

Lecture 8 – Closure Properties of Regular Languages

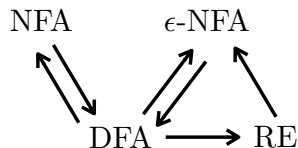
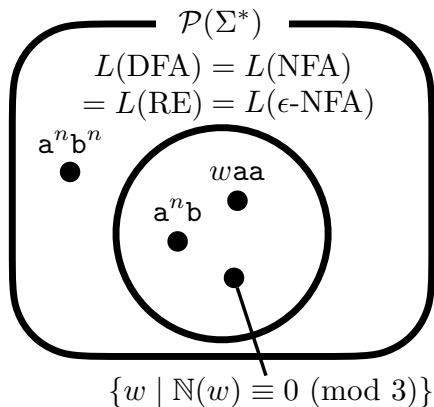
COSE215: Theory of Computation

Jihyeok Park



2024 Spring

- Regular Languages



1. Closure Properties of Regular Languages

Union, Concatenation, and Kleene Star

Complement

Intersection

Difference

Reversal

Homomorphism

Inverse Homomorphism

Closure Properties of Regular Languages

Let's consider a **regular** language. For example,

$$L = \{waa \mid w \in \{a, b\}^*\}$$

Closure Properties of Regular Languages

Let's consider a **regular** language. For example,

$$L = \{waa \mid w \in \{a, b\}^*\}$$

Then, is its **reverse** language L^R also **regular**?

$$L^R = \{aaw \mid w \in \{a, b\}^*\}$$

Let's consider a **regular** language. For example,

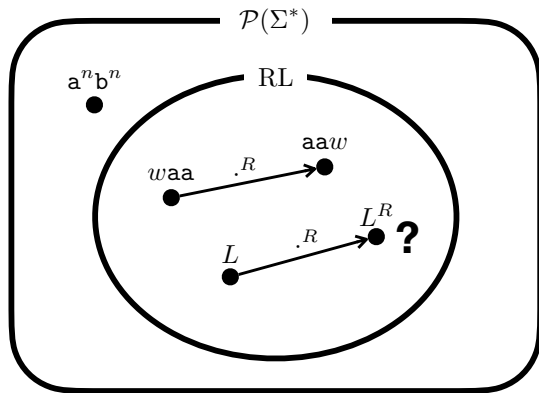
$$L = \{waa \mid w \in \{a, b\}^*\}$$

Then, is its **reverse** language L^R also **regular**?

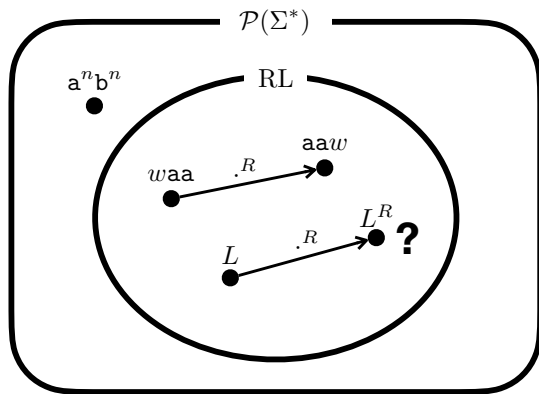
$$L^R = \{aaw \mid w \in \{a, b\}^*\}$$

Yes! We can construct a regular expression whose language is L^R as:

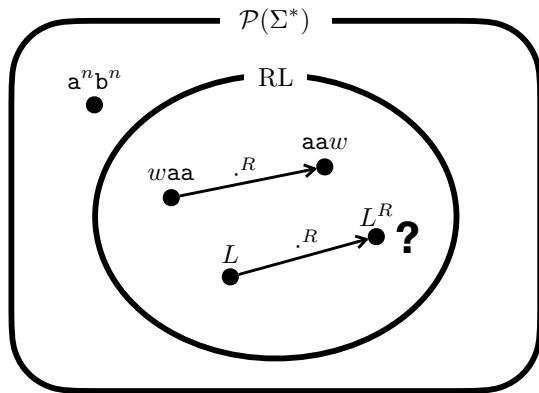
$$L(aa(a|b)^*) = L^R = \{aaw \mid w \in \{a, b\}^*\}$$



Then, for any regular language L , is L^R always regular?

