \cup \{\{q_1, f\}\}, F_d := \emptyset \cup \{\{q_1, f\}\}$

---

for $Q' = \{q_2, f\}$:



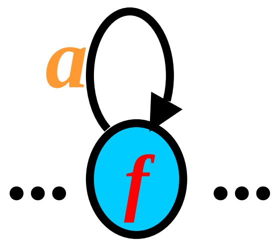


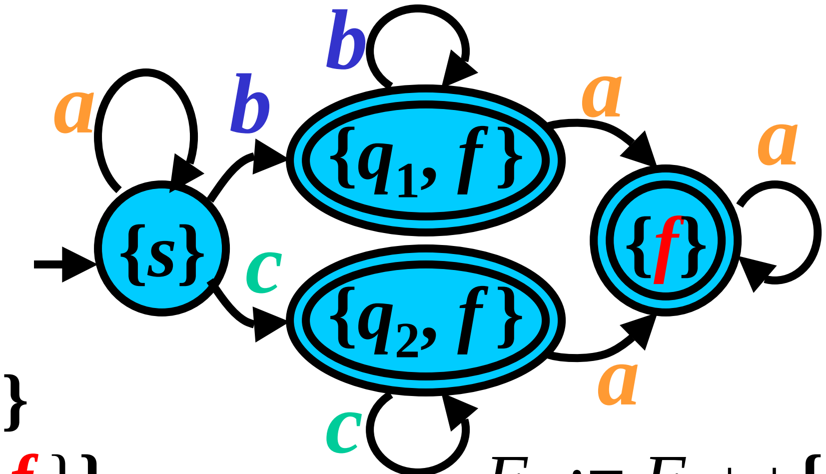
$R_d := R_d \cup \{\{q_2, f\}a \to \{f\}, \{q_2, f\}c \to \{q_2, f\}\}$

$Q_{new} = \{\{f\}\}, Q_d = Q_d \cup \{\{q_2, f\}\}, F_d := F_d \cup \{\{q_2, f\}\}$

ε-free FA to DFA: Example 3/3

for $Q' = \{f\}$:



