Lecture 11 – Context-Free Grammars (CFGs) and Languages (CFLs)

COSE215: Theory of Computation

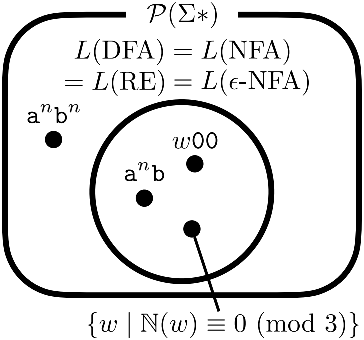
Jihyeok Park

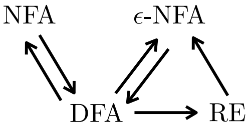
PLRG

2026 Spring

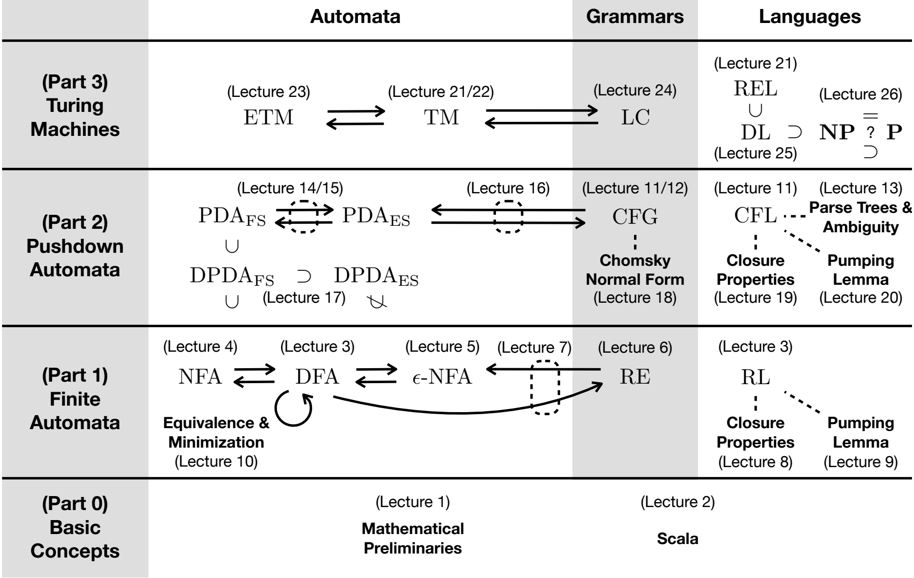
Recall

- Regular Languages
  - Finite Automata - DFA, NFA, ϵ-NFA
  - Regular Expressions





Recall



Automata
Grammars
Languages
(Part 3) Turing Machines
(Lecture 23) ETM
(Lecture 21/22) TM
(Lecture 24) LC
(Lecture 21) REL
∪
DL ⊃
(Lecture 25)
(Lecture 26) NP =? P
⊃
(Part 2) Pushdown Automata
(Lecture 14/15)
PDA_{FS}
PDA_{ES}
∪
$DPDA_{FS}$ ⊃ $DPDA_{ES}$
∪ (Lecture 17) ⊍
(Lecture 16)
(Lecture 11/12) CFG
Chomsky Normal Form (Lecture 18)
(Lecture 11) CFL
(Lecture 13) Parse Trees & Ambiguity
Closure Properties (Lecture 19)
Pumping Lemma (Lecture 20)
(Part 1) Finite Automata
(Lecture 4) NFA
(Lecture 3) DFA
(Lecture 5) ϵ-NFA
(Lecture 7)
(Lecture 6) RE
Equivalence & Minimization (Lecture 10)
(Lecture 3) RL
Closure Properties (Lecture 8)
Pumping Lemma (Lecture 9)
(Part 0) Basic Concepts
(Lecture 1) Mathematical Preliminaries
(Lecture 2) Scala

Context-Free Grammars (CFGs)