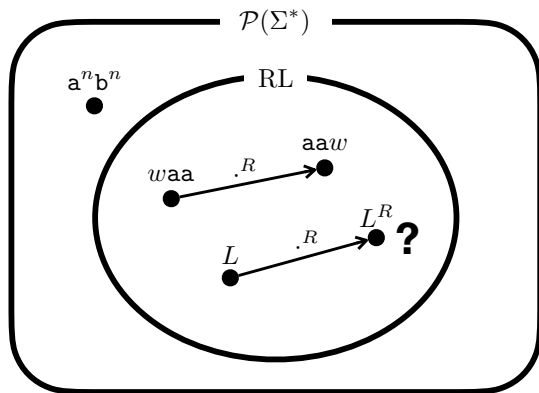


Then, for any regular language L , is L^R always regular? **Yes!**



Then, for any regular language L , is L^R always regular? **Yes!**

The class of regular languages is **closed** under the **reversal** operator.



Then, for any regular language L , is L^R always regular? **Yes!**

The class of regular languages is **closed** under the **reversal** operator.

In this lecture, we will discuss and prove the **closure properties** of regular languages for various language operators.

Definition (Closure Properties)

The class of regular languages is **closed** under an n -ary operator op if and only if $op(L_1, \dots, L_n)$ is regular for any regular languages L_1, \dots, L_n . We say that such properties are **closure properties** of regular languages.

Definition (Closure Properties)

The class of regular languages is **closed** under an n -ary operator op if and only if $op(L_1, \dots, L_n)$ is regular for any regular languages L_1, \dots, L_n . We say that such properties are **closure properties** of regular languages.

To prove the closure properties for the n -ary operator op , we need to provide a way to do the following for any regular languages L_1, \dots, L_n :

Definition (Closure Properties)

The class of regular languages is **closed** under an n -ary operator op if and only if $op(L_1, \dots, L_n)$ is regular for any regular languages L_1, \dots, L_n . We say that such properties are **closure properties** of regular languages.

To prove the closure properties for the n -ary operator op , we need to provide a way to do the following for any regular languages L_1, \dots, L_n :

- 1 Construct a regular expression R whose language is $L(R) = op(L(R_1), \dots, L(R_n))$ for any regular expressions R_1, \dots, R_n .

Definition (Closure Properties)

The class of regular languages is **closed** under an n -ary operator op if and only if $op(L_1, \dots, L_n)$ is regular for any regular languages L_1, \dots, L_n . We say that such properties are **closure properties** of regular languages.

To prove the closure properties for the n -ary operator op , we need to provide a way to do the following for any regular languages L_1, \dots, L_n :

- 1 Construct a regular expression R whose language is $L(R) = op(L(R_1), \dots, L(R_n))$ for any regular expressions R_1, \dots, R_n .
- 2 Construct a finite automaton A whose language is $L(A) = op(L(A_1), \dots, L(A_n))$ for any finite automata A_1, \dots, A_n .

In this lecture, we will prove the closure properties of regular languages for the following operators:

- Union
- Concatenation
- Kleene Star
- Complement
- Intersection
- Difference
- Reversal
- Homomorphism
- Inverse Homomorphism

Theorem (Closure under Union, Concatenation, and Kleene Star)

If L_1 and L_2 are regular languages, then so is $L_1 \cup L_2$, L_1L_2 , and L_1^ .*

Theorem (Closure under Union, Concatenation, and Kleene Star)

If L_1 and L_2 are regular languages, then so is $L_1 \cup L_2$, L_1L_2 , and L_1^ .*

Proof) Let R_1 and R_2 be the regular expressions such that $L(R_1) = L_1$ and $L(R_2) = L_2$, respectively.

