Lecture 7 – Equivalence of Regular Expressions and Finite Automata

COSE215: Theory of Computation

Jihyeok Park

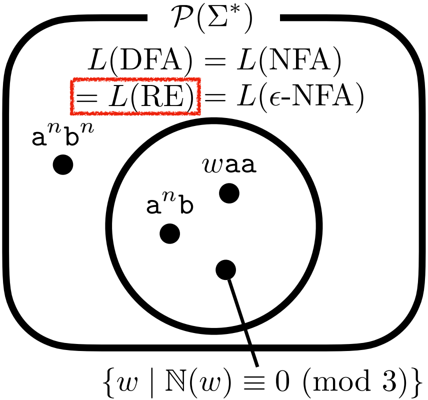
PLRG

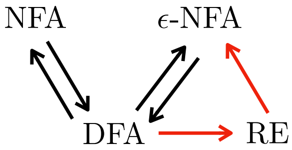
2024 Spring

Recall

- Regular Expressions
  - Operations in languages
  - Definition
  - Precedence order
  - Language of regular expressions
  - Extended regular expressions
  - Examples
- Regular Expressions in Practice

Equivalence of REs and FA





Contents

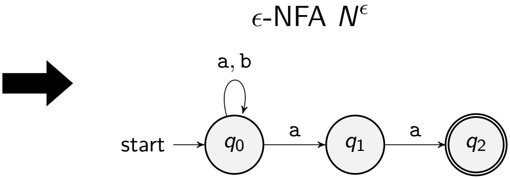
Contents

Regular Expressions to ϵ-NFA

Theorem (Regular Expressions to ϵ-NFA)

For a given regular expression R, $\exists$ ϵ-NFA N^ϵ. $L(R) = L(N^\epsilon)$.

$\mathtt{(a|b)^*aa}$



Regular Expressions to ϵ-NFA

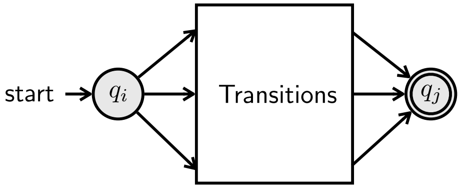
For a given regular expression R and an integer i, we will construct an ϵ-NFA $N^\epsilon = (Q, \Sigma, \delta, q_i, F)$ that accepts the language of R.

Regular Expressions to ϵ-NFA

For a given regular expression R and an integer i, we will construct an ϵ-NFA $N^\epsilon = (Q, \Sigma, \delta, q_i, F)$ that accepts the language of R.

It satisfies the following properties:

- States are q_i, q_{i+1}, $\cdots$, and q_j ($Q = \{q_k \mid i \leq k \leq j\}$)
- The last state is the unique final state ($F = \{q_j\}$)
- No transition to the initial state ($\forall q \in Q.\ \forall x \in \Sigma \cup \{\epsilon\}.\ q_i \notin \delta(q, x)$)
- No transition from the final state ($\forall x \in \Sigma \cup \{\epsilon\}.\ \delta(q_j, x) = \varnothing$)

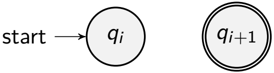


ϵ-NFA for (R, i)

Regular Expressions to ϵ-NFA

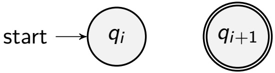
For a given regular expression R and an integer i, the ϵ-NFA for (R, i) is:

- $R = \varnothing$:

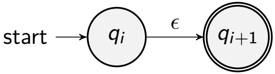


For a given regular expression R and an integer i, the ϵ-NFA for (R, i) is:

- $R = \varnothing$:



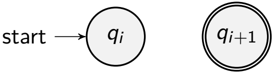
- $R = \epsilon$:



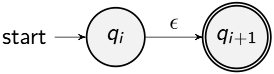
Regular Expressions to ϵ-NFA

For a given regular expression R and an integer i, the ϵ-NFA for (R, i) is:

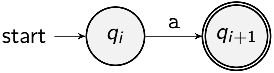
- $R = \varnothing$:



- $R = \epsilon$:

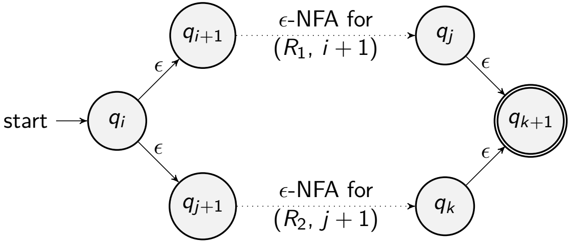


- $R = \mathtt{a}$:



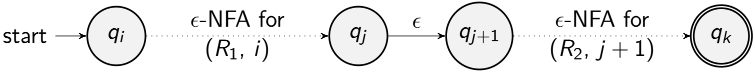
Regular Expressions to ϵ-NFA

- $R = R_1 \mid R_2$:



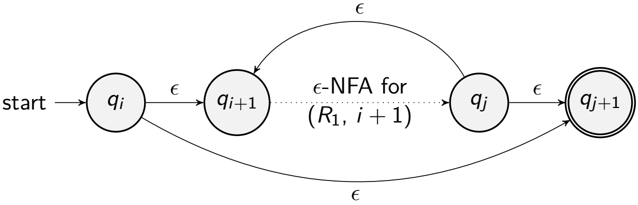
Regular Expressions to ϵ-NFA

- $R = R_1 \ R_2$:



Regular Expressions to ϵ-NFA

- $R = R_1^*$:

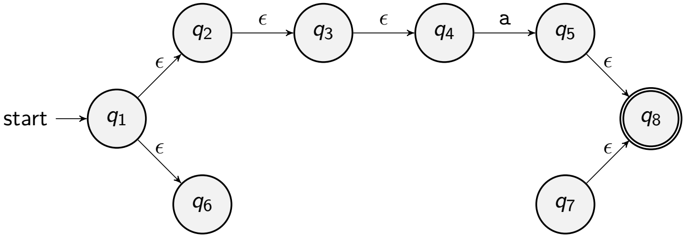


Regular Expressions to ϵ-NFA – Examples

- $R = \epsilon \texttt{a} \mid \varnothing$

Regular Expressions to ϵ-NFA – Examples

- $R = \epsilon \texttt{a} \,|\, \varnothing$

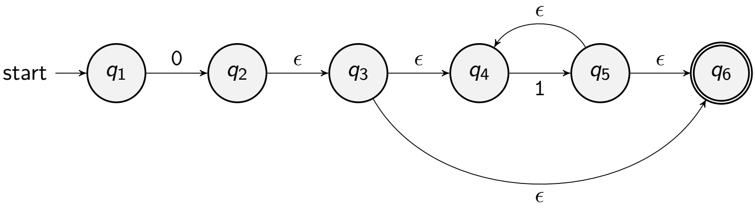


Regular Expressions to ϵ-NFA – Examples

- $R = 01^*$

Regular Expressions to ϵ-NFA – Examples

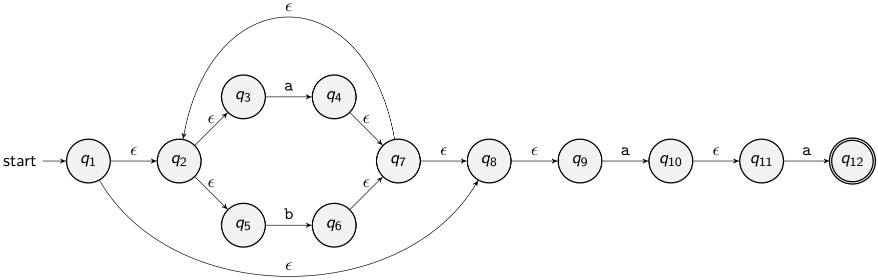
- $R = 01^*$



Regular Expressions to ϵ-NFA – Examples

- $R = \texttt{(a|b)}^*\texttt{aa}$

- $R = \texttt{(a|b)}^*\texttt{aa}$



Contents

DFA to Regular Expressions

Theorem (DFA to Regular Expressions)

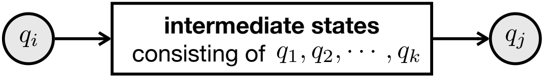
For a given DFA $D = (Q, \Sigma, \delta, q_1, F)$*,* $\exists$ *RE* R*.* $L(D) = L(R)$ *where* $Q = \{q_1, q_2, \cdots, q_n\}$.

We will learn two different way to convert a DFA to a regular expression.

1. **Inductive Construction** of **Regular Expressions** for paths in a DFA with bounded intermediate states
2. **State Elimination Method** in an extended DFA using regular expressions as labels

Inductive Construction of Regular Expressions

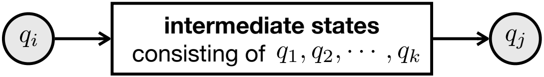
Let $R_{i,j}^{(k)}$ be the **regular expression** that accepts the **paths** from q_i to q_j whose indices of the **intermediate** states are **bounded** by k.



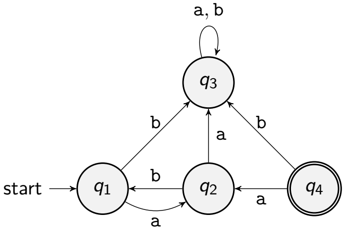
For example, $R_{1,3}^{(2)}$ is the regular expression that accepts the paths from q_1 to q_3 whose intermediate states are q_1 and q_2.

