

# Lecture 19 – Closure Properties of Context-Free Languages

COSE215: Theory of Computation

Jihyeok Park



2026 Spring

- A **context-free language (CFL)** is defined in three different ways:
  - A **context free grammar (CFG)**
  - A **pushdown automaton (PDA)** with **final states**
  - A **pushdown automaton (PDA)** with **empty stacks**
- We have learned that the class of **regular languages** is **closed** under various operations. (**Closure Properties**)
- For which operations is the class of **CFLs closed**?

## 1. Closure Properties of Context-Free Languages

Union

Concatenation

Kleene Star

Reversal

Homomorphism

Inverse Homomorphism

## 2. Non-Closure Properties of Context-Free Languages

Intersection

Complement and Difference

## 3. Closure Properties of CFLs with Regular Languages

Intersection with Regular Languages

Difference with Regular Languages

## 1. Closure Properties of Context-Free Languages

Union

Concatenation

Kleene Star

Reversal

Homomorphism

Inverse Homomorphism

## 2. Non-Closure Properties of Context-Free Languages

Intersection

Complement and Difference

## 3. Closure Properties of CFLs with Regular Languages

Intersection with Regular Languages

Difference with Regular Languages

## Definition (Closure Properties)

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

The class of CFLs is closed under the following operations:

- Union
- Concatenation
- Kleene Star
- Reverse
- Homomorphism
- Inverse Homomorphism

## Theorem (Closure under Union)

*If  $L_1$  and  $L_2$  are context-free languages, then so is  $L_1 \cup L_2$ .*

## Theorem (Closure under Union)

*If  $L_1$  and  $L_2$  are context-free languages, then so is  $L_1 \cup L_2$ .*

**Proof)** For given two CFLs  $L_1$  and  $L_2$ , we can always construct two CFGs:

$$G_1 = (V_1, \Sigma, S_1, R_1)$$

$$G_2 = (V_2, \Sigma, S_2, R_2)$$

such that  $L_1 = L(G_1)$  and  $L_2 = L(G_2)$ .

Note that the variables of  $G_1$  and  $G_2$  should be disjoint. (i.e.,  $V_1 \cap V_2 = \emptyset$ )

Then,  $L_1 \cup L_2$  is accepted by the CFG  $G = (V, \Sigma, S, R)$  where:

- $V = V_1 \cup V_2 \cup \{S\}$
- $S$  is a new start variable (i.e.,  $S \notin V_1 \cup V_2$ )
- $R = R_1 \cup R_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}$

□

## Closure under Union – Example

For example, consider the following two CFLs:

$$L_1 = \{ab^n \mid n \geq 0\} \quad L_2 = \{ac^n \mid n \geq 0\}$$

Then,  $L_1$  is accepted by:

$$S_1 \rightarrow aX \quad X \rightarrow bX \mid \epsilon$$

and  $L_2$  is accepted by:

$$S_2 \rightarrow aX \quad X \rightarrow cX \mid \epsilon$$

But, the same variable  $X$  is used in both grammars.

So, we need to rename it to different variables, such as  $B$  and  $C$ .

## Closure under Union – Example

For example, consider the following two CFLs:

$$L_1 = \{ab^n \mid n \geq 0\} \quad L_2 = \{ac^n \mid n \geq 0\}$$

Then,  $L_1$  is accepted by:

$$S_1 \rightarrow aB \quad B \rightarrow bB \mid \epsilon$$

and  $L_2$  is accepted by:

$$S_2 \rightarrow aC \quad C \rightarrow cC \mid \epsilon$$

## Closure under Union – Example

For example, consider the following two CFLs:

$$L_1 = \{ab^n \mid n \geq 0\} \quad L_2 = \{ac^n \mid n \geq 0\}$$

Then,  $L_1$  is accepted by:

$$S_1 \rightarrow aB \quad B \rightarrow bB \mid \epsilon$$

and  $L_2$  is accepted by:

$$S_2 \rightarrow aC \quad C \rightarrow cC \mid \epsilon$$

Then,  $L_1 \cup L_2$  is accepted by the following CFG:

$$\begin{aligned} S &\rightarrow S_1 \mid S_2 \\ S_1 &\rightarrow aB \quad B \rightarrow bB \mid \epsilon \\ S_2 &\rightarrow aC \quad C \rightarrow cC \mid \epsilon \end{aligned}$$