Theorem (Closure under Union, Concatenation, and Kleene Star)

If L_1 and L_2 are regular languages, then so is $L_1 \cup L_2$, L_1L_2 , and L_1^ .*

Proof) Let R_1 and R_2 be the regular expressions such that $L(R_1) = L_1$ and $L(R_2) = L_2$, respectively.

Consider the following regular expression:

$$R_1 \mid R_2$$

$$R_1R_2$$

$$R^*$$

Theorem (Closure under Union, Concatenation, and Kleene Star)

If L_1 and L_2 are regular languages, then so is $L_1 \cup L_2$, L_1L_2 , and L_1^* .

Proof) Let R_1 and R_2 be the regular expressions such that $L(R_1) = L_1$ and $L(R_2) = L_2$, respectively.

Consider the following regular expression:

$$R_1 | R_2 \qquad R_1 R_2 \qquad R^*$$

Then, by the definition of the union (\cup), concatenation (\cdot), and Kleene star ($*$) operators for regular expressions,

$$L(R_1 | R_2) = L_1 \cup L_2 \qquad L(R_1 R_2) = L_1 L_2 \qquad L(R^*) = L^*$$

Theorem (Closure under Union, Concatenation, and Kleene Star)

If L_1 and L_2 are regular languages, then so is $L_1 \cup L_2$, L_1L_2 , and L_1^* .

Proof) Let R_1 and R_2 be the regular expressions such that $L(R_1) = L_1$ and $L(R_2) = L_2$, respectively.

Consider the following regular expression:

$$R_1 | R_2 \qquad R_1 R_2 \qquad R^*$$

Then, by the definition of the union (\cup), concatenation (\cdot), and Kleene star ($*$) operators for regular expressions,

$$L(R_1 | R_2) = L_1 \cup L_2 \qquad L(R_1 R_2) = L_1 L_2 \qquad L(R^*) = L^*$$

So, we proved that the class of regular languages are **closed** under the **union**, **concatenation**, and **Kleene star** operators. □

Closure under Complement

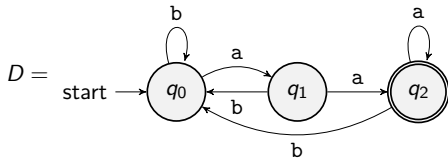
Consider a regular language $L = \{waa \mid w \in \{a, b\}^*\}$.

Is its **complement** $\bar{L} = \Sigma^* \setminus L$ also regular?

Closure under Complement

Consider a regular language $L = \{waa \mid w \in \{a, b\}^*\}$.

Is its **complement** $\bar{L} = \Sigma^* \setminus L$ also regular? **Yes!**

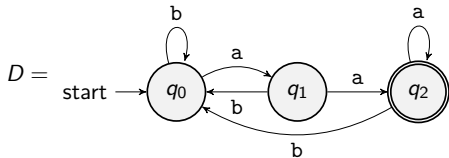


First, consider the above DFA D accepting the language L .

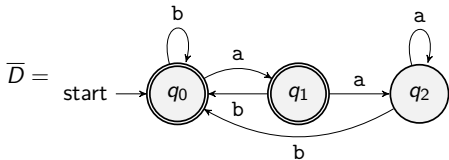
Closure under Complement

Consider a regular language $L = \{waa \mid w \in \{a, b\}^*\}$.

Is its **complement** $\bar{L} = \Sigma^* \setminus L$ also regular? **Yes!**



First, consider the above DFA D accepting the language L .



The key idea is to construct a new DFA \bar{D} by **swapping** the **final** and **non-final** states of the original DFA:

Theorem (Closure under Complement)

If L is a regular language, then so is \bar{L} .

