Chap. 5 Context-Free Grammars

5.1 Context-Free Grammars

5.1.1 An Informal Example

madamimadam *"Madam, I'm Adam"*

A string w is ***palindrome****, if and only if* $w = w^R$.

Palindromes over {0, 1}

basis*:* ε, *0, and 1 are palindromes.*

induction*:* *If w is a palindrome, so are 0w0 and 1w1.*
No other string is palindrome.

Context-Free Grammar

1. $P \rightarrow \varepsilon$
2. $P \rightarrow 0$
3. $P \rightarrow 1$
4. $P \rightarrow 0P0$
5. $P \rightarrow 1P1$

5.1.2 Definition of Context-Free Grammars

A guadruple $G = (V, T, P, S)$ is a ***context-free grammar****, if*

1. *V is a* ***finite*** *set of* ***variables(nonterminals, syntactic categories)****,*
2. *T is a* ***finite*** *set of* ***terminal*** *symbols, where $V \cap T = \varnothing$,*
3. *P is a* ***finite*** *set of* ***productions(rules)****,*

   where each production is a pair (A, α), written $A \to \alpha$,

   $A \in V$ ***left part(head)*** *of production*

   $\alpha \in (V \cup T)^*$, ***right part(body)*** *of production*
4. *$S \in V$ is a distinguished variable, called* ***start(axiom) symbol****.*

Example 5.2

$G_{pal} = (\{P\}, \{0, 1\}, \{P \to \varepsilon, P \to 0, P \to 1, P \to 0P0, P \to 1P1\}, P)$

We write $A \to \alpha_1 \mid \ldots \mid \alpha_n \in P$ ***instead*** *of $A \to \alpha_1, \ldots, A \to \alpha_n \in P$.*

Example 5.2'

$G_{pal} = (\{P\}, \{0, 1\}, \{P \to \varepsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1\}, P)$

Example 5.3 regular expressions *over {***a**, **b**, **0**, **1***}*

$E \rightarrow E + E \mid EE \mid E^* \mid (E) \mid B$ ***induction***

$B \rightarrow \underline{\varepsilon} \mid \varnothing \mid \mathbf{a} \mid \mathbf{b} \mid \mathbf{0} \mid \mathbf{1}$ ***basis***

Note that $\underline{\varepsilon}$ *is not the* ***empty string*** *but a* ***symbol*** *for regular expression.*

$V = \{E, B\}$

$T = \{\underline{\varepsilon}, \varnothing, \mathbf{a}, \mathbf{b}, \mathbf{0}, \mathbf{1}, +, ^*, (,)\}$

Example 5.3 regular expression *revisited*

$E \rightarrow E + E \mid EE \mid E^* \mid (E) \mid \underline{\varepsilon} \mid \varnothing \mid \mathbf{a} \mid \mathbf{b} \mid \mathbf{0} \mid \mathbf{1}$

5.1.3 Derivation Using a Grammar

If $\alpha, \beta \in (V \cup T)^*$ *and* $A \in V$, *and* $A \to \gamma \in P$ *be a* ***production***.
We say string $\alpha A \beta$ ***directly derives*** $\alpha\gamma\beta$ *in CFG G, written*

$\alpha A \beta \Rightarrow_G \alpha\gamma\beta$, *we may omit G when it is understood,* $\alpha A \beta \Rightarrow \alpha\gamma\beta$.

$\Rightarrow \subseteq (V \cup T)^* \times (V \cup T)^*$ *a* ***binary relation on*** $(V \cup T)^*$.

$\to \subseteq \Rightarrow$ $\Rightarrow$ *is an* ***induced*** *binary relation from* $\to$.

Note that $\to$ *is* ***finite*** *but* $\Rightarrow$ *is* ***infinite***.

$\Rightarrow^* = \cup_{i \in N_0} \Rightarrow^i$. ***reflexive transitive closure*** *of* $\Rightarrow$.

iterative *definition for* $\Rightarrow^*$.

Recursive *definition for* $\Rightarrow^*$.

1. $\alpha \Rightarrow^* \alpha$, *for any* $\alpha \in (V \cup T)^*$. ***basis***
2. *If* $\alpha \Rightarrow^* \beta$, *and* $\beta \Rightarrow \gamma$, *then* $\alpha \Rightarrow^* \gamma$. ***recursion***

We say α ***derives*** β, *if* $\alpha \Rightarrow^* \beta$ *for some* $\alpha, \beta \in (V \cup T)^*$.

We say α *is a* ***sentential form*** *of G, if* $S \Rightarrow^* \alpha$ *for some* $\alpha \in (V \cup T)^*$.

We say w is a ***sentence*** *of G, if* $S \Rightarrow^* w$ *for some* $w \in T^*$.

The ***language*** *of G, denoted L(G), is* $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$.

A language L is ***context-free****, if there is a cfg G such that* $L = L(G)$.

Notational conventions for CFG

$a, b, c, \ldots \in T$	***terminal symbols***
$A, B, C, \ldots \in V$	***variable symbols***
$X, Y, Z, \ldots \in V \cup T$	***general symbols***
$x, y, z, \ldots \in T^*$	***terminal strings***
$\alpha, \beta, \gamma, \ldots \in (V \cup T)^*$	***general string***

Derivation** of CFG is **nondeterministic

1. *Which variable to be replaced*
2. *Which right hand side of the rule to be replaced*

***leftmost derivation**, $\Rightarrow_{lm}$, to replace **leftmost** variable*

$$S \Rightarrow_{lm}^{*} xA\gamma \Rightarrow_{lm} x\beta\gamma \Rightarrow_{lm}^{*} xy\gamma \Rightarrow_{lm}^{*} xyz$$

where $x, y, z \in T^{*}, \gamma \in (V \cup T)^{*}, A \rightarrow \beta \in P.$

***rightmost derivation**, $\Rightarrow_{rm}$, to replace **rightmost** variable*

$$S \Rightarrow_{rm}^{*} \alpha Az \Rightarrow_{rm} \alpha\beta z \Rightarrow_{rm}^{*} \alpha yz \Rightarrow_{rm}^{*} xyz$$

where $x, y, z \in T^{*}, \alpha \in (V \cup T)^{*}, A \rightarrow \beta \in P.$

Note that $\Rightarrow_{lm}, \Rightarrow_{rm} \subseteq \Rightarrow$.