## Theorem (Closure under Concatenation)

*If  $L_1$  and  $L_2$  are context-free languages, then so is  $L_1 \cdot L_2$ .*

## Theorem (Closure under Concatenation)

If  $L_1$  and  $L_2$  are context-free languages, then so is  $L_1 \cdot L_2$ .

**Proof)** For given two CFLs  $L_1$  and  $L_2$ , we can always construct two CFGs:

$$G_1 = (V_1, \Sigma, S_1, R_1)$$

$$G_2 = (V_2, \Sigma, S_2, R_2)$$

such that  $L_1 = L(G_1)$  and  $L_2 = L(G_2)$ .

Note that the variables of  $G_1$  and  $G_2$  should be disjoint. (i.e.,  $V_1 \cap V_2 = \emptyset$ )

Then,  $L_1 \cdot L_2$  is accepted by the CFG  $G = (V, \Sigma, S, R)$  where:

- $V = V_1 \cup V_2 \cup \{S\}$
- $S$  is a new start variable (i.e.,  $S \notin V_1 \cup V_2$ )
- $R = R_1 \cup R_2 \cup \{S \rightarrow S_1 S_2\}$



For example, consider the following two CFLs:

$$L_1 = \{ab^n \mid n \geq 0\} \quad L_2 = \{ac^n \mid n \geq 0\}$$

Then,  $L_1$  is accepted by:

$$S_1 \rightarrow aB \quad B \rightarrow bB \mid \epsilon$$

and  $L_2$  is accepted by:

$$S_2 \rightarrow aC \quad C \rightarrow cC \mid \epsilon$$

Then,  $L_1 \cdot L_2$  is accepted by the following CFG:

$$\begin{aligned} S &\rightarrow S_1 S_2 \\ S_1 &\rightarrow aB \quad B \rightarrow bB \mid \epsilon \\ S_2 &\rightarrow aC \quad C \rightarrow cC \mid \epsilon \end{aligned}$$

## Theorem (Closure under Kleene Star)

*If  $L$  is a context-free language, then so is  $L^*$ .*

## Theorem (Closure under Kleene Star)

*If  $L$  is a context-free language, then so is  $L^*$ .*

**Proof)** For a given CFL  $L$ , we can always construct a CFG:

$$G = (V, \Sigma, S, R)$$

such that  $L = L(G)$ .

Then,  $L^*$  is accepted by the CFG  $G' = (V', \Sigma, S', R')$  where:

- $V' = V \cup \{S'\}$
- $S'$  is a new start variable (i.e.,  $S' \notin V$ )
- $R' = R \cup \{S' \rightarrow \epsilon, S' \rightarrow SS'\}$

□

# Closure under Kleene Star – Example

For example, consider the following CFL:

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

Then,  $L$  is accepted by:

$$S \rightarrow \epsilon \mid aSb$$

Then,  $L^*$  is accepted by the following CFG:

$$\begin{aligned} S' &\rightarrow \epsilon \mid SS' \\ S &\rightarrow \epsilon \mid aSb \end{aligned}$$

## Theorem (Closure under Reverse)

*If  $L$  is a context-free language, then so is  $L^R$ .*

## Theorem (Closure under Reverse)

*If  $L$  is a context-free language, then so is  $L^R$ .*

**Proof)** For a given CFL  $L$ , we can always construct a CFG:

$$G = (V, \Sigma, S, R)$$

such that  $L = L(G)$ .

