Chap. 4
Properties of
Regular Languages

Agenda of Chapter 4

How general are regular languages ?

What happens when we perform operations on regular languages?

How can we tell whether a given language is regular or not?

닫혀있음-

- Closure Properties of Regular Languages

  $L_1 \cup L_2$ ← Regular ?

- Elementary Questions about Regular Languages
- Identifying Nonregular Languages
  - Using the Pigeonhole Principle
  - A Pumping Lemma

Closure under Simple Set Operations (1/3)

- **Closure property** under an operation
  - For any given two elements x, y in a set S,
    x ★y is also in S ⇔ **S is closed under the operation ★**

닫혀있다.

자연수의 집합에서 + (닫혀있음) - (닫혀있지 않음).

[THEOREM 4.1]

Regular languages is closed under ∪, ∩, concatenation , ^{c} , *.

⇔ Let L_1, L_2 are regular languages.

Then so are $L_1 \cup L_2$, $L_1 \cap L_2$, L_1L_2, L_1^c and L_1^*.

Regular language 어떤 연산에 대해서 닫혀 있는지

Regular 인 것을 증명 해야 한다.

Proof)

Reg exp r_1 r_2 r_1+r_2 closure $r_1 \cdot r_2$ r_1^*

1. Closure under union, concatenation and star-closure are immediate. (Use regular expressions!)
2. Closure under complementation

   Let $M=(Q,\Sigma,\delta,q_0,F)$ be a dfa that accepts L_1.

   Then $M^c=(Q,\Sigma,\delta,q_0,Q-F)$ accepts L_1^c

nfa 이면 안된다.

total function 이기 때문에 가능.

L is Regular.
→ DFA NFA
→ Reg expression
→ Reg Grammar.

→ L_1이 accept 하는 것 reject.

⇒ final state 를 Q-F로 바꾸면 가능.

$L_1, L_2 \rightarrow$ regular languages

$\rightarrow L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 L_2$, L_1^c, $L_1^* \rightarrow$ regular language.

$r_1 + r_2$ $\quad r_1 r_2$ $\quad r_1^*$

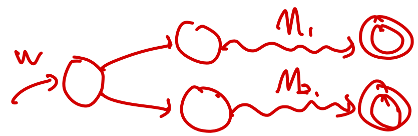
$M = (Q, \Sigma, \delta, Q_0, F) \rightarrow$ dfa accepts L_1

$M^c = (Q, \Sigma, \delta, Q_0, Q - F)$ accepts L_1^c

new dfa

Closure under Simple Set Operations (2/3)

$w \in L_1$ and $w \in L_2 \Rightarrow L_1 \cap L_2$



M_1 $\delta_1(Q_i, a) = Q_j$

M_2 $\delta_2(P_j, a) = P_l$

Proof continued)

3. Closure under intersection

Let $L_1 = L(M_1)$ and $M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$

$L_2 = L(M_2)$ and $M_2 = (P, \Sigma, \delta_2, p_0, F_2)$

M_1 accept w

M_2 accept w.

$(Q_i P_j), a$ $\quad Q_j P_l$

M'의 state

Then we can construct a combined fa M' accepting $L_1 \cap L_2$.

Set $M' = (Q', \Sigma, \delta', (q_0, p_0), F')$ where

둘 다 final이 되야 accept 한다.

Q와 P의 순서쌍

$Q' = Q \times P$, $\quad F' = \{(q_i, p_j) \mid q_i \in F_1, p_j \in F_2\}$, $F_1 \times F_2$

when $\delta_1(q_i, a) = q_k$, $\delta_2(p_j, a) = p_l$, $\delta'((q_i, p_j), a) = (q_k, p_l)$,

순서쌍을 각각 처리

Therefore, $L_1 \cap L_2$ is regular.

* const

[Another proof by using other closure properties]

L_1, L_2 regular.

$\left(\underbrace{L_1^c \cup L_2^c}_{reg} \right)^c$ reg $= L_1 \cap L_2$

$\delta'((Q_i, P_j), a) = (Q_k, P_l)$

Closure under Simple Set Operations (3/3)

Ex4.1] If L_1, L_2 are regular then L_1 - L_2 is regular.

$L_1 - L_2 = L_1 \cap L_2^c$

reg

reg

final 과 initial의 자리를 바꿔준 후

[THEOREM 4.2] Regular languages is closed under reversal.