- Consider the following language:

$$L = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

For example, the following words are in (or not in) L:

$$\begin{array}{rcl} L & \ni & \epsilon,\ (),\ (()),\ ()(),\ (()()),\ (()(())),\ ((())),\ \cdots \\ L & \not\ni & (,\ ),\ )(,\ ((),\ ()),\ (())),\ (()((),\ \cdots \end{array}$$

Context-Free Grammars (CFGs)

- Consider the following language:

$$L = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

For example, the following words are in (or not in) L:

$$\begin{array}{lcl} L & \ni & \epsilon,\ (),\ (()),\ ()(),\ (()()),\ (()(())),\ ((())),\ \cdots \\ L & \not\ni & (,\ ),\ )(,\ ((),\ ()),\ (())),\ (()((),\ \cdots \end{array}$$

- Is this language regular?

Context-Free Grammars (CFGs)

- Consider the following language:

$$L = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

For example, the following words are in (or not in) L:

$$L \ni \epsilon, (), (()), ()(), (()()), (()(())), ((())), \cdots$$
$$L \not\ni (,),)(, ((), ()), (())), ()((), \cdots$$

- Is this language regular? **No**, we can prove that this language is **not regular** using the **Pumping Lemma** (Do it yourself!).

Context-Free Grammars (CFGs)

- Consider the following language:

$$L = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

  For example, the following words are in (or not in) L:

$$\begin{array}{lcl} L & \ni & \epsilon,\ (),\ (()),\ ()(),\ (()()),\ (()(())),\ ((())),\ \cdots \\ L & \not\ni & (,\ ),\ )(,\ ((),\ ()),\ (())),\ (()((),\ \cdots \end{array}$$

- Is this language regular? **No**, we can prove that this language is **not regular** using the **Pumping Lemma** (Do it yourself!).

- Is there a way to describe this language?

Context-Free Grammars (CFGs)

- Consider the following language:

$$L = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

For example, the following words are in (or not in) L:

$$L \ni \epsilon, (), (()), ()(), (()()), (()(())), ((())), \cdots$$
$$L \not\ni (,),)(, ((), ()), (())), (()((), \cdots$$

- Is this language regular? **No**, we can prove that this language is **not regular** using the **Pumping Lemma** (Do it yourself!).

- Is there a way to describe this language?

- Yes, let's learn **Context-Free Grammars (CFGs)**!

Contents

Basic Idea

$$L = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

How to **inductively generate** (or produce) words in the language L?

Basic Idea

$$L = \{w \in \{\texttt{(}, \texttt{)}\}^* \mid w \text{ is balanced}\}$$

How to **inductively generate** (or produce) words in the language L?

- **Base Case:** $\epsilon \in L$
- **Inductive Case:** There are two inductive rules:
  - If $w \in L$, then $\texttt{(}w\texttt{)} \in L$
  - If $w_1, w_2 \in L$, then $w_1 w_2 \in L$

Basic Idea

$$L = \{w \in \{\texttt{(}, \texttt{)}\}^* \mid w \text{ is balanced}\}$$

How to **inductively generate** (or produce) words in the language L?

- **Base Case:** $\epsilon \in L$
- **Inductive Case:** There are two inductive rules:
  - If $w \in L$, then $\texttt{(}w\texttt{)} \in L$
  - If $w_1, w_2 \in L$, then $w_1 w_2 \in L$

ϵ (()())   (())()   ()(())() $\cdots$

Basic Idea

PLRG

$$L = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

How to **inductively generate** (or produce) words in the language L?

- **Base Case:** $\epsilon \in L$
- **Inductive Case:** There are two inductive rules:
  - If $w \in L$, then $(w) \in L$
  - If $w_1, w_2 \in L$, then $w_1 w_2 \in L$

ϵ $\quad$ $(()())$ $\quad$ $(())()$ $\quad$ $()(())()$ $\quad$ $\cdots$

Context-Free Grammars (CFGs) provide a way to describe languages with such **inductive rules** to generate words in the language.

Context-Free Grammars (CFGs)

Definition (Context-Free Grammar (CFG))

A **context-free grammar** is a 4-tuple:

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

where

- V: a finite set of **variables** (nonterminals)
- Σ: a finite set of **symbols** (terminals)
- $S \in V$: the **start variable**
- $R \subseteq V \times (V \cup \Sigma)^*$: a set of **production rules**.

