DFA $\rightleftarrows$ NFA

↓

regular language

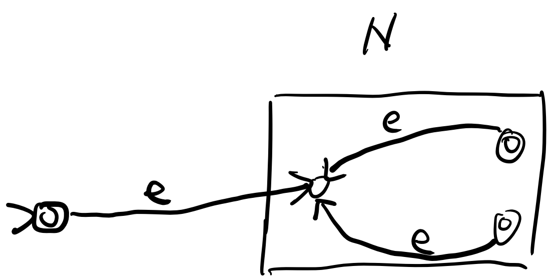
↓

closure property: union. intersection. complement
concatenation. star

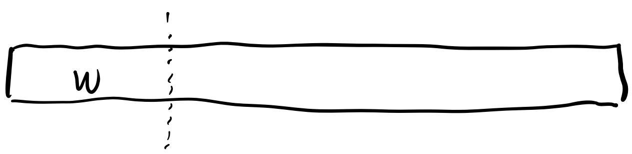
Theorem

If A is regular, so is A^*

Idea



$e \in A^*$

input 

Proof.

Let $N = (\Sigma, K, \Delta, s, F)$ be a NFA accepting A.

Construct $N' = (\Sigma, K', \Delta', s', F')$ as follows.

$K' = K \cup \{s'\}$

$F' = F \cup \{s'\}$

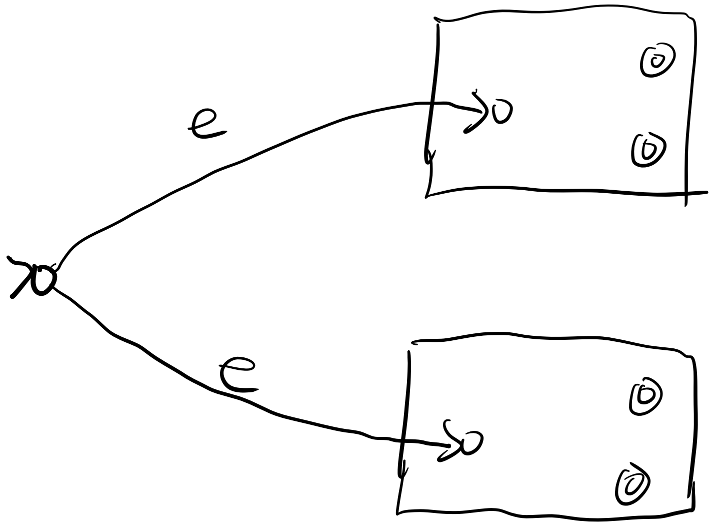
$\Delta' = \Delta \cup \{(s', e, s)\} \cup \{(q, e, s) : q \in F\}$

N' accept A^*

Theorem

If A and B are regular. So is $A \cup B$.

Proof



finite automata: language recoganizer

language generator: grammar

regular language $\longrightarrow$ regular grammar
regular expression

arithmetic $(2+3)\times 5$

regular expression $(a \cup b)^* a$ R

$L(R) = (\{a\} \cup \{b\})^* \circ \{a\}$

$= \{ w \in \{a, b\}^* : w \text{ ends with } a \}$

R decribes L(R)

Definition

The regular expression over an alphabet Σ are defined inductively.

Basis

1. ϕ is a regular expression,

   $L(\phi) = \phi$

2. any symbol $a \in \Sigma$ is a regular expression

   $L(a) = \{a\}$

Induction

3. If R_1 and R_2 are regular expression, so is $(R_1 \cup R_2)$

   $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$

4. " ", so is $(R_1 R_2)$

   $L(R_1 R_2) = L(R_1) \circ L(R_2)$

5. If R is a regular expression, so is (R^*)

   $L(R^*) = (L(R))^*$

$$(((a^*) b) \cup ((b^*) a))$$ $$a^* b \cup b^* a$$

Precedence: star > concatenation > union

Example

$\{e\}$ ϕ^*

$\{w \in \{a,b\}^* : w$ starts with a and ends with $b\}$

$a(a\cup b)^*b$

$\{w \in \{0,1\}^* : w$ has at least two occurrence of $0\}$

$(0\cup 1)^* 0 (0\cup 1)^* 0 (0\cup 1)^*$

Theorem

A language is regular iff it is decribed by some regular expression.