L is regular $\Rightarrow L^R$ is regular.

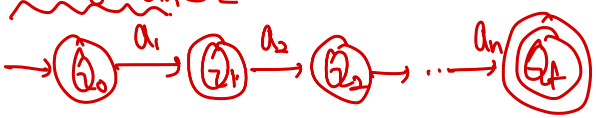
"M^R accepting L^R : $\delta_R(Q_j, a) = Q_i$" $(Q, \Sigma, \delta_R, Q_f, \{Q_0\})$.

NFA

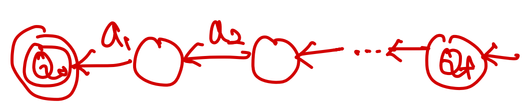
$\exists M$ accepting L : $\delta(Q_i, a) = Q_j$ $(Q, \Sigma, \delta, Q_0, \{Q_f\})$ $\longrightarrow$ $(Q, \Sigma, \delta^R, Q_f, \{Q_0\})$.

$\delta^{R*}(Q_f, a_n a_{n-1} \cdots a_2 a_1) \in \{Q_0\}$.

$a_1 a_2 a_3 \cdots a_n \in L$



$a_n a_{n-1} \cdots a_3 a_2 a_1 \in L^R$



Closure under Other Operations (1/4)

$h(a) = bb \ldots$

어떤 symbol 에서 다른 symbol 의 string 으로.

Definition] Homomorphism

- Let Σ and Γ are alphabets.
- A function $h : \Sigma \to \Gamma^*$ is called a **homomorphism**.

  When $w = a_1 a_2 \ldots a_n$, $h(w) = h(a_1)h(a_2)\ldots h(a_3)$
- "**homomorphic image**" of a language L on Σ

  $h(L) = \{h(w) \mid w \in L\}$

$h(a)=ab$ $h(b)=bbc$.

Let $L = \{aa, aba\}$

Ex4.2] $\Sigma = \{a,b\}$, $\Gamma = \{a,b,c\}$,

homomorphism $h(a)=ab$, $h(b)=bbc$

Let $L=\{aa,aba\}$

$h(aa) = h(a)\,h(a) = abab$.

$h(aba) = abbbcab$

- Then homomorphic image of L

  $h(L) = \{abab, abbbcab\}$.

$h(L) = \{h(aa), h(aba)\}$

$abab$ $\quad$ $abbbcab$

Closure under Other Operations (2/4)

[THEOREM 4.3]

L is regular ⟹ h(L) is regular

Regular languages is closed under arbitrary homomorphism.

symbol → string

Reg Exp이 존재 ⇒ Reg Language.
즉 닫힘도 유지 된다.

Ex4.3] $\Sigma = \{a,b\}$, $\Gamma = \{b,c,d\}$,

$h(a)=dbcc$, $h(b)=bdc$

- A regular language L is denoted by $r=(a+b^*)(aa)^*$
- Then h(L) : $(dbcc+(bdc)^*)(dbcc\,dbcc)^*$.
- Thus, h(L) is Regular languages

$\Sigma \to \Gamma^*$

$h(a)=dbcc \quad h(b)=bdc.$

L → denoted by $r = (a+b^*)(aa)^*.$

Then h(L) : $(dbcc+(bdc)^*)(dbccdbcc)^*.$

then h(L) is Regular language.

Closure under Other Operations (3/4)

Definition] Right quotient

- L_1, L_2 : languages on the same alphabet.
- **Right quotient** of L_1 with L_2

$$L_1/L_2 = \{x \mid xy \in L_1 \text{ for some } y \in L_2\}$$

뒤의 문자열 한개 붙은 부분을 떼버림.

Ex4.4] = { a, ~~aa~~, aaa, ab, aab, aaab, abb, ... } ∪ { ~~ba~~ }

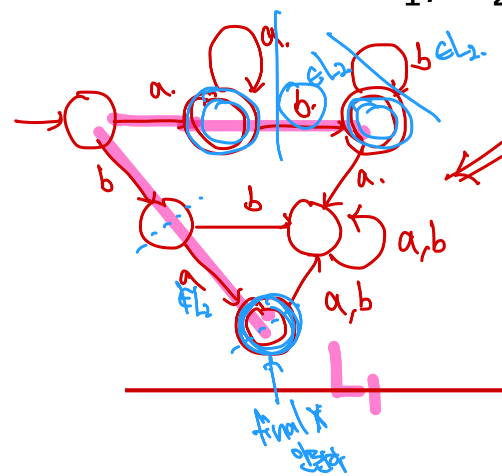
a는 반드시 1번 나옴

하나만 예로 든다. 에 의해 = {b, bb, bbb, bbbb ...}.