Then,  $L^R$  is accepted by the CFG  $G' = (V, \Sigma, S, R')$  where:

- $R' = \{X \rightarrow \alpha^R \mid X \rightarrow \alpha \in R\}$



## Closure under Reverse – Example

For example, consider the following CFL:

$$L = \{(ab)^n c^n d^m \mid n, m \geq 0\}$$

Then,  $L$  is accepted by:

$$\begin{aligned} S &\rightarrow X \mid Sd \\ X &\rightarrow \epsilon \mid abXc \end{aligned}$$

Then,  $L^R$  is accepted by the following CFG:

$$\begin{aligned} S &\rightarrow X \mid dS \\ X &\rightarrow \epsilon \mid cXba \end{aligned}$$

Let's recall the definition of a **homomorphism**.

## Definition (Homomorphism)

Suppose  $\Sigma_0$  and  $\Sigma_1$  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  $L \subseteq \Sigma_0^*$ ,

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

Let's recall the definition of a **homomorphism**.

## Definition (Homomorphism)

Suppose  $\Sigma_0$  and  $\Sigma_1$  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 context-free language, then so is  $h(L)$ .*

## Theorem (Closure under Homomorphism)

*If  $h$  is a homomorphism and  $L$  is a context-free language, then so is  $h(L)$ .*

**Proof)** For a given CFL  $L$ , we can always construct a CFG:

$$G = (V, \Sigma_0, S, R)$$

such that  $L = L(G)$ .

Then, for a given homomorphism  $h : \Sigma_0 \rightarrow \Sigma_1^*$ ,  $h(L)$  is accepted by the CFG  $G' = (V', \Sigma_1, S, R')$  where:

- $V' = V \cup \{X_a \mid a \in \Sigma_0\}$
- $R' = \{Y \rightarrow Y'_1 \cdots Y'_n \mid Y \rightarrow Y_1 \cdots Y_n \in R\} \cup \{X_a \rightarrow h(a) \mid a \in \Sigma_0\}$

$$\text{where } \forall 1 \leq i \leq n. Y'_i = \begin{cases} Y_i & \text{if } Y_i \in V \\ X_a & \text{if } Y_i = a \in \Sigma_0 \end{cases}$$

□

For example, consider the following CFL:

$$L = \{ww^R \mid w \in \{0, 1\}^*\}$$

Then,  $L$  is accepted by:

$$S \rightarrow \epsilon \mid 0S0 \mid 1S1$$

If a homomorphism  $h : \{0, 1\} \rightarrow \{a, b\}^*$  is defined as follows:

$$h(0) = ab \quad h(1) = a$$

Then,  $h(L)$  is accepted by the following CFG:

$$\begin{aligned} S &\rightarrow \epsilon \mid X_0SX_0 \mid X_1SX_1 \\ X_0 &\rightarrow ab \\ X_1 &\rightarrow a \end{aligned}$$

Let's recall the definition of an **inverse homomorphism**.

## Definition (Inverse Homomorphism)

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

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

Let's recall the definition of an **inverse homomorphism**.

## Definition (Inverse Homomorphism)

Suppose  $\Sigma_0$  and  $\Sigma_1$  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  $h(0) = aa$ ,  $h(1) = b$ . Consider the following language  $L \subseteq \Sigma_1^*$ :

$$L = \{a^n b^n \mid n \geq 1\}$$

Then,  $011 \in h^{-1}(L)$  because  $h(011) = aabb \in L$ .

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

## Theorem (Closure under Inverse Homomorphism)

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

## Theorem (Closure under Inverse Homomorphism)

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

**Proof)** Consider a PDA  $P = (Q, \Sigma_1, \Gamma, \delta, q_0, Z, F)$  for  $L$  by final states.

The key idea is to construct a new PDA  $P'$  that simulates  $P$  with pairs of 1) states and 2) remaining symbols of  $\Sigma_1$  as new states.

## Theorem (Closure under Inverse Homomorphism)

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

**Proof)** Consider a PDA  $P = (Q, \Sigma_1, \Gamma, \delta, q_0, Z, F)$  for  $L$  by final states.

The key idea is to construct a new PDA  $P'$  that simulates  $P$  with pairs of 1) states and 2) remaining symbols of  $\Sigma_1$  as new states.