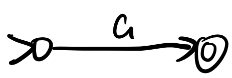
Idea

$$\begin{tikzcd} \text{R.E} \arrow[r, bend left] & \text{NFA} \arrow[l, bend left] \arrow[r, bend left] & \text{DFA} \arrow[l, bend left] \end{tikzcd}$$

R.E $\Longrightarrow$ NFA

$(ab\cup a)^*$

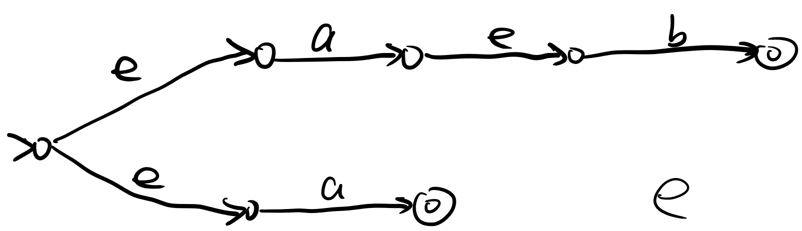
a:



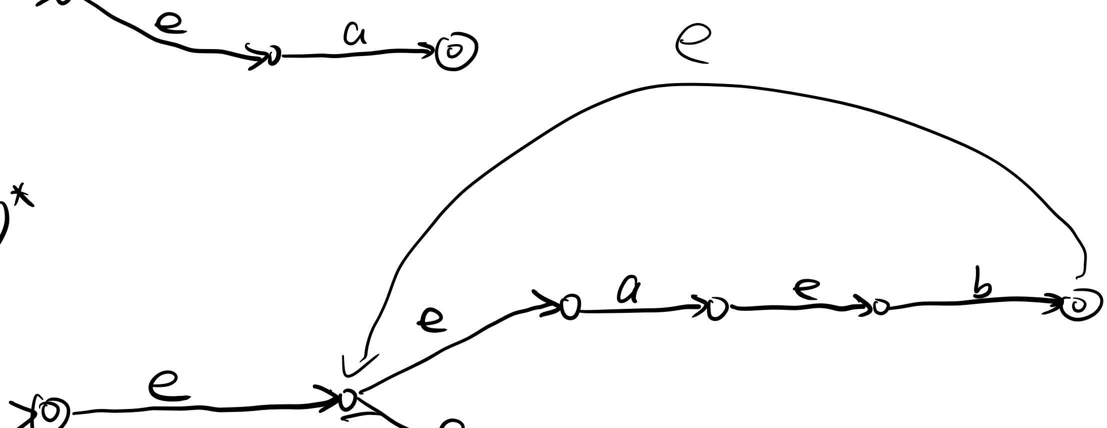
b:

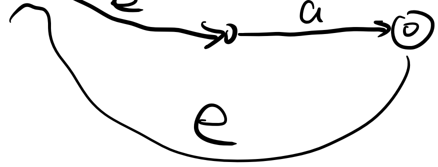
ab:

abUa:



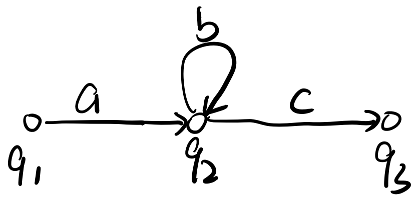
$(ab\cup a)^*$

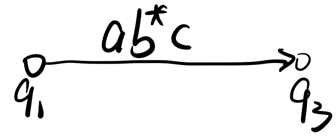




NFA ⟹ regular expression

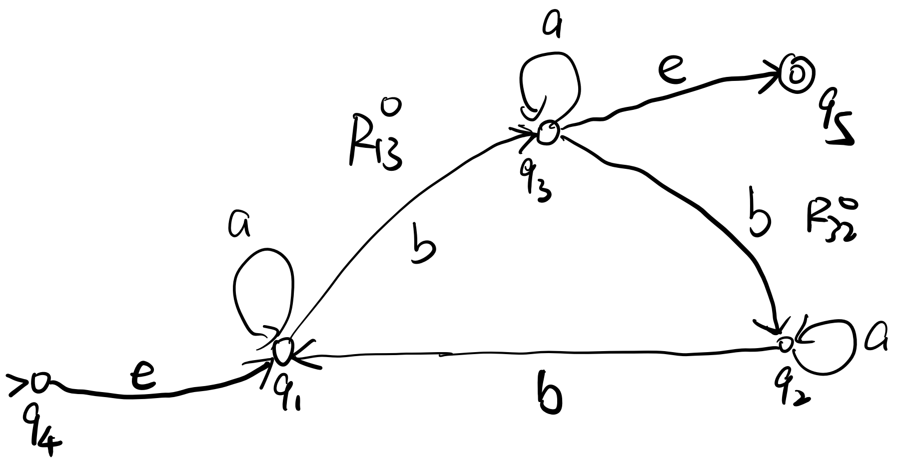
Elimination of states





Given a NFA N,

1. convert N into an equivalent NFA N′ such that
   a) N′ has no arc entering its initial state.
   b) N′ has only one final state, and there is no arc leaving this final state.
2. eliminate, one by one, all the states except the initial state and the final state.



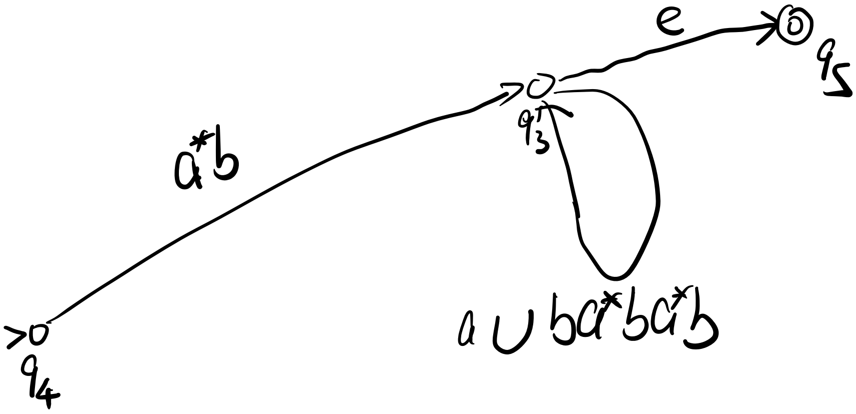
$q_4 \xrightarrow{e} q_1 \xrightarrow{b} q_3$ (a)

$q_2 \xrightarrow{b} q_1 \xrightarrow{b} q_3$ (a)

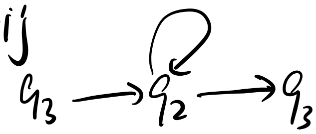
$q_2 \to q_1 \to \cancel{q_3} \to q_5$

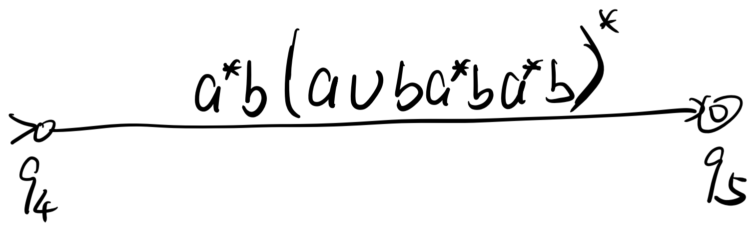
R^1_{23}

R_{ij}



R^2_{ij}





R^3_{ij}

Let $N = (K, \Sigma, \Delta, s, F)$ be a NFA.