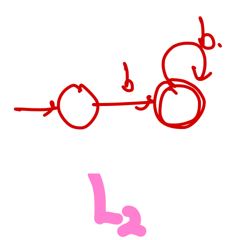
$L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$, $L_2 = \{b^m \mid m \geq 1\}$

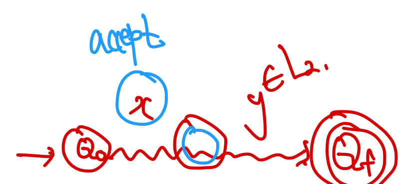
$L_1/L_2 =$ {a, aa, aaa, ab, aab, aaab, abb, aabb ...}. = $\{a^n b^m \mid n \geq 1, m \geq 0\}$.

Show that L_1, L_2 and L_1/L_2 are all regular by using dfa.



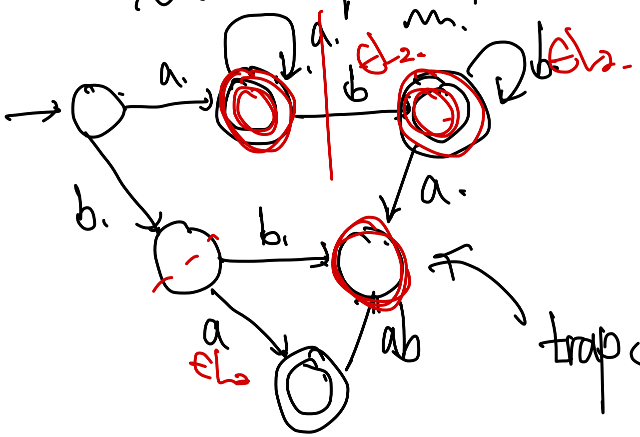
L1 그대로 → final 만 새로 그려주면 된다.



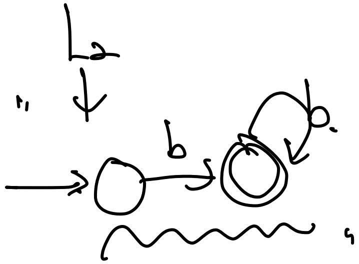


① 전체 accept 되는 때를 x y로 구분.
② 뒷부분 y∈L2 인지 확인.
③ y∈L2 이면 y 시작 state → final.

$L_1 \to$ dfa: $\{a^n b^m \mid n \ge 1, m \ge 0\} \cup \{ba\}$



trap state



Right quotient of "L_1 with L_2"

$L_1 / L_2 = \{x \mid xy \in L_1$ for some $y \in L_2\}$.

$L_1 = \{a^n b^m \mid n \ge 1, m \ge 0\} \cup \{ba\}$.

$L_2 = \{b^m \mid m \ge 1\}$ b, bb.

$L_1/L_2 = \{\cancel{a}, \cancel{aa}, \cancel{aaa}, ab, aab, aaab, abb, \ldots\}$

abb

ab

b

$\boxed{\{a^n b^m \mid n \ge 1, m \ge 0\}}$

Closure under Other Operations (4/4)

[THEOREM 4.4]

L_1, L_2 reg $\Rightarrow$ L_1/L_2 reg

→ Regular language

Regular languages are closed under right quotient with a regular languages.

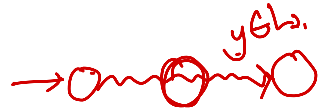
Proof) Let L_1=L(M), M =(Q,Σ,δ,q_0,F) is a dfa.

Construction of dfa M′=(Q,Σ,δ,q_0,F′) → final 만 바뀌었다.

1. For each $q_i \in Q$, consider a dfa $M_i=(Q,\Sigma,\delta,q_i,F)$ and
   If there exists a $y \in L_2 \cap L(M_i)$ then add q_i to F′.
2. Repeat this for every $q_i \in Q$. Then we finally have F′ for M′.

Verification of $L(M') = L_1/L_2$.

