Lecture 12 – Examples of Context-Free Grammars

COSE215: Theory of Computation

Jihyeok Park

PLRG

2023 Spring

Recall

- A **context-free grammar (CFG)**:

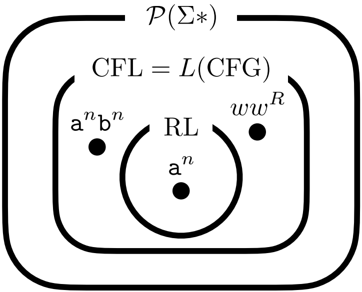
$$G = (V, \Sigma, S, P)$$

- The **language** of a CFG G:

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

- A language L is a **context-free language (CFL)**:

$$\exists \text{ CFG } G.\ L(G) = L$$



Contents

PLRG

Example 1: Regular Languages

PLRG

Theorem (RLs are CFLs)

If a language L is a regular language (RL), then L is a CFL.

Proof) For a given RE R, construct a CFG G such that $L(G) = L(R)$.

RE R	CFG G
$\varnothing$	$S \to S$
ϵ	$S \to \epsilon$
$a \in \Sigma$	$S \to a$
$R_1 \mid R_2$	$S \to S_1 \mid S_2$
$R_1 \cdot R_2$	$S \to S_1 S_2$
R_1^*	$S \to \epsilon \mid S_1 S$
(R_1)	$S \to S_1$

where S_1 and S_2 are start variables of CFGs G_1 and G_2 such that $L(G_1) = L(R_1)$ and $L(G_2) = L(R_2)$, respectively.

Example 1: Regular Languages

For a given RE R, construct a CFG G such that $L(G) = L(R)$.

- $R = \epsilon \,|\, \texttt{ab} \,|\, \texttt{ba}$

$$S \to F \mid D \qquad A \to \texttt{a} \qquad C \to AB \qquad E \to \epsilon$$
$$B \to \texttt{b} \qquad D \to BA \qquad F \to E \mid C$$

  Its simplified version:

$$S \to \epsilon \mid \texttt{ab} \mid \texttt{ba}$$

- $R = (\epsilon \,|\, \texttt{a})^*$

$$S \to \epsilon \mid AS \qquad A \to \epsilon \mid \texttt{a}$$

- $R = (\texttt{0} \,|\, \texttt{1}(\texttt{0}\texttt{1}^*\texttt{0})^*\texttt{1})^*$

$$S \to \epsilon \mid AE \qquad A \to \texttt{0} \mid \texttt{1}B\texttt{1} \qquad C \to \texttt{0}D\texttt{0}$$
$$B \to \epsilon \mid CB \qquad D \to \epsilon \mid \texttt{1}D$$

Example 2: $\mathtt{b}^n\mathtt{a}^m\mathtt{b}^{2n}$

Construct a CFG for the language:

$$L = \{\mathtt{b}^n\mathtt{a}^m\mathtt{b}^{2n} \mid n, m \geq 0\}$$

$$\begin{aligned} S &\to A \mid \mathtt{b}S\mathtt{bb} \\ A &\to \epsilon \mid \mathtt{a}A \end{aligned}$$

A derivation for `bbaaabbbb`:

$$\begin{aligned} S &\Rightarrow \mathtt{b}S\mathtt{bb} &&\Rightarrow \mathtt{bb}S\mathtt{bbbb} &&\Rightarrow \mathtt{bb}A\mathtt{bbbb} \\ &\Rightarrow \mathtt{bba}A\mathtt{bbbb} &&\Rightarrow \mathtt{bbaa}A\mathtt{bbbb} &&\Rightarrow \mathtt{bbaaa}A\mathtt{bbbb} \\ &\Rightarrow \mathtt{bbaaabbbb} \end{aligned}$$