Context-Free Grammars (CFGs)

Definition (Context-Free Grammar (CFG))

A **context-free grammar** is a 4-tuple:

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

where

- V: a finite set of **variables** (nonterminals)
- Σ: a finite set of **symbols** (terminals)
- $S \in V$: the **start variable**
- $R \subseteq V \times (V \cup \Sigma)^*$: a set of **production rules**.

$$G = (\{S, A, B\}, \{\texttt{(}, \texttt{)}\}, S, R)$$

where R is defined as:

$$\begin{array}{lll} S \rightarrow \epsilon & S \rightarrow A & S \rightarrow B \\ A \rightarrow \texttt{(}S\texttt{)} & B \rightarrow SS & \end{array}$$

Context-Free Grammars (CFGs)

$$G = (\{S, A, B\}, \{(,)\}, S, R)$$

where R is defined as:

$$\begin{array}{lll} S \rightarrow \epsilon & S \rightarrow A & S \rightarrow B \\ A \rightarrow (S) & B \rightarrow SS & \end{array}$$

Context-Free Grammars (CFGs)

$$G = (\{S, A, B\}, \{(,)\}, S, R)$$

where R is defined as:

$$\begin{array}{lll} S \rightarrow \epsilon & S \rightarrow A & S \rightarrow B \\ A \rightarrow (S) & B \rightarrow SS & \end{array}$$

We often call the sequence of variables and symbols in the production rule a **right-hand side** (RHS) of the production rule.

Context-Free Grammars (CFGs)

$$G = (\{S, A, B\}, \{(,)\}, S, R)$$

where R is defined as:

$$\begin{array}{lll} S \to \epsilon & S \to A & S \to B \\ A \to (S) & B \to SS & \end{array}$$

We often call the sequence of variables and symbols in the production rule a **right-hand side** (RHS) of the production rule.

$$S \to \epsilon \mid A \mid B \qquad A \to (S) \qquad B \to SS$$

We can simplify the notation using the bar ($|$) notation by **combining** multiple production rules for the **same variable**.

Context-Free Grammars (CFGs)

```scala
// The definition of variables (nonterminals)
type Nt = String
// The type definitions of symbols (terminals)
type Symbol = Char
// The definition of right-hand side of a production rule
case class Rhs(seq: List[Nt | Symbol])
// The definition of context-free grammars
case class CFG(
  nts: Set[Nt],
  symbols: Set[Symbol],
  start: Nt,
  rules: Map[Nt, List[Rhs]],
)
```

Context-Free Grammars (CFGs)

```scala
// The definition of variables (nonterminals)
type Nt = String
// The type definitions of symbols (terminals)
type Symbol = Char
// The definition of right-hand side of a production rule
case class Rhs(seq: List[Nt | Symbol])
// The definition of context-free grammars
case class CFG(
  nts: Set[Nt],
  symbols: Set[Symbol],
  start: Nt,
  rules: Map[Nt, List[Rhs]],
)
```

```scala
// An example of CFG
val cfg: CFG = CFG(
  nts = Set("S", "A", "B"), symbols = Set('(', ')'), start = "S",
  rules = Map(
    "S" -> List(Rhs(List()), Rhs(List("A")), Rhs(List("B"))),
    "A" -> List(Rhs(List('(', "S", ')'))),
    "B" -> List(Rhs(List("S", "S")))
  ),
)
```

Derivation Relations

Definition (Derivation Relation ($\Rightarrow$))

Consider a CFG $G = (V, \Sigma, S, R)$. If a production rule $A \to \gamma \in R$ exists, the **derivation relation** $\Rightarrow \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ is defined as:

$$\alpha A \beta \Rightarrow \alpha \gamma \beta$$

for all $\alpha, \beta \in (V \cup \Sigma)^*$. We say that $\alpha A \beta$ **derives** $\alpha \gamma \beta$.