Inductive Construction of Regular Expressions

Let $R_{i,j}^{(k)}$ be the **regular expression** that accepts the **paths** from q_i to q_j whose indices of the **intermediate** states are **bounded** by k.

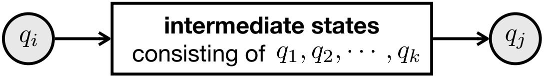


For example, $R_{1,3}^{(2)}$ is the regular expression that accepts the paths from q_1 to q_3 whose intermediate states are q_1 and q_2.

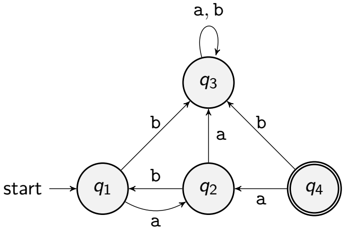


Inductive Construction of Regular Expressions

Let $R_{i,j}^{(k)}$ be the **regular expression** that accepts the **paths** from q_i to q_j whose indices of the **intermediate** states are **bounded** by k.



For example, $R_{1,3}^{(2)}$ is the regular expression that accepts the paths from q_1 to q_3 whose intermediate states are q_1 and q_2.



$$L(R_{1,3}^{(2)}) \ni \quad \texttt{b} \quad \texttt{a a} \quad \texttt{a b a a}$$

$$\not\ni \quad \texttt{a} \quad \texttt{b a} \quad \texttt{a a b}$$

Inductive Construction of Regular Expressions

PLRG

We can **inductively construct** regular expressions $R_{i,j}^{(k)}$ for all combination of i, j, and k (induction on k).

Inductive Construction of Regular Expressions

We can **inductively construct** regular expressions $R_{i,j}^{(k)}$ for all combination of i, j, and k (induction on k).

- **(Basis Case)** $k = 0$

  It means that **no intermediate states** in the path.

  - If $i \neq j$ (source and destination states are different),

$$R_{i,j}^{(0)} = \mathtt{a}_1 \,|\, \mathtt{a}_2 \,|\, \cdots \,|\, \mathtt{a}_m$$

where $q_i \xrightarrow{\mathtt{a}_1} q_j, q_i \xrightarrow{\mathtt{a}_2} q_j, \cdots, q_i \xrightarrow{\mathtt{a}_m} q_j$ are transitions in D.

  - If $i = j$ (source and destination states are same),

$$R_{i,j}^{(0)} = R_{i,i}^{(0)} = \epsilon \,|\, \mathtt{a}_1 \,|\, \mathtt{a}_2 \,|\, \cdots \,|\, \mathtt{a}_m$$

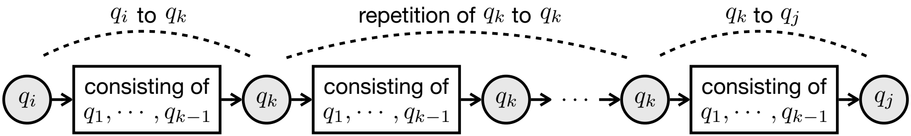
where $q_i \xrightarrow{\mathtt{a}_1} q_i, q_i \xrightarrow{\mathtt{a}_2} q_i, \cdots, q_i \xrightarrow{\mathtt{a}_m} q_i$ are transitions in D.

Inductive Construction of Regular Expressions

- **(Induction Case)** $R_{i,j}^{(k-1)}$ are given for all i and j.

$$R_{i,j}^{(k)} = R_{i,j}^{(k-1)} \mid R_{i,k}^{(k-1)}(R_{k,k}^{(k-1)})^* R_{k,j}^{(k-1)}$$

  - $R_{i,j}^{(k-1)}$: paths from q_i to q_j **NOT** containing q_k as intermediate states.
  - $R_{i,k}^{(k-1)}(R_{k,k}^{(k-1)})^* R_{k,j}^{(k-1)}$: paths from q_i to q_j containing q_k at least once as intermediate states.



Inductive Construction of Regular Expressions

Consider the following DFA:

$$D = (Q, \Sigma, \delta, q_1, F)$$

where $Q = \{q_1, q_2, \cdots, q_n\}$ and $F = \{q_{f_1}, q_{f_2}, \cdots, q_{f_m}\}$.

Inductive Construction of Regular Expressions

Consider the following DFA:

$$D = (Q, \Sigma, \delta, q_1, F)$$

where $Q = \{q_1, q_2, \cdots, q_n\}$ and $F = \{q_{f_1}, q_{f_2}, \cdots, q_{f_m}\}$.