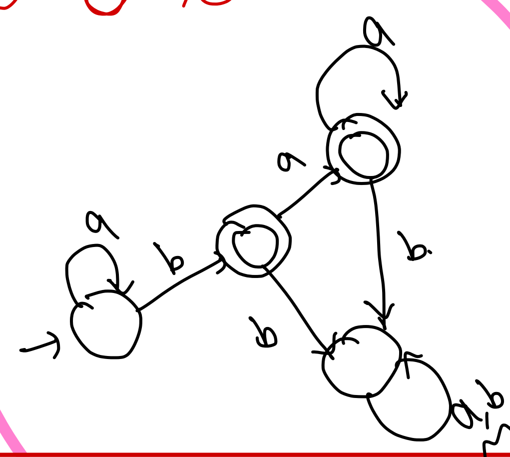
i) $\forall x \in L_1/L_2$, $x \in L(M')$.



ii) $\forall x \in L(M')$, $x \in L_1/L_2$.

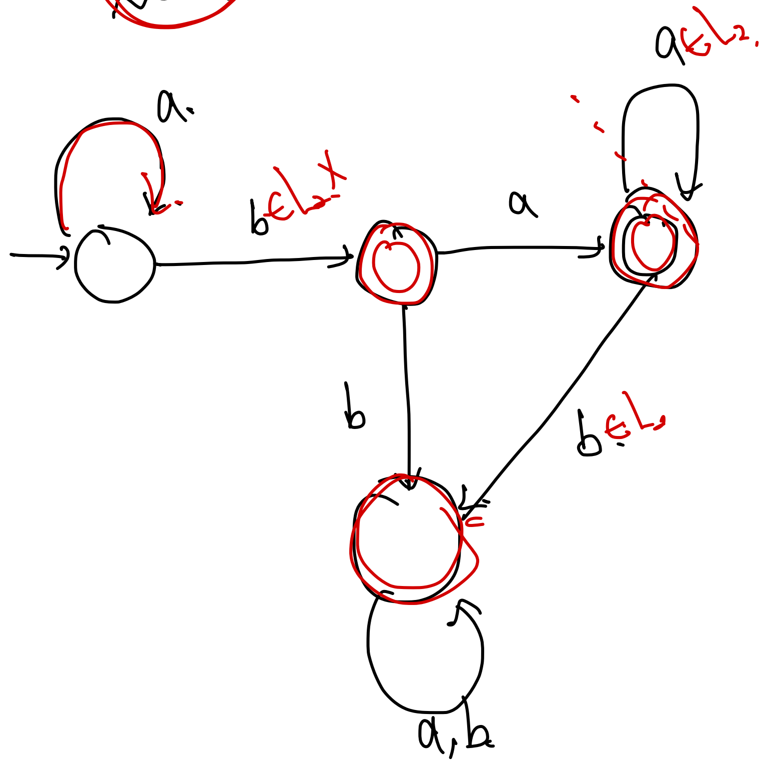
Ex4.5] $L_1 = L(a^*baa^*)$, $L_2=L(ab^*)$

$L_1/L_2=$



Find L_1/L_2 for $L_1 = L(a^*baa^*)$ $L_2 = L(ab^*)$

find dfa that accepts L_1.



Questions on Languages

w∈L. ? 답할 수 있는 Algorithm.

[Q1] Membership question

- Given a language L and a string w, can we determine whether or not w is an element of L?
- We need a membership algorithm

(w가 L에 속하는지 판단할 수 있나?)

[Q2] Finiteness of a language

- Is a language is empty, finite, or infinite?

[Q3] Equality of languages

- Are two languages L_1 and L_2 equal to each other?

For regular languages, we can answer to the questions

Answers for Regular Languages

regular Language → regular grammar
reg. expression
FA.

- L is a regular ⇔
  L has a FA, a reg. expressions, and a reg. grammar.

[THEOREM 4.5] Given a standard representation of any regular languages L on Σ and any $w \in \Sigma^*$, there exist a membership algorithm for w and L.

Tips for Proof) Use dfa

[THEOREM 4.6] There exist an algorithm for determining the emptiness and infiniteness.

Sketch of Proof)

1. Find a dfa accepting the L
2. If there exist a path from q_0 to q_f, then L is **not empty**
3. Find all bases of some cycle in the dfa.
4. If any of the bases are on the path form q_0 to q_f, then L is **infinite**.

xz → accept.
xyz → accept.
xy^2z → accept

infinite language.