Proof) Let $D = (Q, \Sigma, \delta, q_0, F)$ be the DFA such that $L(D) = L$. Consider the following DFA:

$$\bar{D} = (Q, \Sigma, \delta, q_0, Q \setminus F).$$

Then,

$$\begin{aligned} \forall w \in \Sigma^*, w \in L(\bar{D}) &\iff \delta^*(q_0, w) \in Q \setminus F \\ &\iff \delta^*(q_0, w) \notin F \\ &\iff w \notin L(D) \\ &\iff w \notin L \\ &\iff w \in \bar{L} \end{aligned}$$

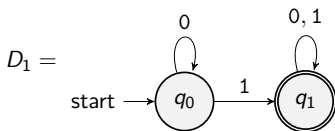
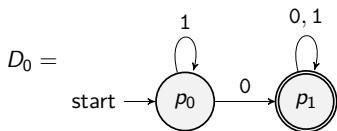
□

$$L_1 = \{w \in \{0, 1\}^* \mid w \text{ has } 0\} \quad L_2 = \{w \in \{0, 1\}^* \mid w \text{ has } 1\}$$

Is the **intersection** of two regular languages $L_1 \cap L_2$ also regular?

$$L_1 = \{w \in \{0, 1\}^* \mid w \text{ has } 0\} \quad L_2 = \{w \in \{0, 1\}^* \mid w \text{ has } 1\}$$

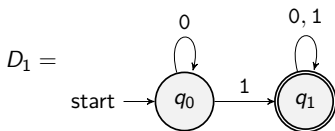
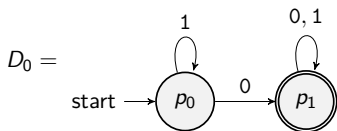
Is the **intersection** of two regular languages $L_1 \cap L_2$ also regular? **Yes!**



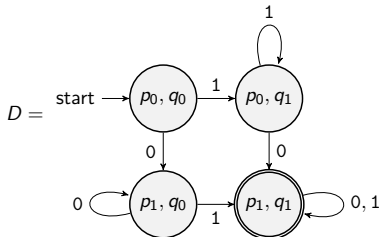
First, consider the above DFAs D_0 and D_1 accepting the languages L_1 and L_2 , respectively.

$$L_1 = \{w \in \{0, 1\}^* \mid w \text{ has } 0\} \quad L_2 = \{w \in \{0, 1\}^* \mid w \text{ has } 1\}$$

Is the **intersection** of two regular languages $L_1 \cap L_2$ also regular? **Yes!**



First, consider the above DFAs D_0 and D_1 accepting the languages L_1 and L_2 , respectively.



The key idea is to construct a new DFA D by **combining** them with their **pair of states** as its states.

Theorem (Closure under Intersection)

If L_0 and L_1 are regular languages, then so is $L_0 \cap L_1$.

Proof) Let $D_0 = (Q_0, \Sigma, \delta_0, q_0, F_0)$ and $D_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ be the DFA such that $L(D_0) = L_0$ and $L(D_1) = L_1$. Consider the following DFA:

$$D = (Q_0 \times Q_1, \Sigma, \delta, (q_0, q_1), F_0 \times F_1).$$

where $\forall q \in Q_0, q' \in Q_1, a \in \Sigma. \delta((q, q'), a) = (\delta_0(q, a), \delta_1(q', a))$.

Theorem (Closure under Intersection)

If L_0 and L_1 are regular languages, then so is $L_0 \cap L_1$.