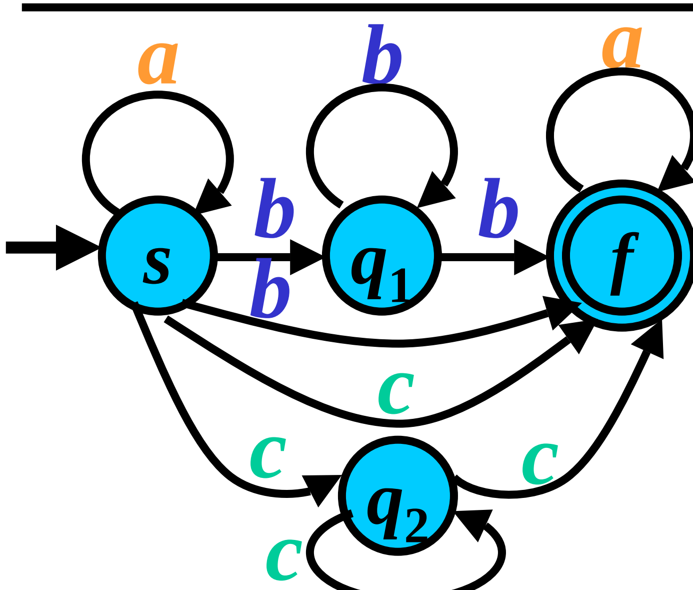


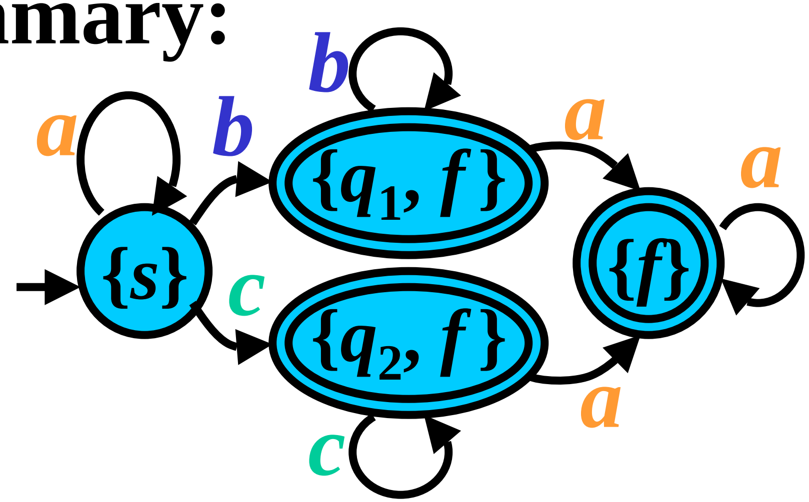
$R_d := R_d \cup \{\{f\}a \to \{f\}\}$

$Q_{new} = \emptyset$, $Q_d = Q_d \cup \{\{f\}\}$,

$F_d := F_d \cup \{\{f\}\}$

Summary:





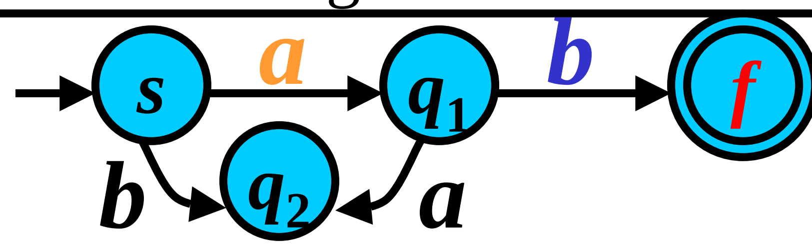
Terminating States

Gist: State q is *terminating* if a string takes DFA from q to a final state.

Definition: Let $M = (Q, \Sigma, R, s, F)$ be a DFA. A state $q \in Q$ is *terminating* if there exists $w \in \Sigma^*$ such that $qw \vdash^* f$ with $f \in F$; otherwise, q is *nonterminating*.

Note: Each nonterminating state can be removed from DFA

Example:



State s - terminating: $w = ab$: $sab \vdash q_1b \vdash f$

State q_1 - terminating: $w = b$: $q_1b \vdash f$

State f - terminating: $w = \varepsilon$: $f \vdash^0 f$

State q_2 - **nonterminating** (there is no $w \in \Sigma^*$ such that $q_2w \vdash^* q$, $q \in F$)

Algorithm: Removal of nont. states

- **Input:** DFA: $M = (Q, \Sigma, R, s, F)$
- **Output:** DFA: $M_t = (Q_t, \Sigma, R_t, s, F)$

- **Method:**
- $Q_0 := F$; $i := 0$;
- **repeat**

  $i := i + 1$;

  $Q_i := Q_{i-1} \cup \{q: qa \rightarrow p \in R, a \in \Sigma, p \in Q_{i-1}\}$;

until $Q_i = Q_{i-1}$;

- $Q_t := Q_i$;
- $R_t := \{qa \rightarrow p: qa \rightarrow p \in R,\ p, q \in Q_t, a \in \Sigma\}$.

Nonterminating States: Example

$M = (Q, \Sigma, R, s, F)$, where: $Q = \{s, q_1, q_2, f\}$, $\Sigma = \{a\}$,
$R = \{sa \to q_1, sb \to q_2, q_1a \to q_2, q_1b \to f\}$, $F = \{f\}$

$Q_0 = \{f\}$

---

1) $qd \to f; q \in Q; d \in \Sigma$: $q_1b \to f$

$Q_1 = \{f\} \cup \{q_1\} = \{f, q_1\}$

---

2) $qd \to f; q \in Q; d \in \Sigma$: $q_1b \to f$
$qd \to q_1; q \in Q; d \in \Sigma$: $sa \to q_1$