Answers for Regular Languages

[THEOREM 4.7]

Given a standard representation of regular languages L_1 and L_2, there exist an algorithm to determine whether or not $L_1=L_2$.

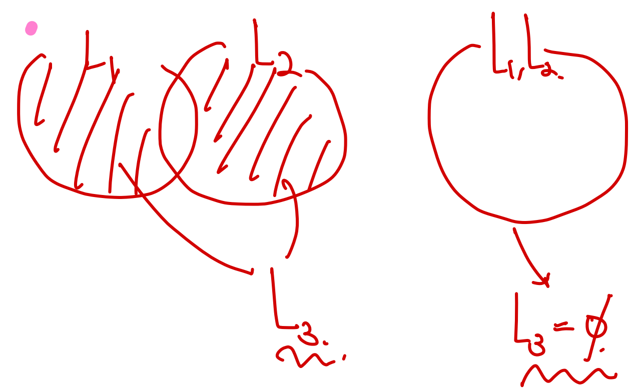
Sketch of Proof)

Let $L_3 = (L_1 \cap L_2^c) \cup (L_1^c \cap L_2)$ → L_3도 regular. → dfa도 존재

Note that L_3 is regular.

Find a dfa accepting L_3.

L_3 is empty $\Leftrightarrow$ $L_1=L_2$.



$L=\{a^n b^n \mid n \geq 0\}$

Regular가 아님을 증명하는 법

Using Pigeonhole Principle

- Pigeonhole principle
  - If we put n objects into m boxes (n>m),
    then at least one box must have more than one item.

Ex4.6] Show that $L=\{a^n b^n | n \geq 0\}$ not regular

Proof using pigeon hole principle)

- Assume that L is regular. ($\exists$ dfa $M=(Q,\Sigma,\delta,q_0,F)$ accepting L)
- Note that $|Q|<\infty$
- Take a string $w=a^n b^n$ ($n>|Q|$) and use pigeonhole principle

CORE

A Pumping Lemma(1/10)

regular language 이면 성립한다.

Theorem 4.8] Pumping Lemma for regular languages

– L : an infinite regular language.

L is finite ⟹ regular.

Then there exists some positive integers m such that

any $w \in L$ with $(|w| \geq m)$ can be decomposed as

$w = xyz$,

with $|xy| \leq m$ a and $(|y| \geq 1)$,

such that $w_i = xy^iz$ is also in L for all $i=0,1,2,\ldots$

Pumping Lemma for regular language – logical expression

If L is an infinite regular language, then

$\exists$ m >0 such that m이 양의 경우.

$\forall$ $w \in L$ ($|w| \geq m$) m보다 같은 L에 속하는 모든 w.

$\exists$ a decomposition $w = xyz$ ($|xy| \leq m$, $|y| \geq 1$) such that

$\forall$ $i=1,2,\ldots$ $w_i = xy_iz \in L$

충분히 크게 잡은 w ∈ L

⟹ xyz로 decomposition

앞부분에

즉 w는 substring y를 가지고 있어서

y는 pumping 된다.

어떤 특정값의 m

y를 반복해서 새롭게 만든 string도 L에 속한다

If L is Infinite regular language.

$\exists m > 0$ such that:

$\forall w \in L \ (|w| \ge m)$

$\exists$ a decomposition

$w = xyz \ (|xy| \le m, \ |y| \ge 1)$

such that $\forall i = 1, 2, \cdots$

$w_i = xy^i z \in L$

$L = \{a(ab)^n \mid n \ge 0\}$.

$a(ab)^*$ ((ab)가 pumping 된다.).

"$w \in L$ with $|w| \ge m$."

"$w = xyz$"

$|xy| \le m$. and $|y| \ge 1$.

such that $w_i = xy^i z$ is also in L.

for all $i = 0, 1, 2 \ldots$

A Pumping Lemma (2/10)

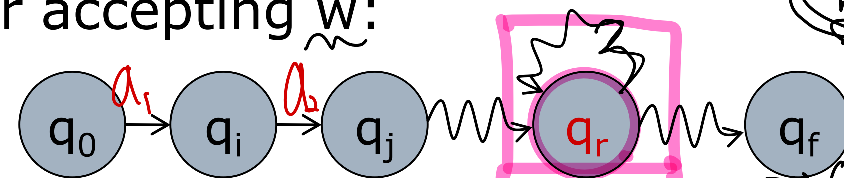
Proof)

L is regular ⇒ There is a dfa with $n+1$ states $q_0,q_1,q_2,\ldots q_n$.

