Lecture 11 – Context-Free Grammars (CFGs) and Languages (CFLs)
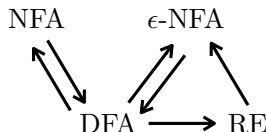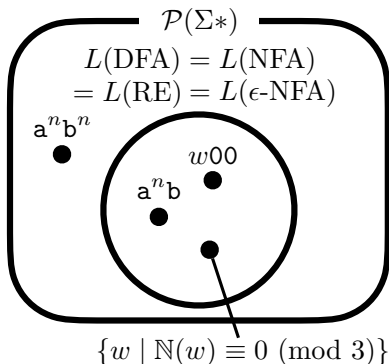
COSE215: Theory of Computation

Jihyeok Park

PLRG

2023 Spring

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

- Is there a way to describe languages that are not regular?

Contents

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:

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

PLRG

- Consider the following language:

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

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

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

- This language is **NOT regular**.
  (Do it yourself using the pumping lemma).

PLRG

- Consider the following language:

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

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

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

- This language is **NOT regular**.
  (Do it yourself using the pumping lemma).

- Then, how can we 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:

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

- This language is **NOT regular**.
  (Do it yourself using the pumping lemma).

- Then, how can we describe this language?
  Context-Free Grammars (CFGs)

Context-Free Grammars (CFGs)

Definition (Context-Free Grammar (CFG))

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

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

where

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

Definition (Context-Free Grammar (CFG))

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

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

where

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

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

where P is defined as:

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

Context-Free Grammars (CFGs)

```scala
// The type definitions of symbols and variables
type Symbol = Char
type Variable = String
// The definition of context-free grammars
case class CFG(
  variables: Set[Variable],
  symbols: Set[Symbol],
  start: Variable,
  productions: Set[(Variable, List[Variable | Symbol])],
)
// An example of CFG
val cfg1: CFG = CFG(
  variables = Set("S", "A", "B"),
  symbols = Set('(', ')'),
  start = "S",
  productions = Set(
    "S" -> Nil,
    "S" -> List("A"),
    "S" -> List("B"),
    "A" -> List('(', "S", ')'),
    "B" -> List("S", "S"),
  )
)
```

Derivation Relations

Definition (Derivation Relation ($\Rightarrow$))

Consider a CFG $G = (V, \Sigma, S, P)$. If a production rule $A \rightarrow \gamma \in P$ 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

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

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

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

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

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

Thus,

$$
\begin{array}{llll}
S \Rightarrow^* S & S \Rightarrow^* B & S \Rightarrow^* SS & \cdots \\
\cdots & S \Rightarrow^* (())A & S \Rightarrow^* (())(S) & S \Rightarrow^* (())()
\end{array}
$$

Leftmost and Rightmost Derivations

- **Leftmost Derivation** ($\overset{\text{lm}}{\Longrightarrow}$): always derive the *leftmost* variable.

- **Rightmost Derivation** ($\overset{\text{rm}}{\Longrightarrow}$): always derive the *rightmost* variable.

Leftmost and Rightmost Derivations

- **Leftmost Derivation** ($\xrightarrow{\text{lm}}$): always derive the *leftmost* variable.

- **Rightmost Derivation** ($\xrightarrow{\text{rm}}$): always derive the *rightmost* variable.

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

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

◆PLRG

- **Leftmost Derivation** ($\xrightarrow{\text{lm}}$): always derive the *leftmost* variable.

- **Rightmost Derivation** ($\xrightarrow{\text{rm}}$): always derive the *rightmost* variable.

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

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

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

$$
\begin{aligned}
S &\xrightarrow{\text{lm}} B &\xrightarrow{\text{lm}} SS &\xrightarrow{\text{lm}} AS &\xrightarrow{\text{lm}} (S)S &\xrightarrow{\text{lm}} (A)S \\
&\xrightarrow{\text{lm}} ((S))S &\xrightarrow{\text{lm}} (())S &\xrightarrow{\text{lm}} (())A &\xrightarrow{\text{lm}} (())(S) &\xrightarrow{\text{lm}} (())()
\end{aligned}
$$

- **Leftmost Derivation** ($\stackrel{\mathsf{lm}}{\Longrightarrow}$): always derive the *leftmost* variable.

- **Rightmost Derivation** ($\stackrel{\mathsf{rm}}{\Longrightarrow}$): always derive the *rightmost* variable.