$Q_2 = \{f, q_1\} \cup \{q_1, s\} = \{f, q_1, s\}$

---

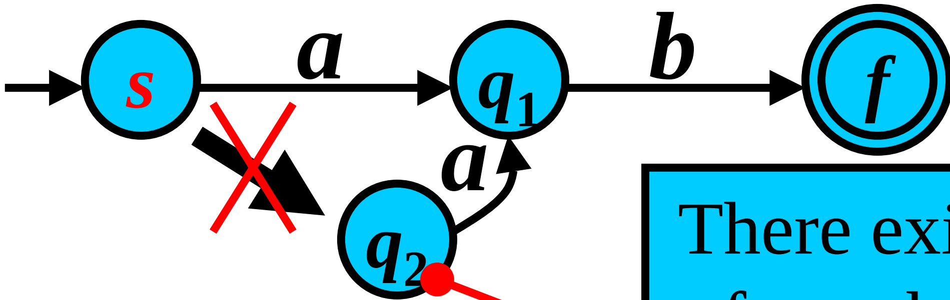
3) $qd \to f; q \in Q; d \in \Sigma$: $q_1b \to f$
$qd \to q_1; q \in Q; d \in \Sigma$: $sa \to q_1$
$qd \to s; q \in Q; d \in \Sigma$: **none**

$Q_3 = \{f, q_1, s\} \cup \{q_1, s\} = \{f, q_1, s\} = Q_2 = Q_t$

$R_t = \{sa \to q_1$, ~~$sb \to q_2$~~, ~~$q_1a \to q_2$~~, $q_1b \to f\}$

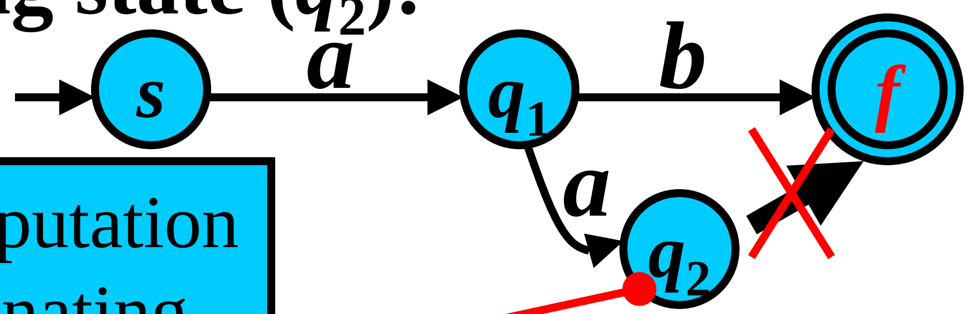
Summary: States to Remove

1) Inaccessible state (q_2):



There exists no computation from the start state to this inaccessible state.

2) Nonterminating state (q_2):