Derivation Relations

PLRG

Definition (Derivation Relation ($\Rightarrow$))

Consider a CFG $G = (V, \Sigma, S, R)$. If a production rule $A \to \gamma \in R$ exists, the **derivation relation** $\Rightarrow \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ is defined as:

$$\alpha A \beta \Rightarrow \alpha \gamma \beta$$

for all $\alpha, \beta \in (V \cup \Sigma)^*$. We say that $\alpha A \beta$ **derives** $\alpha \gamma \beta$.

Definition (Closure of Derivation Relation ($\Rightarrow^*$))

The **closure of derivation relation** $\Rightarrow^*$ is defined as:

- **(Basis Case)** $\forall \alpha \in (V \cup \Sigma)^*.\ \alpha \Rightarrow^* \alpha$
- **(Induction Case)** $\forall \alpha, \beta, \gamma \in (V \cup \Sigma)^*.\ (\alpha \Rightarrow^* \gamma)$ if

$$(\alpha \Rightarrow \beta) \land (\beta \Rightarrow^* \gamma)$$

Derivation Relations

$$G = (\{S, A, B\}, \{\texttt{(}, \texttt{)}\}, S, R)$$

$$S \to \epsilon \mid A \mid B \qquad A \to \texttt{(}S\texttt{)} \qquad B \to SS$$

A derivation for `(())()`:

Derivation Relations

$$G = (\{S, A, B\}, \{(,)\}, S, R)$$

$$S \to \epsilon \mid A \mid B \qquad A \to (S) \qquad B \to SS$$

A derivation for $(())()$:

$$\begin{aligned} S &\Rightarrow B &&\Rightarrow SS &&\Rightarrow AS &&\Rightarrow (S)S \\ &\Rightarrow (A)S &&\Rightarrow ((S))S &&\Rightarrow (())S &&\Rightarrow (())A \\ &\Rightarrow (())(S) &&\Rightarrow (())() \end{aligned}$$

Derivation Relations

$$G = (\{S, A, B\}, \{(,)\}, S, R)$$

$$S \to \epsilon \mid A \mid B \qquad A \to (S) \qquad B \to SS$$

A derivation for $(())()$:

$$\begin{array}{rlllllll} S & \Rightarrow & B & \Rightarrow & SS & \Rightarrow & AS & \Rightarrow & (S)S \\ & \Rightarrow & (A)S & \Rightarrow & ((S))S & \Rightarrow & (())S & \Rightarrow & (())A \\ & \Rightarrow & (())(S) & \Rightarrow & (())() \end{array}$$

Thus, we can **derive** (or generate/produce) the word $(())()$ from S:

$$S \Rightarrow^* (())()$$

Leftmost and Rightmost Derivations

- **Leftmost Derivation** ($\Rightarrow_L$): always derive the *leftmost* variable.
- **Rightmost Derivation** ($\Rightarrow_R$): always derive the *rightmost* variable.

Leftmost and Rightmost Derivations

- **Leftmost Derivation** ($\Rightarrow_L$): always derive the *leftmost* variable.
- **Rightmost Derivation** ($\Rightarrow_R$): always derive the *rightmost* variable.

$$G = (\{S, A, B\}, \{(,)\}, S, R)$$

$$S \rightarrow \epsilon \mid A \mid B \qquad A \rightarrow (S) \qquad B \rightarrow SS$$

Leftmost and Rightmost Derivations

- **Leftmost Derivation** ($\Rightarrow_L$): always derive the *leftmost* variable.
- **Rightmost Derivation** ($\Rightarrow_R$): always derive the *rightmost* variable.

$$G = (\{S, A, B\}, \{\texttt{(}, \texttt{)}\}, S, R)$$

$$S \rightarrow \epsilon \mid A \mid B \qquad A \rightarrow \texttt{(}S\texttt{)} \qquad B \rightarrow SS$$