Proof) Let $D_0 = (Q_0, \Sigma, \delta_0, q_0, F_0)$ and $D_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ be the DFA such that $L(D_0) = L_0$ and $L(D_1) = L_1$. Consider the following DFA:

$$D = (Q_0 \times Q_1, \Sigma, \delta, (q_0, q_1), F_0 \times F_1).$$

where $\forall q \in Q_0, q' \in Q_1, a \in \Sigma. \delta((q, q'), a) = (\delta_0(q, a), \delta_1(q', a))$. Then,

$$\begin{aligned} \forall w \in \Sigma^*, w \in L(D) &\iff \delta^*((q_0, q_1), w) \in F_0 \times F_1 \\ &\iff \delta^*(q_0, w) \in F_0 \text{ and } \delta^*(q_1, w) \in F_1 \\ &\iff w \in L(D_0) \text{ and } w \in L(D_1) \\ &\iff w \in L(D_0) \cap L(D_1) \\ &\iff w \in L_0 \cap L_1 \end{aligned}$$

□

Theorem (Closure under Intersection)

If L_0 and L_1 are regular languages, then so is $L_0 \cap L_1$.

Proof) Another proof is to use De Morgan's law:

$$L_0 \cap L_1 = \overline{\overline{L_0} \cup \overline{L_1}}$$

Since we already know that the regular languages are closed under complement and union, we are done. □

Theorem (Closure under Difference)

If L_0 and L_1 are regular languages, then so is $L_0 \setminus L_1$.

Proof) Similarly, we can use the following fact:

$$L_0 \setminus L_1 = L_0 \cap \overline{L_1}$$

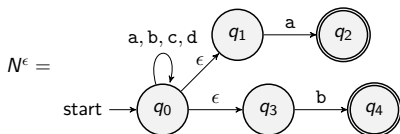
Since we already know that the regular languages are closed under complement and intersection, we are done. □

$$L = \{wa \text{ or } wb \mid w \in \{a, b, c, d\}^*\}$$

Is the **reversal** L^R of the above regular language L also regular?

$$L = \{wa \text{ or } wb \mid w \in \{a, b, c, d\}^*\}$$

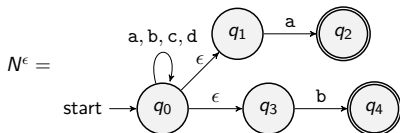
Is the **reversal** L^R of the above regular language L also regular? **Yes!**



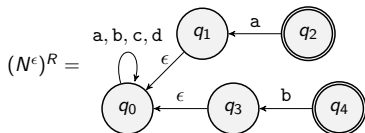
The above ϵ -NFA N^ϵ accepts the language L .

$$L = \{wa \text{ or } wb \mid w \in \{a, b, c, d\}^*\}$$

Is the **reversal** L^R of the above regular language L also regular? **Yes!**



The above ϵ -NFA N^ϵ accepts the language L .

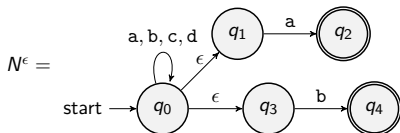


The key idea is to construct a new ϵ -NFA $(N^\epsilon)^R$ by

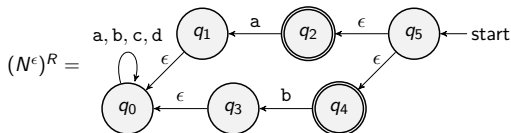
- 1 **reversing** the direction of the transitions
- 2
- 3

$$L = \{wa \text{ or } wb \mid w \in \{a, b, c, d\}^*\}$$

Is the **reversal** L^R of the above regular language L also regular? **Yes!**



The above ϵ -NFA N^ϵ accepts the language L .

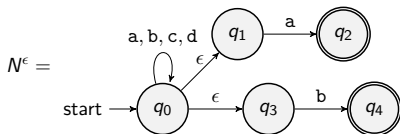


The key idea is to construct a new ϵ -NFA $(N^\epsilon)^R$ by

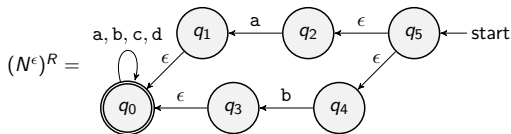
- 1 **reversing** the direction of the transitions
- 2 **adding** new initial state having ϵ -transitions to the original final states
- 3

$$L = \{wa \text{ or } wb \mid w \in \{a, b, c, d\}^*\}$$

Is the **reversal** L^R of the above regular language L also regular? **Yes!**



The above ϵ -NFA N^ϵ accepts the language L .