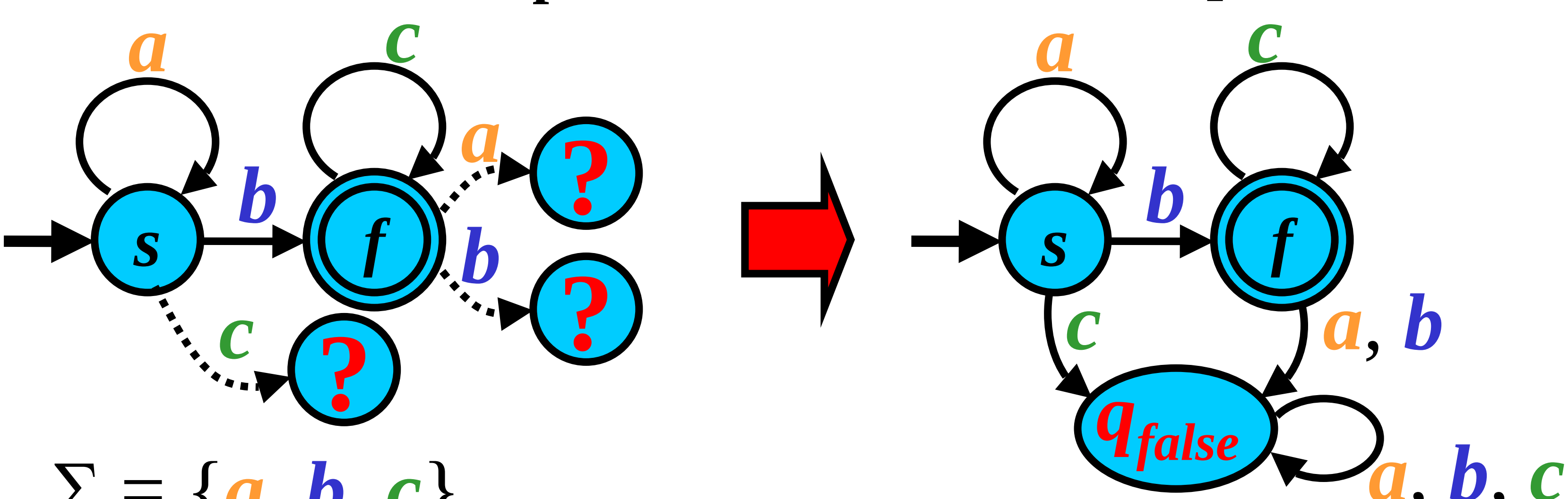
There exists no computation from this nonterminating state to a final state.

Complete DFA

Gist: Complete DFA cannot get stuck.

Definition: Let $M = (Q, \Sigma, R, s, F)$ be a **DFA**. M is ***complete***, if for any $p \in Q$, $a \in \Sigma$ there is exactly one rule of the form $pa \rightarrow q \in R$ for some $q \in Q$; otherwise, M is ***incomplete***

Conversion: **Incomplete DFA** **to Complete DFA**



$\Sigma = \{a, b, c\}$

Algorithm: DFA to Complete DFA

Gist: Add a "trap" state

- **Input:** Incomplete DFA $M = (Q, \Sigma, R, s, F)$
- **Output:** Complete DFA $M_c = (Q_c, \Sigma, R_c, s, F)$

- **Method:**
- $Q_c := Q \cup \{q_{false}\}$;
- $R_c := R \cup \{qa \rightarrow q_{false}: a \in \Sigma, q \in Q_c,$
  $qa \rightarrow p \notin R,\ p \in Q\}$.

Well-Specified FA

Definition: Let $M = (Q, \Sigma, R, s, F)$ be a **complete DFA**. Then, M is ***well-specified FA*** (WSFA) if:

1) Q has no inaccessible state

2) Q has no more than one nonterminating state

Note: If well-specified FA has one nonterminating state, then it is q_{false} from the previous algorithm.

Theorem: For every FA M, there is an equivalent WSFA M_{ws}.

Proof: Use the next algorithm.

Algorithm: FA to WSFA

- **Input:** FA M
- **Output:** WSFA M_{ws}

- **Method:**
- convert a FA M to an equivalent ε-free FA M'
- convert a M' to an equivalent DFA M_d without any inaccessible state
- convert M_d to an equivalent DFA M_t without any nonterminating state
- convert M_t to an equivalent complete FA M_c
- $M_{ws} := M_c$