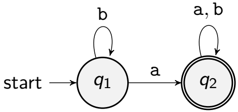
Then, using the regular expressions $R^{(k)}_{i,j}$ with bounded intermediate states, we can construct the regular expression R that accepts the language of the DFA D as follows:

$$R = R^{(n)}_{1,f_1} \mid R^{(n)}_{1,f_2} \mid \cdots \mid R^{(n)}_{1,f_m}$$

The regular expression R accepts all the paths from the **initial state** q_1 to one of the **final states** q_{f_1}, q_{f_2}, $\cdots$, q_{f_m} but **no bound** on the intermediate states (because $k = n$).

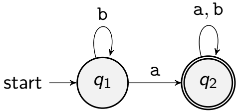
Examples

Let's construct the regular expressions $R_{i,j}^{(k)}$ for the following DFA:



Examples

Let's construct the regular expressions $R_{i,j}^{(k)}$ for the following DFA:



When $k = 0$, we have:

- $R_{1,1}^{(0)} = \epsilon \,|\, \texttt{b}$
- $R_{1,2}^{(0)} = \texttt{a}$
- $R_{2,1}^{(0)} = \varnothing$
- $R_{2,2}^{(0)} = \epsilon \,|\, \texttt{a} \,|\, \texttt{b}$

Examples

When $k = 0$, we have:

- $R_{1,1}^{(0)} = \epsilon \,|\, \mathtt{b}$
- $R_{1,2}^{(0)} = \mathtt{a}$
- $R_{2,1}^{(0)} = \varnothing$
- $R_{2,2}^{(0)} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

Examples

When $k = 0$, we have:

- $R_{1,1}^{(0)} = \epsilon \,|\, \texttt{b}$
- $R_{1,2}^{(0)} = \texttt{a}$
- $R_{2,1}^{(0)} = \varnothing$
- $R_{2,2}^{(0)} = \epsilon \,|\, \texttt{a} \,|\, \texttt{b}$

When $k = 1$, we have:

- $R_{1,1}^{(1)} = R_{1,1}^{(0)} \;|\; R_{1,1}^{(0)}(R_{1,1}^{(0)})^* R_{1,1}^{(0)} = (R_{1,1}^{(0)})^+ = (\epsilon \,|\, \texttt{b})^+ = \texttt{b}^*$

Examples

When $k = 0$, we have:

- $R^{(0)}_{1,1} = \epsilon \,|\, \mathtt{b}$
- $R^{(0)}_{1,2} = \mathtt{a}$
- $R^{(0)}_{2,1} = \varnothing$
- $R^{(0)}_{2,2} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

When $k = 1$, we have:

- $R^{(1)}_{1,1} = R^{(0)}_{1,1} \mid R^{(0)}_{1,1}(R^{(0)}_{1,1})^* R^{(0)}_{1,1} = (R^{(0)}_{1,1})^+ = (\epsilon \,|\, \mathtt{b})^+ = \mathtt{b}^*$
- $R^{(1)}_{1,2} = R^{(0)}_{1,2} \mid R^{(0)}_{1,1}(R^{(0)}_{1,1})^* R^{(0)}_{1,2} = (R^{(0)}_{1,1})^* R^{(0)}_{1,2} = (\epsilon \,|\, \mathtt{b})^* \mathtt{a} = \mathtt{b}^* \mathtt{a}$

Examples

When $k = 0$, we have:

- $R_{1,1}^{(0)} = \epsilon \,|\, \mathtt{b}$
- $R_{1,2}^{(0)} = \mathtt{a}$
- $R_{2,1}^{(0)} = \varnothing$
- $R_{2,2}^{(0)} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

When $k = 1$, we have:

- $R_{1,1}^{(1)} = R_{1,1}^{(0)} \mid R_{1,1}^{(0)}(R_{1,1}^{(0)})^* R_{1,1}^{(0)} = (R_{1,1}^{(0)})^+ = (\epsilon \,|\, \mathtt{b})^+ = \mathtt{b}^*$
- $R_{1,2}^{(1)} = R_{1,2}^{(0)} \mid R_{1,1}^{(0)}(R_{1,1}^{(0)})^* R_{1,2}^{(0)} = (R_{1,1}^{(0)})^* R_{1,2}^{(0)} = (\epsilon \,|\, \mathtt{b})^* \mathtt{a} = \mathtt{b}^* \mathtt{a}$
- $R_{2,1}^{(1)} = R_{2,1}^{(0)} \mid R_{2,1}^{(0)}(R_{1,1}^{(0)})^* R_{1,1}^{(0)} = R_{2,1}^{(0)}(R_{1,1}^{(0)})^* = \varnothing(\epsilon \,|\, \mathtt{b})^* = \varnothing$

Examples

When $k = 0$, we have:

- $R_{1,1}^{(0)} = \epsilon \,|\, \mathtt{b}$
- $R_{1,2}^{(0)} = \mathtt{a}$
- $R_{2,1}^{(0)} = \varnothing$
- $R_{2,2}^{(0)} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

When $k = 1$, we have:

- $R_{1,1}^{(1)} = R_{1,1}^{(0)} \mid R_{1,1}^{(0)}(R_{1,1}^{(0)})^* R_{1,1}^{(0)} = (R_{1,1}^{(0)})^+ = (\epsilon \,|\, \mathtt{b})^+ = \mathtt{b}^*$
- $R_{1,2}^{(1)} = R_{1,2}^{(0)} \mid R_{1,1}^{(0)}(R_{1,1}^{(0)})^* R_{1,2}^{(0)} = (R_{1,1}^{(0)})^* R_{1,2}^{(0)} = (\epsilon \,|\, \mathtt{b})^* \mathtt{a} = \mathtt{b}^* \mathtt{a}$
- $R_{2,1}^{(1)} = R_{2,1}^{(0)} \mid R_{2,1}^{(0)}(R_{1,1}^{(0)})^* R_{1,1}^{(0)} = R_{2,1}^{(0)}(R_{1,1}^{(0)})^* = \varnothing(\epsilon \,|\, \mathtt{b})^* = \varnothing$
- $R_{2,2}^{(1)} = R_{2,2}^{(0)} \mid R_{2,1}^{(0)}(R_{1,1}^{(0)})^* R_{1,2}^{(0)} = R_{2,2}^{(0)} \mid \varnothing = R_{2,2}^{(0)} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

Examples

When $k = 1$, we have:

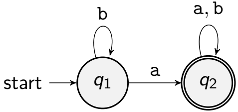
- $R^{(1)}_{1,1} = \mathtt{b}^*$
- $R^{(1)}_{1,2} = \mathtt{b}^*\mathtt{a}$
- $R^{(1)}_{2,1} = \varnothing$
- $R^{(1)}_{2,2} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

Examples

When $k = 1$, we have:

- $R^{(1)}_{1,1} = \mathtt{b}^*$
- $R^{(1)}_{1,2} = \mathtt{b}^*\mathtt{a}$
- $R^{(1)}_{2,1} = \varnothing$
- $R^{(1)}_{2,2} = \epsilon \mid \mathtt{a} \mid \mathtt{b}$

Let's focus on the regular expression for the language of the DFA.

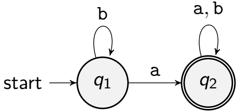


Examples

When $k = 1$, we have:

- $R_{1,1}^{(1)} = \texttt{b}^*$
- $R_{1,2}^{(1)} = \texttt{b}^*\texttt{a}$
- $R_{2,1}^{(1)} = \varnothing$
- $R_{2,2}^{(1)} = \epsilon \,|\, \texttt{a} \,|\, \texttt{b}$

Let's focus on the regular expression for the language of the DFA.



We need to compute $R_{1,2}^{(2)}$ using the results when $k = 1$:

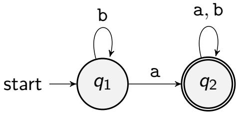
- $R_{1,2}^{(2)} \quad =$

Examples

When $k = 1$, we have:

- $R_{1,1}^{(1)} = \mathtt{b}^*$
- $R_{1,2}^{(1)} = \mathtt{b}^*\mathtt{a}$
- $R_{2,1}^{(1)} = \varnothing$
- $R_{2,2}^{(1)} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

Let's focus on the regular expression for the language of the DFA.



We need to compute $R_{1,2}^{(2)}$ using the results when $k = 1$:

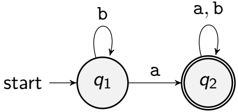
- $R_{1,2}^{(2)} \quad = \quad R_{1,2}^{(1)} \mid R_{1,2}^{(1)}(R_{2,2}^{(1)})^* R_{2,2}^{(1)} =$

Examples

When $k = 1$, we have:

- $R_{1,1}^{(1)} = \mathtt{b}^*$
- $R_{1,2}^{(1)} = \mathtt{b}^*\mathtt{a}$
- $R_{2,1}^{(1)} = \varnothing$
- $R_{2,2}^{(1)} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

Let's focus on the regular expression for the language of the DFA.



We need to compute $R_{1,2}^{(2)}$ using the results when $k = 1$:

- $R_{1,2}^{(2)} \quad = \quad R_{1,2}^{(1)} \mid R_{1,2}^{(1)}(R_{2,2}^{(1)})^* R_{2,2}^{(1)} = R_{1,2}^{(1)}(R_{2,2}^{(1)})^*$

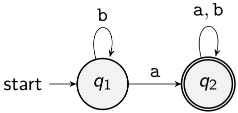
$\qquad\quad =$

Examples

When $k = 1$, we have:

- $R^{(1)}_{1,1} = \mathtt{b}^*$
- $R^{(1)}_{1,2} = \mathtt{b}^*\mathtt{a}$
- $R^{(1)}_{2,1} = \varnothing$
- $R^{(1)}_{2,2} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

Let's focus on the regular expression for the language of the DFA.