The key idea is to construct a new ϵ -NFA $(N^\epsilon)^R$ by

- 1 **reversing** the direction of the transitions
- 2 **adding** new initial state having ϵ -transitions to the original final states
- 3 **change** original initial state to the unique new final state

Theorem (Closure under Reversal)

If L is a regular language, then so is L^R .

Proof) Let $N^\epsilon = (Q, \Sigma, \delta, q_0, F)$ be the ϵ -NFA such that $L(N^\epsilon) = L$. Consider the following

$$(N^\epsilon)^R = (Q \uplus \{q_s\}, \Sigma, \delta^R, q_s, \{q_0\})$$

where

$$\begin{aligned} \forall q \in Q. \forall a \in \Sigma. \delta^R(q, a) &= \{q' \in Q \mid q \in \delta(q', a)\} \\ \forall q \in Q. \delta^R(q, \epsilon) &= \{q' \in Q \mid q \in \delta(q', \epsilon)\} \\ \forall a \in \Sigma. \delta^R(q_s, a) &= \emptyset \\ \delta^R(q_s, \epsilon) &= F \end{aligned}$$

□

Theorem (Closure under Reversal)

If L is a regular language, then so is L^R .

Proof) Another proof is to use the structural induction on the regular expressions. Let R be a regular expression. Then, we can define its reversed regular expression R^R as follows:

- If $R = \emptyset$, then $R^R = \emptyset$.
- If $R = \epsilon$, then $R^R = \epsilon$.
- If $R = a$, then $R^R = a$.
- If $R = R_0 \mid R_1$, then $R^R = R_0^R \mid R_1^R$.
- If $R = R_0 R_1$, then $R^R = R_1^R R_0^R$.
- If $R = R_0^*$, then $R^R = (R_0^R)^*$.
- If $R = (R_0)$, then $R^R = (R_0^R)$. □

Theorem (Closure under Reversal)

If L is a regular language, then so is L^R .

Proof) Another proof is to use the structural induction on the regular expressions. Let R be a regular expression. Then, we can define its reversed regular expression R^R as follows:

- If $R = \emptyset$, then $R^R = \emptyset$.
- If $R = \epsilon$, then $R^R = \epsilon$.
- If $R = a$, then $R^R = a$.
- If $R = R_0 | R_1$, then $R^R = R_0^R | R_1^R$.
- If $R = R_0 R_1$, then $R^R = R_1^R R_0^R$.
- If $R = R_0^*$, then $R^R = (R_0^R)^*$.
- If $R = (R_0)$, then $R^R = (R_0^R)$. □

$$R = ab(cd)^* | ef$$

$$R^R = (dc)^* ba | fe$$

Definition (Homomorphism)

Suppose Σ_0 and Σ_1 are two finite sets of symbols. Then, a function

$$h : \Sigma_0 \rightarrow \Sigma_1^*$$

is called a **homomorphism**. For a given word $w = a_1a_2 \cdots a_n \in \Sigma_0^*$,

$$h(w) = h(a_1)h(a_2) \cdots h(a_n)$$

For a language $L \subseteq \Sigma_0^*$,

$$h(L) = \{h(w) \mid w \in L\} \subseteq \Sigma_1^*$$

Definition (Homomorphism)

Suppose Σ_0 and Σ_1 are two finite sets of symbols. Then, a function

$$h : \Sigma_0 \rightarrow \Sigma_1^*$$

is called a **homomorphism**. For a given word $w = a_1a_2 \cdots a_n \in \Sigma_0^*$,

$$h(w) = h(a_1)h(a_2) \cdots h(a_n)$$

For a language $L \subseteq \Sigma_0^*$,

$$h(L) = \{h(w) \mid w \in L\} \subseteq \Sigma_1^*$$

Example

Let $\Sigma_0 = \{0, 1\}$, $\Sigma_1 = \{a, b\}$, and $h(0) = ab$, $h(1) = a$. Then,

$$h(10) = aab \quad h(010) = abaab \quad h(1100) = aaabab$$

Theorem (Closure under Homomorphism)

If h is a homomorphism and L is a regular language, then so is $h(L)$.