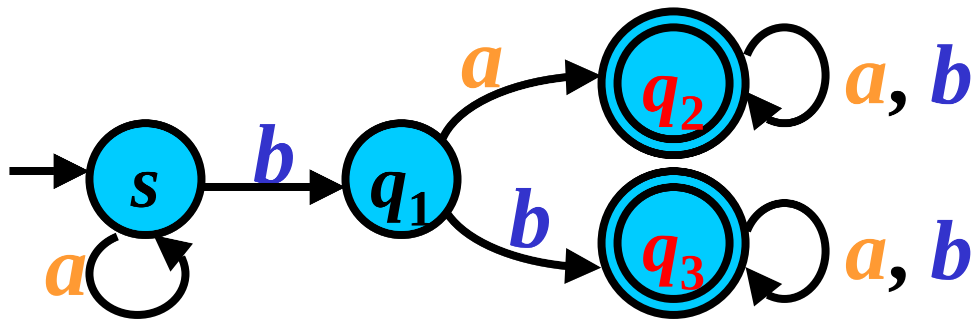
Note: No more than one nonterminating state in M_{ws}—q_{false}

Distinguishable States

Gist: String *w distinguishes* states *p* and *q* if WSFA reaches a final state from precisely one of configurations *pw* and *qw*.

Definition: Let $M = (Q, \Sigma, R, s, F)$ be a WSFA, and let $p, q \in Q$, $p \neq q$. States p and q are *distinguishable* if there exists $w \in \Sigma^*$ such that: $pw \vdash^* p'$ and $qw \vdash^* q'$, where $p', q' \in Q$ and $((p' \in F$ and $q' \notin F)$ or $(p' \notin F$ and $q' \in F))$; otherwise, states p and q are *indistinguishable*

Distinguishable States: Example



- s and q_1 are **distinguishable**, because for $w = a$:

$$sa \vdash s \ , s \notin F$$
$$q_1 a \vdash q_2 \ , q_2 \in F$$

- q_2 and q_3 are **indistinguishable**, because for each $w \in \Sigma^*$:

$$q_2 w \vdash^* q_2, \ q_2 \in F$$
$$q_3 w \vdash^* q_3, \ q_3 \in F$$

- Other pairs of states are trivially **distinguishable** for $w = \varepsilon$.

Minimum-State FA

Definition: Let M be a **WSFA**. Then, M is ***minimum-state FA*** if M contains only distinguishable states.

Theorem: For every WSFA M, there is an equivalent minimum-state FA M_m

Proof: Use the next algorithm.

Algorithm: WSFA to Min-State FA

- **Input:** WSFA $M = (Q, \Sigma, R, s, F)$
- **Output:** Minimum-State FA $M_m = (Q_m, \Sigma, R_m, s_m, F_m)$
- **Method:**
- $Q_m = \{\{p: p \in F\}, \{q: q \in Q - F\}\}$;
- **repeat**

  if there exist $X \in Q_m$, $d \in \Sigma$, $X_1, X_2 \subset X$ such that

  $X = X_1 \cup X_2$, $X_1 \cap X_2 = \emptyset$ **and**

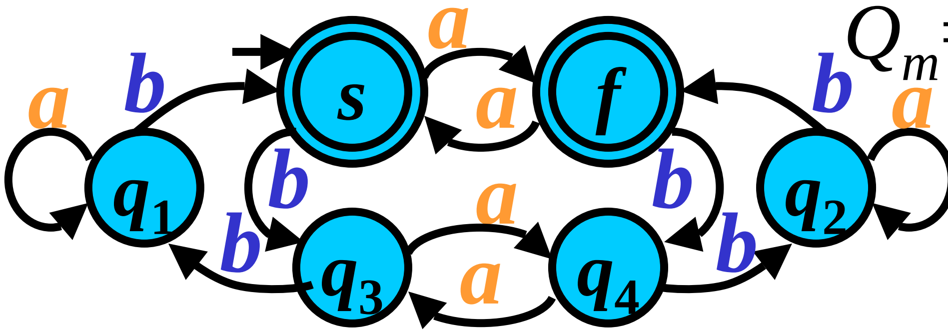
  $\{q_1: p_1 \in X_1, p_1 d \to q_1 \in R\} \subseteq Q_1$, $Q_1 \in Q_m$,

  $\{q_2: p_2 \in X_2, p_2 d \to q_2 \in R\} \cap Q_1 = \emptyset$

  then divide X into X_1 and X_2 in Q_m

  until no division is possible;
- $R_m = \{ Xa \to Y: X, Y \in Q_m, pa \to q \in R, p \in X, q \in Y, a \in \Sigma\}$;
- $s_m = X$ with $s \in X$; $F_m := \{X: X \in Q_m, X \cap F \neq \emptyset\}$.