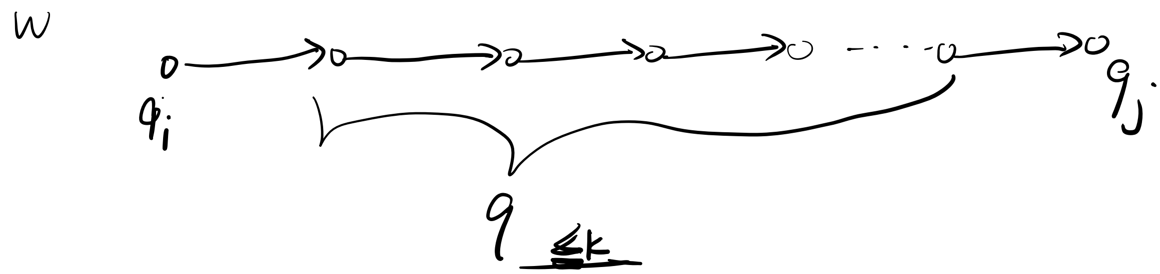
WLOG, assume that

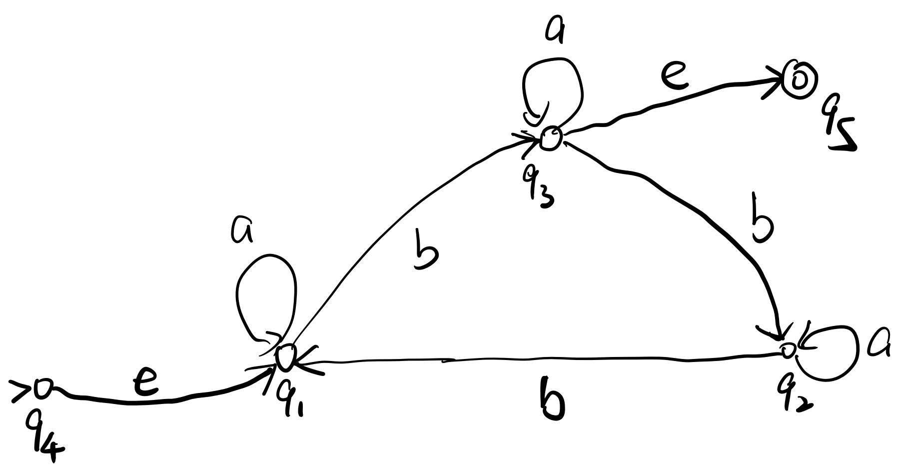
1. $K = \{q_1, q_2, \ldots, q_n\}$, $s = q_{n-1}$, $F = \{q_n\}$
2. $(p, a, q_{n-1}) \notin \Delta$ for any $p \in K$, $a \in \Sigma \cup \{e\}$
3. $(q_n, a, p) \notin \Delta$ for any $p \in K$, $a \in \Sigma \cup \{e\}$

Subproblem

for every $i, j \in [1, n]$, for every $k \in [0, n]$, define

$L^k_{ij} = \{ w \in \Sigma^* :$ w drives N from q_i to q_j without passing any intermediate state having index greater than $k.\}$

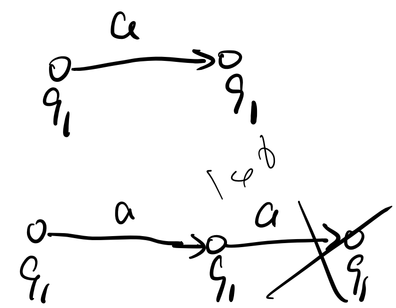




$L^0_{41} = \{e\}$ $\quad$ $L^0_{13} = \{b\}$

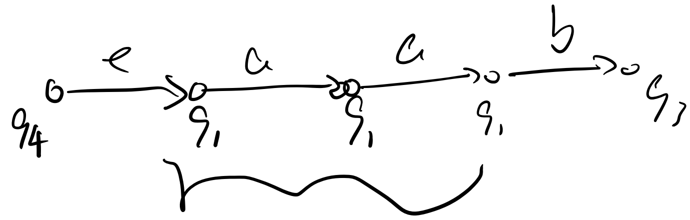
$L^0_{44} = \{e\}$ $\quad$ $L^0_{11} = \{e, a\}$

$L^0_{43} = \emptyset$



$L^1_{43} = \{b, ab, aab, aaab, \ldots\ldots\}$

$R^k_{ij} \longrightarrow L^k_{ij}$



Goal

$L(N) = L^{n-2}_{(n-1)n}$ $\quad$ $R^{n-2}_{(n-1)n}$

Base case $k = 0$

if $i \neq j$,

$L^0_{ij} = \{a : (q_i, a, q_j) \in \Delta\}$

R^0_{ij} is easy to obtain

if $i=j$

$$L^0_{ij} = \{a : (q_i, a, q_j) \in \Delta\} \cup \{e\}$$

Recurrence $k \geq 1$

$$L^k_{ij} = L^{k-1}_{ij} \cup L^{k-1}_{ik}\left(L^{k-1}_{kk}\right)^* L^{k-1}_{kj}$$

$$R^k_{ij} = R^{k-1}_{ij} \cup R^{k-1}_{ik}\left(R^{k-1}_{kk}\right)^* R^{k-1}_{kj}$$