Example 3: Well-Formed Brackets

Construct a CFG for the language:

$$L = \{ w \in \{ (,), \{, \}, [,] \}^* \mid w \text{ is well-formed} \}$$

$$S \to \epsilon \mid (S) \mid \{S\} \mid [S] \mid SS$$

A derivation for `({}){}[()[]]`:

$$\begin{array}{llllll}
S & \Rightarrow & SS & \Rightarrow & SSS & \Rightarrow & (S)SS \\
 & \Rightarrow & (\{S\})SS & \Rightarrow & (\{\})SS & \Rightarrow & (\{\})\{S\}S \\
 & \Rightarrow & (\{\})\{\}S & \Rightarrow & (\{\})\{\}[S] & \Rightarrow & (\{\})\{\}[SS] \\
 & \Rightarrow & (\{\})\{\}[(S)S] & \Rightarrow & (\{\})\{\}[()S] & \Rightarrow & (\{\})\{\}[()[S]] \\
 & \Rightarrow & (\{\})\{\}[()[]] & & & &
\end{array}$$

Example 4: Equal Number of a and b

Construct a CFG for the language:

$$L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) = N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.

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

The left-most derivation for abbaaabb:

$$\begin{array}{llllll} S & \overset{\text{lm}}{\Longrightarrow} & \mathtt{a}S\mathtt{b} & \overset{\text{lm}}{\Longrightarrow} & \mathtt{a}SS\mathtt{b} & \overset{\text{lm}}{\Longrightarrow} \ \mathtt{ab}S\mathtt{a}S\mathtt{b} \\ & \overset{\text{lm}}{\Longrightarrow} & \mathtt{abb}S\mathtt{aa}S\mathtt{b} & \overset{\text{lm}}{\Longrightarrow} & \mathtt{abbaa}S\mathtt{b} & \overset{\text{lm}}{\Longrightarrow} \ \mathtt{abbaaa}S\mathtt{bb} \\ & \overset{\text{lm}}{\Longrightarrow} & \mathtt{abbaaabb} & & & \end{array}$$

Example 5: Unequal Number of a and b

Construct a CFG for the language:

$$L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) \neq N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.

$$\begin{aligned}
S &\to P \mid N \\
P &\to ZP \mid \mathtt{a}P \mid \mathtt{a}Z \\
N &\to ZN \mid \mathtt{b}N \mid \mathtt{b}Z \\
Z &\to \epsilon \mid \mathtt{a}Z\mathtt{b} \mid \mathtt{b}Z\mathtt{a} \mid ZZ
\end{aligned}$$

The right-most derivation for aabbbaaab:

$$\begin{array}{llllll}
S \overset{\text{rm}}{\Longrightarrow} & P & \overset{\text{rm}}{\Longrightarrow} & ZP & \overset{\text{rm}}{\Longrightarrow} & Z\mathtt{a}Z \\
\overset{\text{rm}}{\Longrightarrow} & Z\mathtt{aa}Z\mathtt{b} & \overset{\text{rm}}{\Longrightarrow} & Z\mathtt{aab} & \overset{\text{rm}}{\Longrightarrow} & ZZ\mathtt{aab} \\
\overset{\text{rm}}{\Longrightarrow} & Z\mathtt{b}Z\mathtt{aaab} & \overset{\text{rm}}{\Longrightarrow} & Z\mathtt{baaab} & \overset{\text{rm}}{\Longrightarrow} & \mathtt{a}Z\mathtt{bbaaab} \\
\overset{\text{rm}}{\Longrightarrow} & \mathtt{aa}Z\mathtt{bbbaaab} & \overset{\text{rm}}{\Longrightarrow} & \mathtt{aabbbaaab} & &
\end{array}$$

Example 6: Arithmetic Expressions

An **arithmetic expression** is defined with the following CFG:

$$
\begin{aligned}
S &\to N \mid X \mid S\texttt{+}S \mid S\texttt{*}S \mid \texttt{(}S\texttt{)} \\
N &\to \texttt{0} \mid \cdots \mid \texttt{9} \mid \texttt{0}N \mid \cdots \mid \texttt{9}N \\
X &\to \texttt{a} \mid \cdots \mid \texttt{z}
\end{aligned}
$$