L is infinite ⇒ There is a string $w \in L$, $|w| \geq m=n+1$.

The sequence of states for accepting w:

$q_0,q_i,q_j,\ldots,q_r,\ldots q_r,\ldots,q_f,$

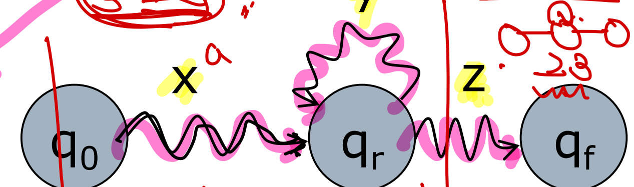


Note that there exist at least one repeated state.

⇒ there exist a decomposition of $w=xyz$

$\delta^*(q_0,x)=q_r$, $\delta^*(q_r,y)=q_r$, $\delta^*(q_r,z)=q_f$,

with $|xy| \leq n+1=m$ and $|y| \geq 1$.



This immediately follows that

$\delta^*(q_0,xz)=q_f$, $\delta^*(q_r,xy^2z)=q_f$, $\delta^*(q_r,xy^3z)=q_f$, and so on.

A Pumping Lemma (3/10)

L is reg → P.L is true

- Proving "L is not regular" using proof by contradiction
  - Assume L is regular and show that pumping lemma is false.

부정

P.L is false → L is not regular.

[Steps for Proving L is not regular]

pumping lemma 의 부정

Assume L is regular.
Show that the negation of pumping lemma is true → contradiction!
Thus, we have the conclusion: L is not regular

[Negation of pumping Lemma]

$\forall$ m >0 such that
  $\exists$ w$\in$L (|w| $\geq$ m) such that
    for all possible decompositions w=xyz (|xy| $\leq$ m, |y| $\geq$ 1)
      $\exists$ i such that $w_i = xy^iz \notin L$

A Pumping Lemma (4/10)

모든 m에 대해서 성립.
해야 해서.

[Steps for proving the negation of pumping lemma]

Suppose an arbitrary **m** (>0) is given.

① Choose a string **w**$\in$L ($|w| \geq m$)

② Consider all possible decompositions of w, $w=xyz$ ($|xy| \leq m$, $|y| \geq 1$)

책임을 지고 모든 decomposition
을 찾아야 한다.

③ Find i satisfying $w_i=xy^iz \notin L$

Ex4.7] Show that $L=\{a^nb^n|n\geq 0\}$ is not regular

- Assume L is regular
- For any m, we can choose $w=a^mb^m$. → arbitrary m (>0).
- Consider all possible decompositions of $w=xyz$ ($|xy|\leq m, |y|\geq 1$)

  $w\in L$ $|w|\geq m$

  $w=a......ab......b$ → $y=a^k$ ($1\leq k \leq m$)

  → y가 취할 수 있는 형태

- For any y, set i=0 then $w_0=xz=a^{m-k}b^m \notin L$ ⟹ ~pumping lemma is true.
- This implies that pumping lemma is false → contradiction!
- Thus, L is not regular.

$\overbrace{a......a}^{m}\overbrace{b.......b}^{m}$ →

y는 a로만 이루어짐.

not regular

A Pumping Lemma (5/10)

Ex4.8] Show that $L=\{ww^R | w\in\{a,b\}^*\}$is not regular

- Assume L is regular
- For any m, we choose $w=a^m b^m$
  $ww^R = a^m b^m b^m a^m \in L.$
- Consider all possible decomposition of $ww^R=a^m b^{2m} a^m=xyz$ $(|xy|\leq m, |y|\geq 1)$.

  Since $ww^R=\underbrace{a....a}_{m}\overbrace{b......b}^{2m}\underbrace{a....a}_{m}$, we have $y=a^k$ $(1\leq k \leq m)$
- For any $y=a^k$, set i=0, then $w_0=xz=\boxed{a^{m-k}b^mb^ma^m}$ L
  pumping lemma is false
  → L is not regular.
- Thus, L is not regular.