We need to compute $R^{(2)}_{1,2}$ using the results when $k = 1$:

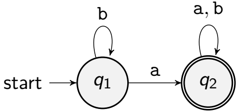
- $$\begin{aligned} R^{(2)}_{1,2} &= R^{(1)}_{1,2} \mid R^{(1)}_{1,2}(R^{(1)}_{2,2})^* R^{(1)}_{2,2} = R^{(1)}_{1,2}(R^{(1)}_{2,2})^* \\ &= \mathtt{b}^*\mathtt{a}(\epsilon \,|\, \mathtt{a} \,|\, \mathtt{b})^* \\ &= \end{aligned}$$

Examples

When $k = 1$, we have:

- $R^{(1)}_{1,1} = \mathtt{b}^*$
- $R^{(1)}_{1,2} = \mathtt{b}^*\mathtt{a}$
- $R^{(1)}_{2,1} = \varnothing$
- $R^{(1)}_{2,2} = \epsilon \,|\, \mathtt{a} \,|\, \mathtt{b}$

Let's focus on the regular expression for the language of the DFA.



We need to compute $R^{(2)}_{1,2}$ using the results when $k = 1$:

$$\begin{aligned} \bullet\ R^{(2)}_{1,2} &= R^{(1)}_{1,2} \,|\, R^{(1)}_{1,2}(R^{(1)}_{2,2})^* R^{(1)}_{2,2} = R^{(1)}_{1,2}(R^{(1)}_{2,2})^* \\ &= \mathtt{b}^*\mathtt{a}(\epsilon \,|\, \mathtt{a} \,|\, \mathtt{b})^* \\ &= \mathtt{b}^*\mathtt{a}(\mathtt{a} \,|\, \mathtt{b})^* \qquad \text{(the regular expression for the above DFA)} \end{aligned}$$

State Elimination Method

- There is another way to convert a DFA to a regular expression.

State Elimination Method

- There is another way to convert a DFA to a regular expression.
- It is **more intuitive** and easier to understand but **not easy to implement**.

State Elimination Method

- There is another way to convert a DFA to a regular expression.
- It is **more intuitive** and easier to understand but **not easy to implement**.
- The idea is to **eliminate** the **states** of the DFA one by one and **construct the regular expressions**.

State Elimination Method

- There is another way to convert a DFA to a regular expression.
- It is **more intuitive** and easier to understand but **not easy to implement**.
- The idea is to **eliminate** the **states** of the DFA one by one and **construct the regular expressions**.
- We will assign constructed **regular expressions** instead of symbols as **labels** on the transitions between the states in the DFA.

State Elimination Method

We can convert a DFA to a regular expression using the following steps for **each final state** $q_f \in F$:

State Elimination Method

We can convert a DFA to a regular expression using the following steps for **each final state** $q_f \in F$:

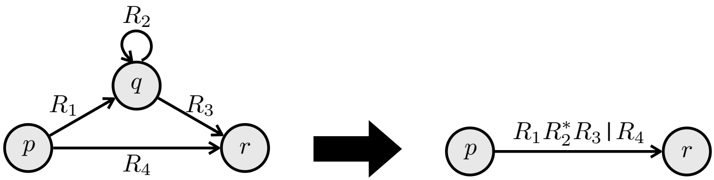
1. **Merge** symbols x_1, x_2, $\cdots$, x_m on the transition from q_i to q_j into a single regular expression $x_1 \,|\, x_2 \,|\, \cdots \,|\, x_m$.

State Elimination Method

We can convert a DFA to a regular expression using the following steps for **each final state** $q_f \in F$:

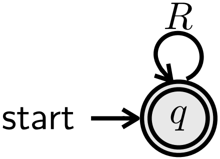
1. **Merge** symbols $x_1, x_2, \cdots, x_m$ on the transition from q_i to q_j into a single regular expression $x_1 \mid x_2 \mid \cdots \mid x_m$.

2. **Eliminate** a state q that is **not** the **initial** state or the **target final** state using the following mechanism:

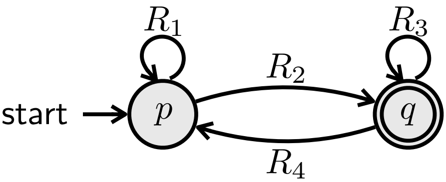


State Elimination Method

❸ **Construct** regular expressions for the remaining one or two states using the regular expressions on the transitions between the states.