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

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

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

$$
\begin{array}{llllll}
S & \stackrel{\mathsf{lm}}{\Longrightarrow} B & \stackrel{\mathsf{lm}}{\Longrightarrow} SS & \stackrel{\mathsf{lm}}{\Longrightarrow} AS & \stackrel{\mathsf{lm}}{\Longrightarrow} (S)S & \stackrel{\mathsf{lm}}{\Longrightarrow} (A)S \\
& \stackrel{\mathsf{lm}}{\Longrightarrow} ((S))S & \stackrel{\mathsf{lm}}{\Longrightarrow} (())S & \stackrel{\mathsf{lm}}{\Longrightarrow} (())A & \stackrel{\mathsf{lm}}{\Longrightarrow} (())(S) & \stackrel{\mathsf{lm}}{\Longrightarrow} (())()
\end{array}
$$

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

$$
\begin{array}{llllll}
S & \stackrel{\mathsf{rm}}{\Longrightarrow} B & \stackrel{\mathsf{rm}}{\Longrightarrow} SS & \stackrel{\mathsf{rm}}{\Longrightarrow} SA & \stackrel{\mathsf{rm}}{\Longrightarrow} S(S) & \stackrel{\mathsf{rm}}{\Longrightarrow} S() \\
& \stackrel{\mathsf{rm}}{\Longrightarrow} A() & \stackrel{\mathsf{rm}}{\Longrightarrow} (S)() & \stackrel{\mathsf{rm}}{\Longrightarrow} (A)() & \stackrel{\mathsf{rm}}{\Longrightarrow} ((S))() & \stackrel{\mathsf{rm}}{\Longrightarrow} (())()
\end{array}
$$

Sentential Forms

Definition (Sentential Form)

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

Definition (Sentential Form)

For a given CFG $G = (V, \Sigma, S, P)$, 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 $S \stackrel{\text{lm}}{\Longrightarrow}^* \alpha$.

- α is a **right-sentential form** if $S \stackrel{\text{rm}}{\Longrightarrow}^* \alpha$.

Definition (Sentential Form)

For a given CFG $G = (V, \Sigma, S, P)$, 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 $S \overset{\text{lm}}{\Longrightarrow}{}^* \alpha$.

- α is a **right-sentential form** if $S \overset{\text{rm}}{\Longrightarrow}{}^* \alpha$.

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

$$S \overset{\text{lm}}{\Longrightarrow} B \overset{\text{lm}}{\Longrightarrow} SS \overset{\text{lm}}{\Longrightarrow} AS \overset{\text{lm}}{\Longrightarrow} (S)S \overset{\text{lm}}{\Longrightarrow} (A)S$$

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

$$S \overset{\text{rm}}{\Longrightarrow} B \overset{\text{rm}}{\Longrightarrow} SS \overset{\text{rm}}{\Longrightarrow} SA \overset{\text{rm}}{\Longrightarrow} S(S)$$

Context-Free Languages (CFLs)

Definition (Language of CFG)

For a given CFG $G = (V, \Sigma, S, P)$, 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** if there exists a CFG G such that $LG = L$.

Definition (Language of CFG)

For a given CFG $G = (V, \Sigma, S, P)$, 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** if there exists a CFG G such that $LG = L$.

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

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

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

Example 1

What is the language of the following CFG?

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

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

Example 1

 PLRG

What is the language of the following CFG?

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

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

The language of G is:

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

Example 1

PLRG

What is the language of the following CFG?

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

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

The language of G is:

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

We can define this CFG in a more compact way using the bar ($|$) notation:

$$
\begin{array}{l}
S \to \epsilon \mid A \mid B \\
A \to (S) \\
B \to SS
\end{array}
$$

Example 1 **PLRG**

What is the language of the following CFG?

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

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

The language of G is:

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

We can define this CFG in a more compact way using the bar (|) notation:

$$
\begin{aligned}
S &\to \epsilon \mid A \mid B \\
A &\to (S) \\
B &\to SS
\end{aligned}
$$

In addition, it is equivalent to the following CFG:

$$S \to \epsilon \mid (S) \mid SS$$

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:

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

Example 3

OPLRG

Define a CFG whose language is:

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

Example 3

PLRG

Define a CFG whose language is:

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

The answer is:

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

- Midterm exam will be given in class.
- **Date:** 14:00-15:15 (1 hour 15 minutes), April 24 (Mon.).
- **Location:** 302, Aegineung (애기능생활관)
- **Coverage:** Lectures 1 – 13
- **Format:** short- or long-answer questions, including proofs
    - Closed book, closed notes
    - Some questions may require you to fill in the blanks in Scala code. But, it is about the concepts we have learned so far, not Scala itself.
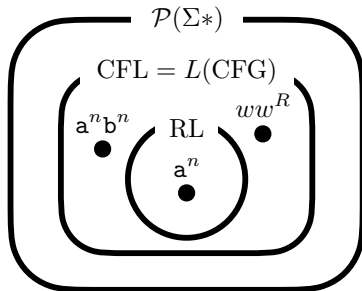
1. Context-Free Grammars (CFGs)

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

- Examples of Context-Free Grammars

Jihyeok Park

jihyeok_park@korea.ac.kr

https://plrg.korea.ac.kr