A Pumping Lemma (6/10)

Ex4.9] $L=\{w \in \{a,b\}^* \mid n_a(w) < n_b(w)\}$ is not regular

Assume L is Regular.

$\forall m > 0$, we chose. $w = a^m b^{m+1} \in L$. $(|w| \ge m)$.

Consider all possible decomp of $w = xyz$. $(|xy| \le m,\ |y| \ge 1)$.

y has the form. $y = a^k$ $(1 \le k \le m)$.

~~set $i=0$~~ $\cancel{w_0 = a^{m-k} b^{m+1} \in L.}$

i를 잘못 잡은 경우.

$i=2$. $w_0 = a^{m+k} b^{m+1} \notin L$. pumping Lema is false so L is not regular.

A Pumping Lemma (7/10)

Ex4.10] $L=\{(ab)^n a^k \mid n>k,\ k\geq 0\}$ is not regular

Assume L is regular.

$\forall m>0$, we choose $w=(ab)^{m+1}a^m \in L$. $(|w|\geq m)$

Consider $\forall$ possible decomp of $w=xyz$. $(|xy|\leq m,\ |y|\geq 1)$

$w = \underbrace{ab\,ab \cdots abab}_{2(m+1)}\,\overbrace{a \cdots\cdots a}^{m}$

$(ab)^{m+1-\frac{k}{2}}a^m \notin L$ $(\because m+1-\frac{k}{2}\leq m)$

$\Downarrow$

(xz ∉ L
애초에 a로 끝나든가 (a 시작).
(ab) 묶음이 만들어지지 않는다.)

각각의 적절한 i를 찾아야 한다.

① $y=(ab)^{\frac{k}{2}}$ ($1\leq k\leq m$, k는 짝수)

② $y=(ba)^{\frac{k}{2}}$ ($1\leq k\leq m$, k는 짝수).

③ $y=(ab)^{\frac{k-1}{2}}a$ ($1\leq k\leq m$, k는 홀수).

④ $y=(ba)^{\frac{k-1}{2}}b$ ($1\leq k\leq m$, k는 홀수).

", 마찬가지 생략}

$i=0$: $xz=(ab)^{m+1-\frac{k}{2}}a^m \notin L$. $(m+1-\frac{k}{2}\leq m)$

$i=0$: $xz=(ab)^{m+1-\frac{k}{2}}a^k \notin L$ $(m+1-\frac{k}{2}\leq m)$.

A Pumping Lemma (8/10)

Ex4.11] L={$a^{n!}$|n≥0} is not regular

Assume. L is regular. $(m \geq 3)$.

$\forall m > 0$, we choose $w = a^{m!} \in L$ $(|w| \geq m.)$

Consider $\forall$ possible decomposition of $w = xyz$ $(|xy| \leq m, |y| \geq 1.)$

$w = \overbrace{a \cdots\cdots a}^{m!}$

we have $y = a^k$ $(1 \leq k \leq m)$

set $i = 0$. $w = (a^{m!-k}) \notin L$. pumping Lemma is false.

$(m!-k \geq m!-m > (m-1)!)$

A Pumping Lemma (9/10)

Ex4.12] $L=\{a^n b^k c^{n+k} \mid n \geq 0,\ k \geq 0\}$ is not regular

- Use homomorphism

(이미 Regular 가 아닌 language 를 활용 한다)

Assume L is regular

$h(L)$ is regular language too.

Define. $\begin{pmatrix} h(a) = a \\ h(b) = a \\ h(c) = b \end{pmatrix}$

$h(L) = \{a^n a^k b^{n+k}\} = \{a^{n+k} b^{n+k}\}$ = is regular.

우리는 이미 $\{a^n b^n \mid n \geq 0\}$ 은 regular 가 아님을 알고 있음.

So 모순 발생 $\Rightarrow$ L is not regular.

- Use pumping lemma

A Pumping Lemma (10/10)

Ex4.13] Show that $L=\{a^n b^l | n \neq l\}$ is not regular

- Use known fact ($L=\{a^n b^n | n \geq 0\}$ is not regular)

L is regular. $L_1 = \{a^n b^n\} = \underbrace{\bar{L} \cap \boxed{L(a^* b^*)}}$ → reg expression

(Regular Language.) $L_1 = \{a^n b^n\}$ is regular. → contradiction!

⟹ L is not regular.

- Use pumping lemma