Proof) Let R be the regular expression such that $L(R) = L$. Then, we can define its homomorphic regular expression $h(R)$ as follows:

- If $R = \emptyset$, then $h(R) = \emptyset$.
- If $R = \epsilon$, then $h(R) = \epsilon$.
- If $R = a$, then $h(R) = h(a)$.
- If $R = R_0 | R_1$, then $h(R) = h(R_0) | h(R_1)$.
- If $R = R_0 R_1$, then $h(R) = h(R_0) h(R_1)$.
- If $R = R_0^*$, then $h(R) = (h(R_0))^*$.
- If $R = (R_0)$, then $h(R) = (h(R_0))$. \square

Theorem (Closure under Homomorphism)

If h is a homomorphism and L is a regular language, then so is $h(L)$.

Proof) Let R be the regular expression such that $L(R) = L$. Then, we can define its homomorphic regular expression $h(R)$ as follows:

- If $R = \emptyset$, then $h(R) = \emptyset$.
- If $R = \epsilon$, then $h(R) = \epsilon$.
- If $R = a$, then $h(R) = h(a)$.
- If $R = R_0 | R_1$, then $h(R) = h(R_0) | h(R_1)$.
- If $R = R_0 R_1$, then $h(R) = h(R_0) h(R_1)$.
- If $R = R_0^*$, then $h(R) = (h(R_0))^*$.
- If $R = (R_0)$, then $h(R) = (h(R_0))$. □

$$h(0) = ab$$

$$h(1) = a$$

$$R = 0(0|1)^*0^*$$

$$h(R) = ab(ab|a)^*(ab)^*$$

Definition (Inverse Homomorphism)

Suppose Σ_0 and Σ_1 are two finite sets of symbols. For a given language $L \subseteq \Sigma_1^*$ and a homomorphism $h : \Sigma_0 \rightarrow \Sigma_1^*$,

$$h^{-1}(L) = \{w \in \Sigma_0^* \mid h(w) \in L\} \subseteq \Sigma_0^*$$

Definition (Inverse Homomorphism)

Suppose Σ_0 and Σ_1 are two finite sets of symbols. For a given language $L \subseteq \Sigma_1^*$ and a homomorphism $h : \Sigma_0 \rightarrow \Sigma_1^*$,

$$h^{-1}(L) = \{w \in \Sigma_0^* \mid h(w) \in L\} \subseteq \Sigma_0^*$$

Example

Let $\Sigma_0 = \{0, 1\}$, $\Sigma_1 = \{a, b\}$, and $h(0) = ba$, $h(1) = a$. Consider the following language $L \subseteq \Sigma_1^*$:

$$L = \{waa \mid w \in \{a, b\}^*\}$$

Then, $01 \in h^{-1}(L)$ because $h(01) = baa \in L$.

However, $10 \notin h^{-1}(L)$ because $h(10) = aba \notin L$.

$$L = \{waa \mid w \in \{a, b\}^*\}$$

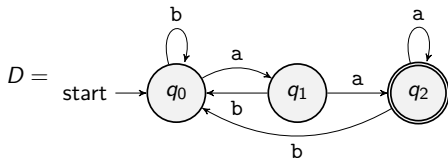
$$h : \Sigma_0 \rightarrow \Sigma_1^*. h(0) = ba \wedge h(1) = a$$

Is the **inverse homomorphism** $h^{-1}(L)$ of the above regular language L also regular ($\Sigma_0 = \{0, 1\}$ and $\Sigma_1 = \{a, b\}$)?