The left-most derivation for `13*(2+x)`:

$$
\begin{array}{llllll}
S & \xRightarrow{\text{lm}} S\texttt{*}S & \xRightarrow{\text{lm}} & N\texttt{*}S & \xRightarrow{\text{lm}} & \texttt{1}N\texttt{*}S \\
 & \xRightarrow{\text{lm}} \texttt{13*}S & \xRightarrow{\text{lm}} & \texttt{13*(}S\texttt{)} & \xRightarrow{\text{lm}} & \texttt{13*(}S\texttt{+}S\texttt{)} \\
 & \xRightarrow{\text{lm}} \texttt{13*(}N\texttt{+}S\texttt{)} & \xRightarrow{\text{lm}} & \texttt{13*(2+}S\texttt{)} & \xRightarrow{\text{lm}} & \texttt{13*(2+}X\texttt{)} \\
 & \xRightarrow{\text{lm}} \texttt{13*(2+x)} & & & &
\end{array}
$$

Example 7: Regular Expressions

Is the following language regular? or context-free?

$$L = \{w \in \{\varnothing, \epsilon, \texttt{a}, \texttt{b}, \mathsf{|}, \cdot, {}^*, \texttt{(}, \texttt{)}\}^* \mid w \text{ is a regular expression over } \{\texttt{a}, \texttt{b}\}\}$$

We can prove that L is not regular using the pumping lemma.
(Hint: consider a word $\texttt{(}^n\epsilon\texttt{)}^n$ for a given $n > 0$)
The language L is context-free:

$$S \to \varnothing \mid \epsilon \mid \texttt{a} \mid \texttt{b} \mid S\,\mathsf{|}\,S \mid S \cdot S \mid S^* \mid \texttt{(}S\texttt{)}$$

The right-most derivation for $\texttt{(}\epsilon\,\mathsf{|}\,\texttt{a}\cdot\texttt{b}\texttt{)}^*$:

$$\begin{array}{llllll} S & \overset{\text{rm}}{\Longrightarrow} & S^* & \overset{\text{rm}}{\Longrightarrow} & \texttt{(}S\texttt{)}^* & \overset{\text{rm}}{\Longrightarrow} & \texttt{(}S\,\mathsf{|}\,S\texttt{)}^* \\ & \overset{\text{rm}}{\Longrightarrow} & \texttt{(}S\,\mathsf{|}\,S\cdot S\texttt{)}^* & \overset{\text{rm}}{\Longrightarrow} & \texttt{(}S\,\mathsf{|}\,S\cdot\texttt{b}\texttt{)}^* & \overset{\text{rm}}{\Longrightarrow} & \texttt{(}S\,\mathsf{|}\,\texttt{a}\cdot\texttt{b}\texttt{)}^* \\ & \overset{\text{rm}}{\Longrightarrow} & \texttt{(}\epsilon\,\mathsf{|}\,\texttt{a}\cdot\texttt{b}\texttt{)}^* & & & & \end{array}$$

Example 8: Simplified Scala Syntax

We can define a CFG for a simplified version of Scala syntax[1]:

$$
\begin{array}{lrl}
\text{(Scala Program)} & S \rightarrow & E \mid E \; \texttt{;} \; S \\
\text{(Expressions)} & E \rightarrow & N \mid X \mid E \; \texttt{+} \; E \mid E \; \texttt{-} \; E \mid E \; \texttt{*} \; E \mid E \; \texttt{/} \; E \\
& \mid & \texttt{val} \; X \; \texttt{:} \; T \; \texttt{=} \; E \\
& \mid & \texttt{def} \; X \; \texttt{(} \; P \; \texttt{):} \; T \; \texttt{=} \; E \\
& \mid & E \; \texttt{(} \; A \; \texttt{)} \\
& \mid & \texttt{if} \; \texttt{(} \; E \; \texttt{)} \; E \; \texttt{else} \; E \\
& \mid & \texttt{trait} \; T \; \texttt{(} \; P \; \texttt{)} \\
& \mid & \texttt{case class} \; T \; \texttt{(} \; P \; \texttt{)} \\
& \mid & E \; \texttt{match} \; \texttt{\{} \; C \; \texttt{\}} \\
\text{(Numbers)} & N \rightarrow & \texttt{0} \mid \cdots \mid \texttt{9} \mid \texttt{0}N \mid \cdots \mid \texttt{9}N \\
\text{(Variables)} & X \rightarrow & A \mid AX \\
& A \rightarrow & \texttt{\_} \mid \texttt{a} \mid \cdots \mid \texttt{z} \mid \texttt{A} \mid \cdots \mid \texttt{Z} \\
\text{(Types)} & T \rightarrow & X \mid T \; \texttt{[} \; T \; \texttt{]} \mid T \; \texttt{=>} \; T \\
\text{(Parameters)} & P \rightarrow & \epsilon \mid X \; \texttt{:} \; T \mid P \; \texttt{,} \; X \; \texttt{:} \; T \\
\text{(Arguments)} & A \rightarrow & \epsilon \mid E \mid A \; \texttt{,} \; E \\
\text{(Cases)} & C \rightarrow & \texttt{case} \; E \; \texttt{=>} \; E \mid C \; \texttt{;} \; \texttt{case} \; E \; \texttt{=>} \; E
\end{array}
$$

[1] https://docs.scala-lang.org/scala3/reference/syntax.html

Example 8: Simplified Scala Syntax

```
def sum(n: Int): Int = n match { case 0 => 0; case n => n + sum(n - 1) }
```

The left-most derivation for this program:

$$
\begin{aligned}
S &\overset{\text{lm}}{\Longrightarrow} \texttt{def}\ X\ (\ P\ )\texttt{:}\ T\ \texttt{=}\ E \qquad \overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(}\ P\ \texttt{):}\ T\ \texttt{=}\ E \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(}\ X\texttt{:}\ T\ \texttt{):}\ T\ \texttt{=}\ E \qquad \overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int =}\ E \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int =}\ E\ \texttt{match \{}\ C\ \texttt{\}} \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int = n match \{}\ C\ \texttt{\}} \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int = n match \{ case}\ E\ \texttt{=>}\ E\ \texttt{;}\ C\ \texttt{\}} \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int = n match \{ case 0 => 0;}\ C\ \texttt{\}} \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int = n match \{ case 0 => 0; case}\ E\ \texttt{=>}\ E\ \texttt{\}} \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int = n match \{ case 0 => 0; case n =>}\ E\ \texttt{\}} \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int = n match \{ case 0 => 0; case n =>}\ E\ \texttt{+}\ E\ \texttt{\}} \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int = n match \{ case 0 => 0; case n => n +}\ E\ \texttt{\}} \\
&\overset{\text{lm}}{\Longrightarrow}{}^{*} \texttt{def sum(n: Int): Int = n match \{ case 0 => 0; case n => n + sum(n - 1) \}}
\end{aligned}
$$

Summary

1. Examples of Context-Free Grammars
   - Example 1: Regular Languages
   - Example 2: $\mathtt{b}^n\mathtt{a}^m\mathtt{b}^{2n}$
   - Example 3: Well-Formed Brackets
   - Example 4: Equal Number of `a` and `b`
   - Example 5: Unequal Number of `a` and `b`
   - Example 6: Arithmetic Expressions
   - Example 7: Regular Expressions
   - Example 8: Simplified Scala Syntax

Next Lecture

- Parse Trees and Ambiguity

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