For example, the **leftmost derivation** for `(())()`:

$$\begin{array}{llllllll} S & \Rightarrow_L & B & \Rightarrow_L & SS & \Rightarrow_L & AS & \\ & \Rightarrow_L & \texttt{(}S\texttt{)}S & \Rightarrow_L & \texttt{(}A\texttt{)}S & \Rightarrow_L & \texttt{((}S\texttt{))}S & \\ & \Rightarrow_L & \texttt{(())}S & \Rightarrow_L & \texttt{(())}A & \Rightarrow_L & \texttt{(())(}S\texttt{)} & \Rightarrow_L \;\; \texttt{(())()} \end{array}$$

Leftmost and Rightmost Derivations

PLRG

- **Leftmost Derivation** ($\Rightarrow_L$): always derive the *leftmost* variable.
- **Rightmost Derivation** ($\Rightarrow_R$): always derive the *rightmost* variable.

$$G = (\{S, A, B\}, \{\texttt{(}, \texttt{)}\}, S, R)$$

$$S \to \epsilon \mid A \mid B \qquad A \to \texttt{(}S\texttt{)} \qquad B \to SS$$

For example, the **leftmost derivation** for `(())()`:

$$\begin{array}{lllllll} S & \Rightarrow_L & B & \Rightarrow_L & SS & \Rightarrow_L & AS \\ & \Rightarrow_L & \texttt{(}S\texttt{)}S & \Rightarrow_L & \texttt{(}A\texttt{)}S & \Rightarrow_L & \texttt{((}S\texttt{))}S \\ & \Rightarrow_L & \texttt{(())}S & \Rightarrow_L & \texttt{(())}A & \Rightarrow_L & \texttt{(())(}S\texttt{)} \quad \Rightarrow_L \quad \texttt{(())()} \end{array}$$

and, the **rightmost derivation** for `(())()`:

$$\begin{array}{lllllll} S & \Rightarrow_R & B & \Rightarrow_R & SS & \Rightarrow_R & SA \\ & \Rightarrow_R & S\texttt{(}S\texttt{)} & \Rightarrow_R & S\texttt{()} & \Rightarrow_R & A\texttt{()} \\ & \Rightarrow_R & \texttt{(}S\texttt{)()} & \Rightarrow_R & \texttt{(}A\texttt{)()} & \Rightarrow_R & \texttt{((}S\texttt{))()} \quad \Rightarrow_R \quad \texttt{(())()} \end{array}$$

Sentential Forms

Definition (Sentential Form)

For a given CFG $G = (V, \Sigma, S, R)$, a sequence of variables or symbols $\alpha \in (V \cup \Sigma)^*$ is a **sentential form** if and only if $S \Rightarrow^* \alpha$.

Sentential Forms

Definition (Sentential Form)

For a given CFG $G = (V, \Sigma, S, R)$, a sequence of variables or symbols $\alpha \in (V \cup \Sigma)^*$ is a **sentential form** if and only if $S \Rightarrow^* \alpha$.

- α is a **left-sentential form** if $S \Rightarrow_L^* \alpha$.
- α is a **right-sentential form** if $S \Rightarrow_R^* \alpha$.

Definition (Sentential Form)

For a given CFG $G = (V, \Sigma, S, R)$, a sequence of variables or symbols $\alpha \in (V \cup \Sigma)^*$ is a **sentential form** if and only if $S \Rightarrow^* \alpha$.

- α is a **left-sentential form** if $S \Rightarrow_L^* \alpha$.
- α is a **right-sentential form** if $S \Rightarrow_R^* \alpha$.

For example, $(A)S$ is a **left-sentential form**:

$$S \Rightarrow_L B \Rightarrow_L SS \Rightarrow_L AS \Rightarrow_L (S)S \Rightarrow_L (A)S$$

Sentential Forms

Definition (Sentential Form)

For a given CFG $G = (V, \Sigma, S, R)$, a sequence of variables or symbols $\alpha \in (V \cup \Sigma)^*$ is a **sentential form** if and only if $S \Rightarrow^* \alpha$.

- α is a **left-sentential form** if $S \Rightarrow_L^* \alpha$.
- α is a **right-sentential form** if $S \Rightarrow_R^* \alpha$.