Minimization: Example 1/4



$Q_m = \{\{s, f\}, \{q_1, q_2, q_3, q_4\}\}$

Final States | **Other States**

1) $X = \{s, f\}$:

$d = a$: $sa \rightarrow f$, $fa \rightarrow s$ — **From one set**

$d = b$: $sb \rightarrow q_3$, $fb \rightarrow q_4$ — **From one set**

2) $X = \{q_1, q_2, q_3, q_4\}$:

$d = a$: $q_1a \rightarrow q_1$, $q_2a \rightarrow q_2$, $q_3a \rightarrow q_4$, $q_4a \rightarrow q_3$ — **From one set**

$d = b$: $q_1b \rightarrow s$, $q_2b \rightarrow f$ — $\subseteq Q_1 = \{s, f\}$

$q_3b \rightarrow q_1$, $q_4b \rightarrow q_2$ — $\{q_1, q_2\} \cap Q_1 = \emptyset$

Division: $\{q_1, q_2, q_3, q_4\} \Rightarrow \{q_1, q_2\}, \{q_3, q_4\}$, where $X_1 = \{q_1, q_2\}$ and $X_2 = \{q_3, q_4\}$

Minimization: Example 2/4

$Q_m = \{\{s,f\}, \{q_1,q_2\}, \{q_3,q_4\}\}$

1) $X = \{s,f\}$:

From one set

$d = a$: $sa \rightarrow f$, $fa \rightarrow s$

From one set

$d = b$: $sb \rightarrow q_3$, $fb \rightarrow q_4$

2) $X = \{q_1,q_2\}$:

From one set

$d = a$: $q_1a \rightarrow q_1$, $q_2a \rightarrow q_2$

From one set

$d = b$: $q_1b \rightarrow s$, $q_2b \rightarrow f$

3) $X = \{q_3,q_4\}$:

From one set

$d = a$: $q_3a \rightarrow q_3$, $q_4a \rightarrow q_4$

From one set

$d = b$: $q_3b \rightarrow q_1$, $q_4b \rightarrow q_2$

No next divisions !!!

Minimization: Example 3/4

$Q_m = \{\{s,f\}, \{q_1,q_2\}, \{q_3,q_4\}\}$

$sa \rightarrow f \in R$:
$fa \rightarrow s \in R$: $\} \Rightarrow \{s,f\}a \rightarrow \{s,f\} \in R_m$

$sb \rightarrow q_3 \in R$:
$fb \rightarrow q_4 \in R$: $\} \Rightarrow \{s,f\}b \rightarrow \{q_3,q_4\} \in R_m$

$q_1a \rightarrow q_1 \in R$:
$q_2a \rightarrow q_2 \in R$: $\} \Rightarrow \{q_1,q_2\}a \rightarrow \{q_1,q_2\} \in R_m$

$q_1b \rightarrow s \in R$:
$q_2b \rightarrow f \in R$: $\} \Rightarrow \{q_1,q_2\}b \rightarrow \{s,f\} \in R_m$

$q_3a \rightarrow q_3 \in R$:
$q_4a \rightarrow q_4 \in R$: $\} \Rightarrow \{q_3,q_4\}a \rightarrow \{q_3,q_4\} \in R_m$

$q_3b \rightarrow q_1 \in R$:
$q_4b \rightarrow q_2 \in R$: $\} \Rightarrow \{q_3,q_4\}b \rightarrow \{q_1,q_2\} \in R_m$

