

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

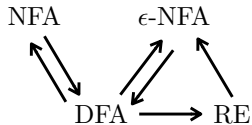
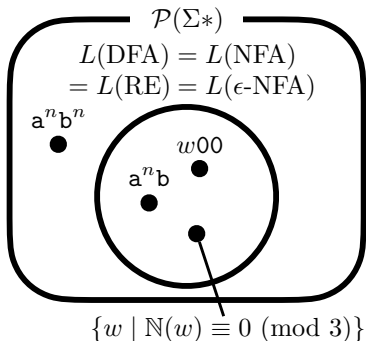
COSE215: Theory of Computation

Jihyeok Park



2026 Spring

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



	Automata	Grammars	Languages
(Part 3) Turing Machines	(Lecture 23) $\text{ETM} \rightleftharpoons \text{TM}$ (Lecture 21/22)	(Lecture 24) LC	(Lecture 21) REL (Lecture 26) \cup $\text{DL} \supset \text{NP} \stackrel{?}{=} \text{P}$ (Lecture 25) \supset
(Part 2) Pushdown Automata	(Lecture 14/15) $\text{PDA}_{\text{FS}} \rightleftharpoons \text{PDA}_{\text{ES}}$ \cup $\text{DPDA}_{\text{FS}} \supset \text{DPDA}_{\text{ES}}$ \cup (Lecture 17) $\not\subseteq$	(Lecture 16) CFG (Lecture 11/12) \vdots Chomsky Normal Form (Lecture 18)	(Lecture 11) CFL (Lecture 13) Parse Trees & Ambiguity \vdots Closure Properties (Lecture 19) Pumping Lemma (Lecture 20)
(Part 1) Finite Automata	(Lecture 4) $\text{NFA} \rightleftharpoons \text{DFA}$ (Lecture 3) (Lecture 5) $\text{DFA} \rightleftharpoons \epsilon\text{-NFA}$ (Lecture 7) (Lecture 6) $\epsilon\text{-NFA} \rightleftharpoons \text{RE}$ (Lecture 10) Equivalence & Minimization		(Lecture 3) RL \vdots Closure Properties (Lecture 8) Pumping Lemma (Lecture 9)
(Part 0) Basic Concepts	(Lecture 1) Mathematical Preliminaries	(Lecture 2) Scala	

- Consider the following language:

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

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

$L \ni \epsilon, (), (()), ()(), (())(), (())(), ((())), \dots$

$L \not\ni (,),)(), ((), ()), (())(), \dots$

- 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)**!

1. Context-Free Grammars (CFGs)

- Definition

- Derivation Relations

- Leftmost and Rightmost Derivations

- Sentential Forms

- Context-Free Languages (CFLs)

- Examples

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

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 $((\))$ $((\))(\)$ $(\)((\))(\)$...

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

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**.

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

where R is defined as:

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

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

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.

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

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

```
// 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 Rhc(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[Rhc]],
)
```

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

Definition (Derivation Relation (\Rightarrow))

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} S &\Rightarrow_R B && \Rightarrow_R SS && \Rightarrow_R SA \\ &\Rightarrow_R S(S) && \Rightarrow_R S() && \Rightarrow_R A() \\ &\Rightarrow_R (S)() && \Rightarrow_R (A)() && \Rightarrow_R ((S))() \Rightarrow_R (())() \end{aligned}$$

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**:

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

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 $L(G) = L$.

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

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

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

Example 1

What is the language of the following CFG?

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

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

The language of G is:

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

In addition, it is equivalent to the following CFG:

$$S \rightarrow \epsilon \mid (S)S$$

Example 2

Define a CFG whose language is:

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

The answer is:

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

Example 3

Define a CFG whose language is:

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

The answer is:

$$S \rightarrow \epsilon \mid aSa \mid bSb$$

1. Context-Free Grammars (CFGs)

Definition

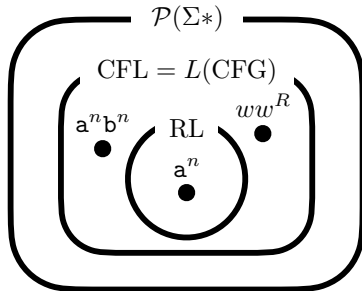
Derivation Relations

Leftmost and Rightmost Derivations

Sentential Forms

Context-Free Languages (CFLs)

Examples



- 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

- Examples of Context-Free Grammars

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