Then, a PDA  $P' = (Q \times h(\Sigma_0)_{\succeq}, \Sigma_0, \Gamma, \delta', (q_0, \epsilon), Z, F \times \{\epsilon\})$  accepts  $h^{-1}(L)$  by final states where:

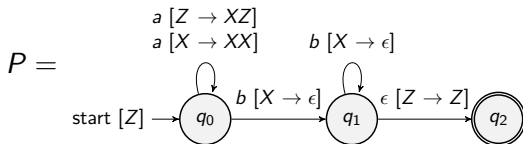
- $A_{\succeq} = \{x \in \Sigma_1^* \mid x \text{ is a suffix of } w \in A\}$  for any  $A \subseteq \Sigma_1^*$
- For all  $a \in \Sigma_0$ ,  $q \in Q$ , and  $X \in \Sigma_1$ ,

$$\delta'((q, \epsilon), a, X) = \{((q, h(a)), X)\}$$

- For all  $b \in \Sigma_1 \cup \{\epsilon\}$ ,  $bx \in h(\Sigma_0)_{\succeq}$ ,  $q \in Q$ , and  $X \in \Sigma_1$ ,

$$\delta'((q, bx), \epsilon, X) = \{((p, x), \gamma) \mid (p, \gamma) \in \delta(q, b, X)\}$$

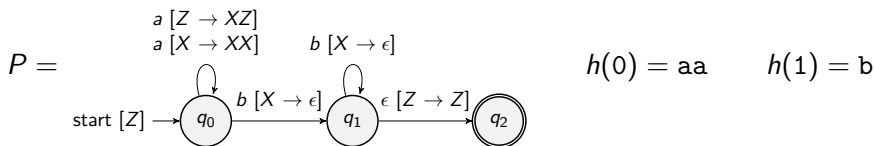
For example, consider the following PDA:



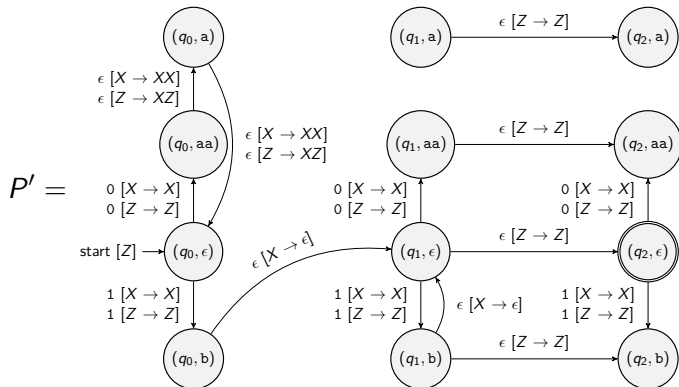
that accepts  $L = \{a^n b^n \mid n \geq 1\}$  by final states.

If a homomorphism  $h : \{0, 1\}^* \rightarrow \{a, b\}^*$  is defined as follows:

$$h(0) = aa \qquad h(1) = b$$



Then, the following PDA accepts  $h^{-1}(L)$  by final states:



## 1. Closure Properties of Context-Free Languages

Union

Concatenation

Kleene Star

Reversal

Homomorphism

Inverse Homomorphism

## 2. Non-Closure Properties of Context-Free Languages

Intersection

Complement and Difference

## 3. Closure Properties of CFLs with Regular Languages

Intersection with Regular Languages

Difference with Regular Languages

## Definition (Closure Properties)

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

The class of CFLs is **NOT** closed under the following operations:

- Intersection
- Complement
- Difference

## Definition (Closure Properties)

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

The class of CFLs is **NOT** closed under the following operations:

- Intersection
- Complement
- Difference

We will prove it by using the fact that the following language is not a CFL:

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

## Definition (Closure Properties)

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

The class of CFLs is **NOT** closed under the following operations:

- Intersection
- Complement
- Difference

We will prove it by using the fact that the following language is not a CFL:

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

We will learn how to prove that  $L$  is not a CFL in the next lecture (Pumping Lemma for CFLs).