Minimization: Example 4/4

$s \in \{s,f\} \Rightarrow s_m := \{s,f\}$

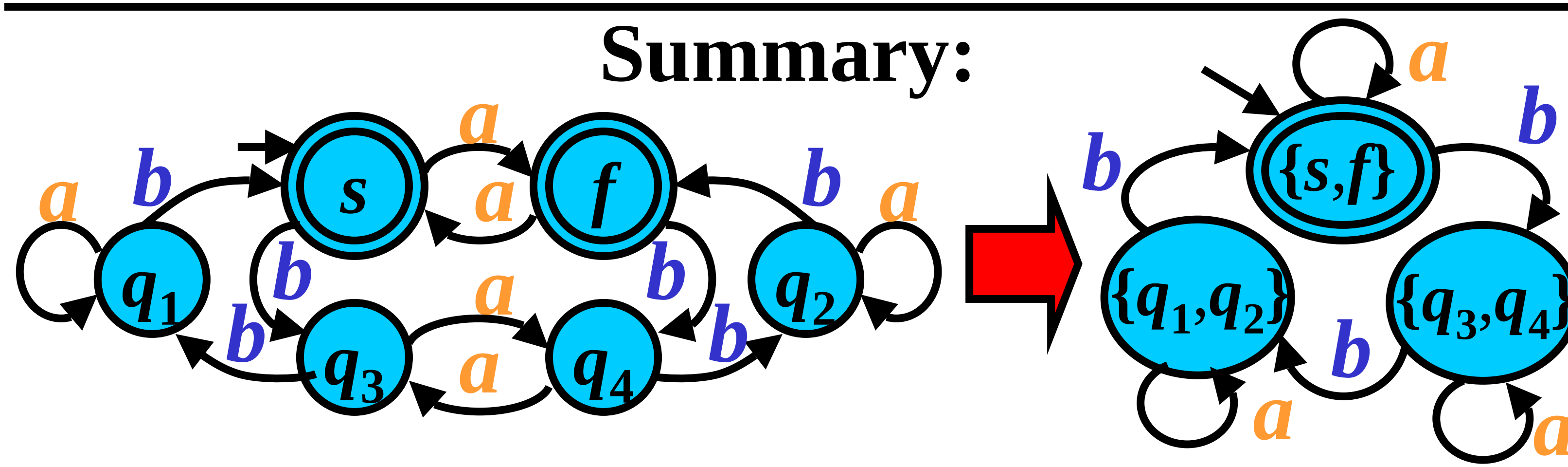
$s \in F$, $f \in F$ $\Rightarrow$ $\{s,f\} \in F_m$

$M_m = (Q_m, \Sigma, R_m, s_m, F_m)$, where: $\Sigma = \{a, b\}$, $s_m = \{s,f\}$

$Q_m = \{\{s,f\}, \{q_1,q_2\}, \{q_3,q_4\}\}$, $F_m = \{\{s,f\}\}$

$R_m = \{\{s,f\}a \rightarrow \{s,f\}, \{s,f\}b \rightarrow \{q_3,q_4\}, \{q_1,q_2\}a \rightarrow \{q_1,q_2\}, \{q_1,q_2\}b \rightarrow \{s,f\}, \{q_3,q_4\}a \rightarrow \{q_3,q_4\}, \{q_3,q_4\}b \rightarrow \{q_1,q_2\}\}$

Summary:



Variants of FA: Summary

	FA	ε-free FA	DFA	Complete FA	WSFA	Min-State FA
Number of rules of the form $p \to q$, where $p, q \in Q$	**0-n**	**0**	**0**	**0**	**0**	**0**
Number of rules of the form $pa \to q$, for any $p \in Q$, $a \in \Sigma$	**0-n**	**0-n**	**0-1**	**1**	**1**	**1**
Number of inaccessible states	**0-n**	**0-n**	**0-n**	**0-n**	**0**	**0**
Number of nonterminating states	**0-n**	**0-n**	**0-n**	**0-n**	**0-1**	**0-1**
Number of this FAs for any regular language.	∞	∞	∞	∞	∞	**1**