$$L = \{waa \mid w \in \{a, b\}^*\}$$

$$h : \Sigma_0 \rightarrow \Sigma_1^*. h(0) = ba \wedge h(1) = a$$

Is the **inverse homomorphism** $h^{-1}(L)$ of the above regular language L also regular ($\Sigma_0 = \{0, 1\}$ and $\Sigma_1 = \{a, b\}$)? **Yes!**

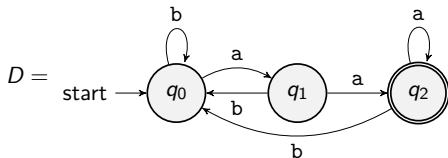


The above DFA D accepts the language L .

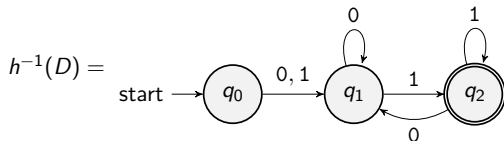
$$L = \{waa \mid w \in \{a, b\}^*\}$$

$$h : \Sigma_0 \rightarrow \Sigma_1^*. h(0) = ba \wedge h(1) = a$$

Is the **inverse homomorphism** $h^{-1}(L)$ of the above regular language L also regular ($\Sigma_0 = \{0, 1\}$ and $\Sigma_1 = \{a, b\}$)? **Yes!**



The above DFA D accepts the language L .



The key idea is to construct a new DFA $h^{-1}(D)$ by **reconstructing** the **transitions** by following the path $h(a)$ for each symbol in $a \in \Sigma_0$.

Theorem (Closure under Inverse Homomorphism)

If $h : \Sigma_0 \rightarrow \Sigma_1^*$ is a homomorphism and $L \subseteq \Sigma_1^*$ is a regular language, then so is $h^{-1}(L)$.

Proof) Let $D = (Q, \Sigma_1, \delta, q_0, F)$ be the DFA such that $L(D) = L$.

Consider the following DFA:

$$h^{-1}(D) = (Q, \Sigma_0, \delta', q_0, F).$$

where $\forall q \in Q, a \in \Sigma_0. \delta'(q, a) = \delta^*(q, h(a))$. Then, $\forall w = a_1 \cdots a_n \in \Sigma_0^*$.

$$\begin{aligned} w \in L(h^{-1}(D)) &\iff (\delta')^*(q_0, w) \in F \\ &\iff \delta'(\dots(\delta'(\delta'(q_0, a_1), a_2), \dots, a_n)) \in F \\ &\iff \delta(\dots(\delta(\delta(q_0, h(a_1)), h(a_2)), \dots, h(a_n)) \in F \\ &\iff \delta^*(q_0, h(a_1) \cdots h(a_n)) \in F \\ &\iff \delta^*(q_0, h(w)) \in F \\ &\iff h(w) \in L(D) \\ &\iff h(w) \in L \end{aligned}$$

1. Closure Properties of Regular Languages

Union, Concatenation, and Kleene Star

Complement

Intersection

Difference

Reversal

Homomorphism

Inverse Homomorphism

- Please see this document for the exercise.

<https://github.com/ku-plrg-classroom/docs/tree/main/cose215/r1-closure>

- Please implement the following functions in `Implementation.scala`.
 - `complementDFA` for the **complement** of a DFA.
 - `intersectDFA` for the **intersection** of two DFAs.
 - `reverseENFA` for the **reverse** of an ϵ -NFA.
 - `reverseRE` for the **reverse** of a regular expression.
 - `homRE` for the **homomorphism** of a regular expression.
 - `ihomDFA` for the **inverse homomorphism** of a DFA.
- It is just an exercise, and you **don't need to submit** anything.

- The Pumping Lemma for Regular Languages

Jihyeok Park

jihyeok_park@korea.ac.kr

<https://plrg.korea.ac.kr>