## Theorem (Non-Closure under Intersection)

*The class of CFLs is **NOT** closed under intersection.*

## Theorem (Non-Closure under Intersection)

The class of CFLs is **NOT** closed under intersection.

**Proof)** Consider the following two languages:

$$L_1 = \{a^n b^n c^m \mid n, m \geq 0\} \quad L_2 = \{a^m b^n c^n \mid n, m \geq 0\}$$

Then,  $L_1$  is accepted by:

$$S_1 \rightarrow X \mid S_1 c \quad X \rightarrow \epsilon \mid aXb$$

and  $L_2$  is accepted by:

$$S_2 \rightarrow Y \mid aS_2 \quad Y \rightarrow \epsilon \mid bYc$$

Thus, they are both CFLs. However, their intersection is not a CFL:

$$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$$



## Theorem (Non-Closure under Complement)

*The class of CFLs is **NOT** closed under complement.*

**Theorem (Non-Closure under Complement)**

*The class of CFLs is **NOT** closed under complement.*

**Proof)** Assume that the class of CFLs is closed under complement. Then, for any two CFLs  $L_1$  and  $L_2$ ,  $L_1 \cap L_2$  is also a CFL:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

However, we have already proved that the class of CFLs is not closed under intersection. Thus, the class of CFLs is not closed under complement.  $\square$

**Theorem (Non-Closure under Complement)**

*The class of CFLs is **NOT** closed under complement.*

**Proof)** Assume that the class of CFLs is closed under complement. Then, for any two CFLs  $L_1$  and  $L_2$ ,  $L_1 \cap L_2$  is also a CFL:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

However, we have already proved that the class of CFLs is not closed under intersection. Thus, the class of CFLs is not closed under complement.  $\square$

**Theorem (Non-Closure under Difference)**

*The class of CFLs is **NOT** closed under difference.*

### Theorem (Non-Closure under Complement)

*The class of CFLs is **NOT** closed under complement.*

**Proof)** Assume that the class of CFLs is closed under complement. Then, for any two CFLs  $L_1$  and  $L_2$ ,  $L_1 \cap L_2$  is also a CFL:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

However, we have already proved that the class of CFLs is not closed under intersection. Thus, the class of CFLs is not closed under complement.  $\square$

### Theorem (Non-Closure under Difference)

*The class of CFLs is **NOT** closed under difference.*

**Proof)** Similarly, we can prove it using the following fact:

$$L_1 \cap L_2 = L_1 \setminus (L_1 \setminus L_2)$$

 $\square$

## 1. Closure Properties of Context-Free Languages

Union

Concatenation

Kleene Star

Reversal

Homomorphism

Inverse Homomorphism

## 2. Non-Closure Properties of Context-Free Languages

Intersection

Complement and Difference

## 3. Closure Properties of CFLs with Regular Languages

Intersection with Regular Languages

Difference with Regular Languages

## Definition (Closure Properties)

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

The class of CFLs is closed under the following operations with RLs:

- Intersection
- Difference

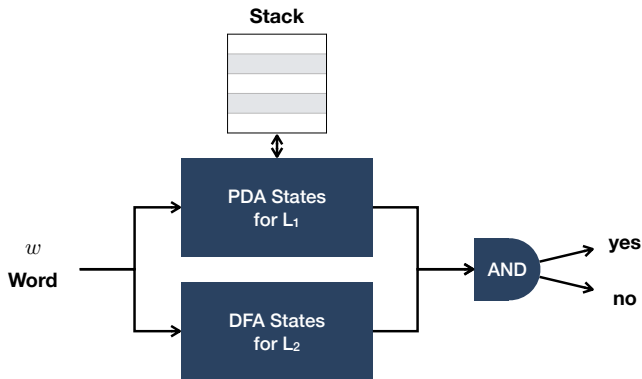
## Theorem (Closure under Intersection with RLs)

*If  $L_1$  is a CFL and  $L_2$  is a RL, then  $L_1 \cap L_2$  is a CFL.*

## Theorem (Closure under Intersection with RLs)

If  $L_1$  is a CFL and  $L_2$  is a RL, then  $L_1 \cap L_2$  is a CFL.