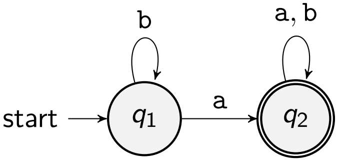
R^*



$(R_1 \mid R_2 R_3^* R_4)^* R_2 R_3^*$

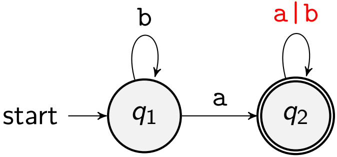
Example 1

Let's convert the following DFA to a regular expression using the **state elimination** method:



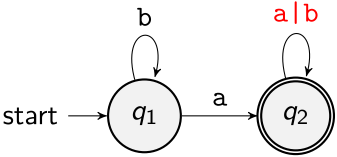
Example 1

Let's convert the following DFA to a regular expression using the **state elimination** method:



Example 1

Let's convert the following DFA to a regular expression using the **state elimination** method:

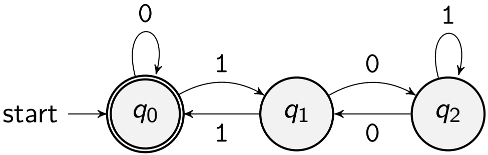


The regular expression for the above DFA is:

$$\texttt{b}^*\texttt{a}(\texttt{a|b})^*$$

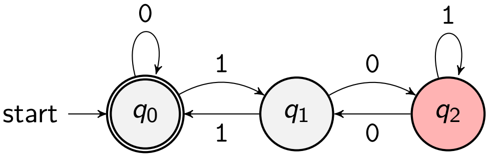
Example 2

The following DFA accepts $L = \{w \in \{0,1\}^* \mid \mathbb{N}(w) \equiv 0 \pmod 3\}$ where $\mathbb{N}(w)$ is the natural number represented by w in binary:



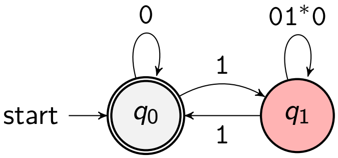
Example 2

The following DFA accepts $L = \{w \in \{0, 1\}^* \mid \mathbb{N}(w) \equiv 0 \pmod 3\}$ where $\mathbb{N}(w)$ is the natural number represented by w in binary:



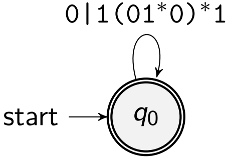
Example 2

The following DFA accepts $L = \{w \in \{0,1\}^* \mid \mathbb{N}(w) \equiv 0 \pmod 3\}$ where $\mathbb{N}(w)$ is the natural number represented by w in binary:



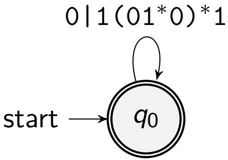
Example 2

The following DFA accepts $L = \{w \in \{0, 1\}^* \mid \mathbb{N}(w) \equiv 0 \pmod 3\}$ where $\mathbb{N}(w)$ is the natural number represented by w in binary:



Example 2

The following DFA accepts $L = \{w \in \{0,1\}^* \mid \mathbb{N}(w) \equiv 0 \pmod 3\}$ where $\mathbb{N}(w)$ is the natural number represented by w in binary:

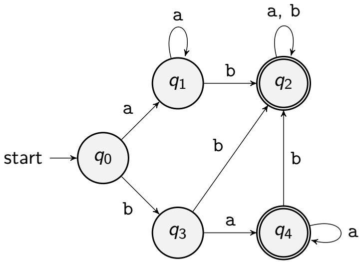


Then, the regular expression for the above DFA is:

$$\texttt{(0|1(01}^*\texttt{0)}^*\texttt{1)}^*$$

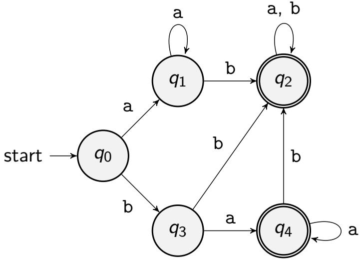
Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:



Example 3

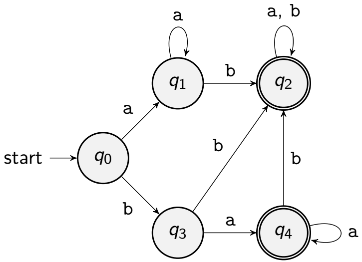
Let's convert the following DFA to a regular expression using the **state elimination** method:



We need to consider two final states q_2 and q_4.

Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:

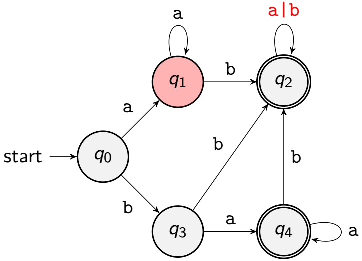


We need to consider two final states q_2 and q_4.

Let's start by eliminating non-final states.

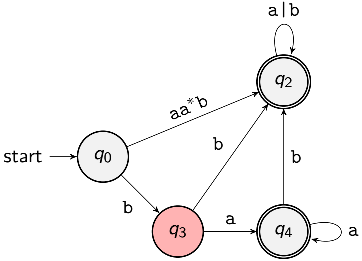
Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:



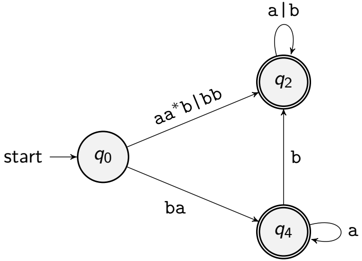
Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:



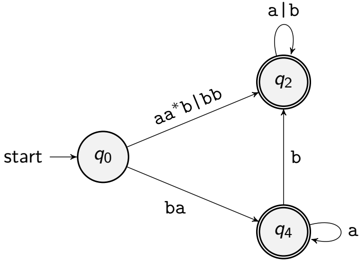
Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:



Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:

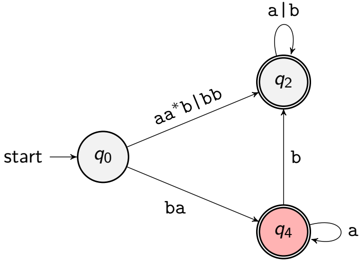


Now, we know that the regular expression for the **final state** q_4 is:

$$\texttt{baa}^*$$

Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:



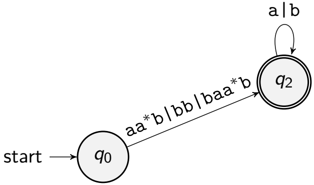
Now, we know that the regular expression for the **final state** q_4 is:

$$\texttt{baa}^*$$

Let's keep eliminating q_4 to know the regular expression for q_2.

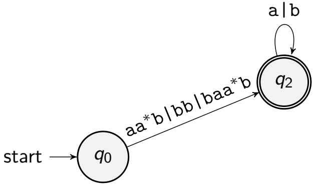
Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:



Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:

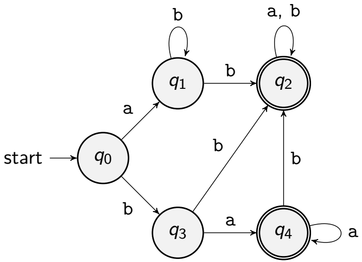


Now, we know that the regular expression for the **final state** q_2 is:

$$\texttt{(ab}^*\texttt{b|bb|baa}^*\texttt{b)(a|b)}^*$$

Example 3

Let's convert the following DFA to a regular expression using the **state elimination** method:

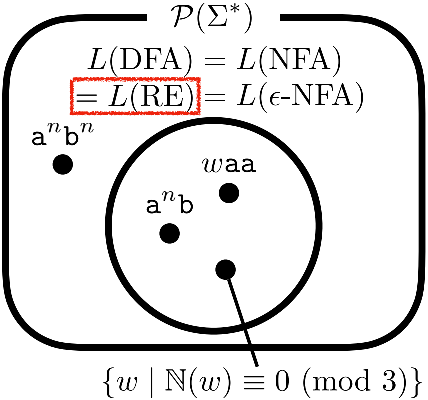


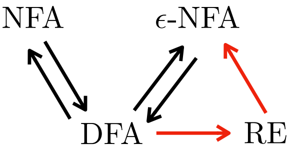
Finally, we have the regular expression for the above DFA:

$$\texttt{(aa}^*\texttt{b|bb|baa}^*\texttt{b)(a|b)}^* \texttt{|baa}^*$$

Note that $\texttt{(aa}^*\texttt{b|bb|baa}^*\texttt{b)(a|b)}^*$ is for q_2 and $\texttt{baa}^*$ is for q_4.

Summary


$\mathcal{P}(\Sigma^*)$
$L(\text{DFA}) = L(\text{NFA})$
$\boxed{= L(\text{RE})} = L(\epsilon\text{-NFA})$
$\texttt{a}^n\texttt{b}^n$
$w\texttt{aa}$
$\texttt{a}^n\texttt{b}$
$\{w \mid \mathbb{N}(w) \equiv 0 \pmod 3\}$


NFA
ϵ-NFA
DFA
RE

Next Lecture

- Properties of Regular Languages

Jihyeok Park
jihyeok_park@korea.ac.kr
https://plrg.korea.ac.kr