For example, $(A)S$ is a **left-sentential form**:

$$S \Rightarrow_L B \Rightarrow_L SS \Rightarrow_L AS \Rightarrow_L (S)S \Rightarrow_L (A)S$$

and, $S(S)$ is a **right-sentential form**:

$$S \Rightarrow_R B \Rightarrow_R SS \Rightarrow_R SA \Rightarrow_R S(S)$$

Context-Free Languages (CFLs)

Definition (Language of CFG)

For a given CFG $G = (V, \Sigma, S, R)$, the **language** of G is defined as:

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$$

Context-Free Languages (CFLs)

Definition (Language of CFG)

For a given CFG $G = (V, \Sigma, S, R)$, the **language** of G is defined as:

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$$

Definition (Context-Free Language)

A language L is **context-free language (CFL)** if and only if there exists a CFG G such that $L(G) = L$.

Context-Free Languages (CFLs)

Definition (Language of CFG)

For a given CFG $G = (V, \Sigma, S, R)$, the **language** of G is defined as:

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$$

Definition (Context-Free Language)

A language L is **context-free language (CFL)** if and only if there exists a CFG G such that $L(G) = L$.

$$G = (\{S, A, B\}, \{\texttt{(}, \texttt{)}\}, S, R)$$

$$S \rightarrow \epsilon \mid A \mid B \qquad A \rightarrow \texttt{(}S\texttt{)} \qquad B \rightarrow SS$$

Then, $\texttt{(())()} \in L(G)$ because $S \Rightarrow^* \texttt{(())()}$.

Example 1

What is the language of the following CFG?

$$G = (\{S, A, B\}, \{(,)\}, S, R)$$

$$S \to \epsilon \mid A \mid B \qquad A \to (S) \qquad B \to SS$$

Example 1

What is the language of the following CFG?

$$G = (\{S, A, B\}, \{(,)\}, S, R)$$

$$S \to \epsilon \mid A \mid B \qquad A \to (S) \qquad B \to SS$$

The language of G is:

$$L(G) = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

Example 1

What is the language of the following CFG?

$$G = (\{S, A, B\}, \{\texttt{(}, \texttt{)}\}, S, R)$$

$$S \to \epsilon \mid A \mid B \qquad A \to \texttt{(}S\texttt{)} \qquad B \to SS$$

The language of G is:

$$L(G) = \{w \in \{\texttt{(}, \texttt{)}\}^* \mid w \text{ is balanced}\}$$

In addition, it is equivalent to the following CFG:

$$S \to \epsilon \mid \texttt{(}S\texttt{)}S$$

Example 2

Define a CFG whose language is:

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

Example 2

Define a CFG whose language is:

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

The answer is:

$$S \to \epsilon \mid \mathtt{a}S\mathtt{b}$$

Example 3

Define a CFG whose language is:

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

Example 3

Define a CFG whose language is:

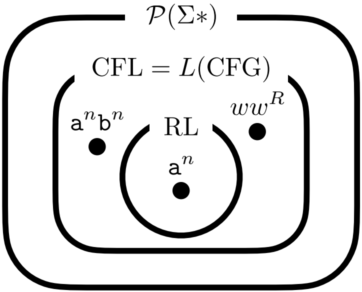
$$L = \{ww^R \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$$

The answer is:

$$S \to \epsilon \mid \mathtt{a}S\mathtt{a} \mid \mathtt{b}S\mathtt{b}$$

Summary

1. Context-Free Grammars (CFGs)
    - Definition
    - Derivation Relations
    - Leftmost and Rightmost Derivations
    - Sentential Forms
    - Context-Free Languages (CFLs)
    - Examples



Homework #2

- Please see this document on GitHub:

`https://github.com/ku-plrg-classroom/docs/tree/main/cose215/cfg-examples`

- The due date is 23:59 on Apr. 20 (Mon.).
- Please only submit **`Implementation.scala`** file to **LMS**.
- Late submission policy:
  - 1 day late: 20% penalty
  - 2 or more days late: no credit

Next Lecture

- Examples of Context-Free Grammars

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