There exists a PDA  $P$  that accepts  $L_1$  by final states and a DFA  $D$  that accepts  $L_2$ . We will construct a PDA  $P'$  that accepts  $L_1 \cap L_2$  as follows:



### Theorem (Closure under Intersection with RLs)

If  $L_1$  is a CFL and  $L_2$  is a RL, then  $L_1 \cap L_2$  is a CFL.

**Proof)** Consider a PDA  $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z, F_P)$  and a DFA  $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$  such that:

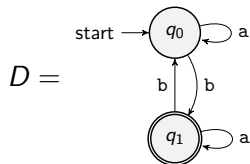
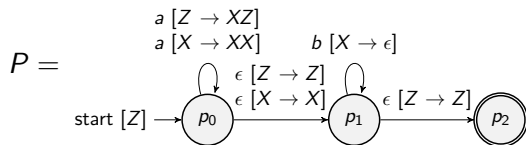
$$L_F(P) = L_1 \quad L(D) = L_2$$

Then,  $L_1 \cap L_2$  is accepted by the PDA  $P' = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$  by final states, where:

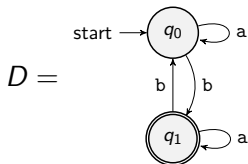
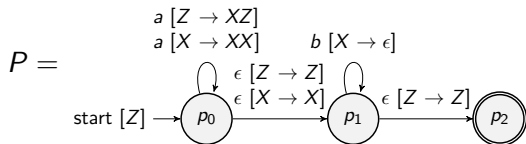
- $Q = Q_P \times Q_D$
- $\delta((p, q), \epsilon, X) = \{((p', q), \alpha) \mid (p', \alpha) \in \delta_P(p, \epsilon, X)\}$
- $\delta((p, q), a, X) = \{((p', q'), \alpha) \mid (p', \alpha) \in \delta_P(p, a, X) \wedge q' = \delta_D(q, a)\}$
- $q_0 = (q_P, q_D)$
- $F = F_P \times F_D$



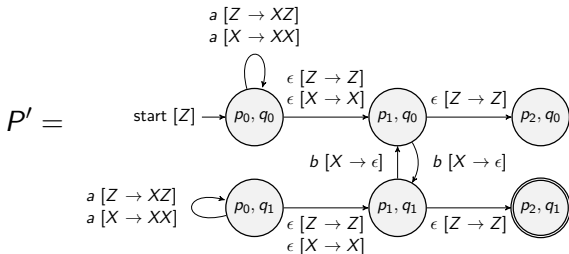
For example, consider the following PDA  $P$  and DFA  $D$ :



For example, consider the following PDA  $P$  and DFA  $D$ :



Then, a PDA  $P'$  that accepts  $L_F(P) \cap L(D)$  by the final states can be constructed as follows:



## Theorem (Closure under Difference with RLs)

*If  $L_1$  is a CFL and  $L_2$  is a RL, then  $L_1 \setminus L_2$  is a CFL.*

**Theorem (Closure under Difference with RLs)**

*If  $L_1$  is a CFL and  $L_2$  is a RL, then  $L_1 \setminus L_2$  is a CFL.*

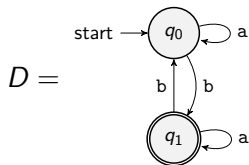
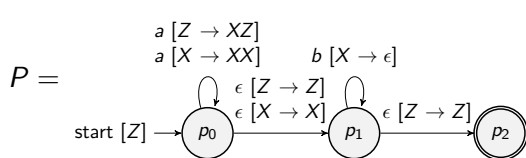
**Proof)** We know the following fact:

$$L_1 \setminus L_2 = L_1 \cap \overline{L_2}$$

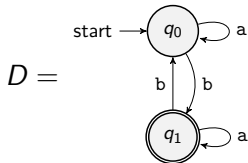
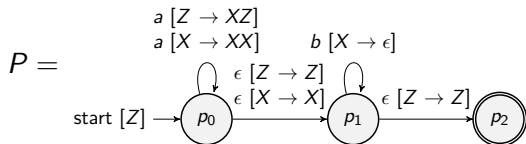
Since the class of RLs is closed under complement,  $\overline{L_2}$  is a RL. In addition, we know that the class of CFLs is closed under intersection with RLs.

Thus,  $L_1 \setminus L_2$  is a CFL. □

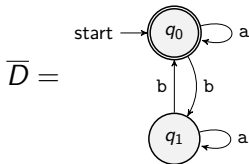
For example, consider the following PDA  $P$  and DFA  $D$ :



For example, consider the following PDA  $P$  and DFA  $D$ :

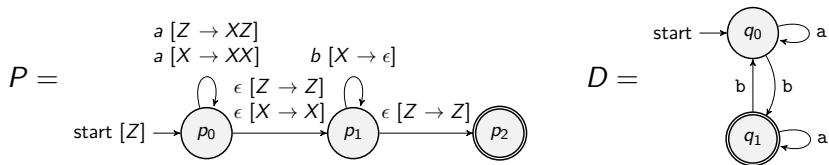


Then, a DFA  $\overline{D}$  that accepts  $\overline{L(D)}$  and a PDA  $P'$  that accepts  $L_F(P) \setminus L(D) = L_F(P) \cap \overline{L(D)}$  can be constructed as follows:

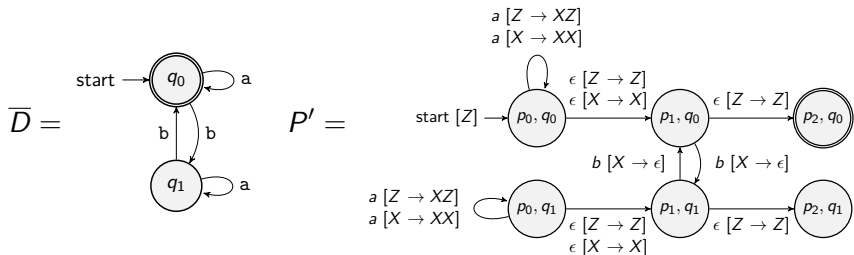


# Closure under Difference with RLs – Example

For example, consider the following PDA  $P$  and DFA  $D$ :



Then, a DFA  $\overline{D}$  that accepts  $\overline{L(D)}$  and a PDA  $P'$  that accepts  $L_F(P) \setminus L(D) = L_F(P) \cap \overline{L(D)}$  can be constructed as follows:



## 1. Closure Properties of Context-Free Languages

Union

Concatenation

Kleene Star

Reversal

Homomorphism

Inverse Homomorphism

## 2. Non-Closure Properties of Context-Free Languages

Intersection

Complement and Difference

## 3. Closure Properties of CFLs with Regular Languages

Intersection with Regular Languages

Difference with Regular Languages

- Please see this document for the exercise.

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

- Please implement the following functions in `Implementation.scala`.
  - `unionCFG` for the **union** of two CFGs.
  - `concatCFG` for the **concatenation** of two CFGs.
  - `starCFG` for the **Kleene star** of a CFG.
  - `reverseCFG` for the **reversal** of a CFG.
  - `homCFG` for the **homomorphism** of a CFG.
  - `ihomPDA` for the **inverse homomorphism** of a PDA.
  - `intersectPDA` for the **intersection** of a PDA with a DFA.
- It is just an exercise, and you **don't need to submit** anything.

- The Pumping Lemma for Context-Free Languages

Jihyeok Park

[jihyeok\\_park@korea.ac.kr](mailto:jihyeok_park@korea.ac.kr)

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