5.2 Parse Trees

Let G = (V, T, P, S) be a cfg. The ***parse tree*** *for G are trees*

1. *Each* ***interior*** *node is labelled by a* ***variable*** $A \in V$
2. *Each* ***leaf*** *node is labelled by a* ***terminal*** $a \in T$ *or* ε.
3. *If an interior node is labelled A and its children are labelled* $X_1, X_2, \ldots, X_n$ *from left to right*

   $A \to X_1X_2\ldots X_n \in P$.

yield *of a tree*

concatenation *of* ***leaves*** *of a tree* ***from left to right***

Following four statements are ***equivalent*** *for some* ***terminal*** *string* $x \in T^*$.

(1) $A \Rightarrow^* x$,

(2) $A \Rightarrow_{lm}^* x$,

(3) $A \Rightarrow_{rm}^* x$,

(4) *There is a parse tree with* ***root*** *A and* ***yield*** *x.*

Proof *(2) ⇒ (1), (3) ⇒ (1) are trivial.*
(1) ⇒ (4): Thm. 5.12
(4) ⇒ (2), (4) ⇒ (3): Thm 5.14 and 5.16

Theorem 5.12 *If $A \Rightarrow^* x$, then there is a parse tree with* ***root*** *A and* ***yield*** *x*
Proof *Induction on* ***number*** *of* ***derivations***
Basis *$A \to x \in P$. ∴ Parse tree in Fig. 5.8(p187).*
Induction *Assume $A \to X_1X_2\ldots X_n \in P$, $1 \le^{\forall} i \le n$: $X_i \Rightarrow^* x_i$, and $x = x_1x_2\ldots x_n$*
1) If $X_i \in T$, $X_i = x_i$. ∴ ***parse tree*** *with* ***leaf*** *$x_i \in T$.*
2) If $X_i \in N$, $X_i \Rightarrow^* x_i$. ∴ ***parse tree*** *with* ***root*** *X_i and* ***yield*** *x_i.(IH)*
∴ $A \Rightarrow X_1X_2\ldots X_n \Rightarrow^* x_1X_2\ldots X_n \Rightarrow^* x_1x_2\ldots X_n \Rightarrow^* x_1x_2\ldots x_n$. and
parse tree *with* ***root*** *A and* ***yield*** *x.(Fig. 5.9; p188)*

Theorem 5.14 *If there is a parse tree with* ***root*** *A and* ***yield*** *x,* $A \Rightarrow_{lm}^{*} x$.

Proof *Induction on* ***height*** *of a tree*

Basis *Parse tree in Fig. 5.8.* $A \rightarrow x \in P$. $A \Rightarrow_{lm} x$.

Induction *Consider a* ***parse tree*** *with* ***root*** *A, sons* $X_1, X_2, \ldots, X_n$ *from left to right, and* ***yield*** $x = x_1x_2\ldots x_n$*.(Fig. 5.9)*

1) If $X_i \in T$, $X_i = x_i$. $X_i \Rightarrow_{lm}^{0} x_i$.

2) If $X_i \in N$, $X_i \Rightarrow_{lm}^{+} x_i$*.(IH)*

$A \Rightarrow_{lm} X_1X_2\ldots X_n \Rightarrow_{lm}^{*} x_1X_2\ldots X_n \Rightarrow_{lm}^{*} x_1x_2\ldots X_n \Rightarrow_{lm}^{*} x_1x_2\ldots x_n$.

Theorem 5.16 *If there is a parse tree with* ***root*** *A and* ***yield*** *x,* $A \Rightarrow_{rm}^{*} x$.

Proof *Induction on* ***height*** *of a tree(similar to leftmost derivation)*

$A \Rightarrow_{rm} X_1X_2\ldots X_n \Rightarrow_{rm}^{*} X_1X_2\ldots x_n \Rightarrow_{rm}^{*} X_1x_2\ldots x_n \Rightarrow_{rm}^{*} x_1x_2\ldots x_n$.

5.4 Ambiguity in Grammars and Languages

G_1: $E \to E + E \mid E * E \mid a \mid (E)$

$E \Rightarrow_{lm} E + E \Rightarrow_{lm} a + E \Rightarrow_{lm} a + E * E \Rightarrow_{lm}^{*} a + a * a.$

$E \Rightarrow_{lm} E * E \Rightarrow_{lm} E + E * E \Rightarrow_{lm}^{*} a + a * a.$

A grammar G is said to be ***ambiguous****, if* $^{\exists}x \in L(G)$ *.э.*

x has more than one ***parse trees****(****syntactic structure****),*

(x has more that one ***leftmost****(****rightmost****) derivation sequences)*

otherwise, ***unambiguous****.*

Derivation revisited

We may write $\alpha A\beta \Rightarrow^{r} \alpha\gamma\beta$, *if* $r = A \to \gamma \in P$.

Recursive extension of derivation with rules(rule string)

$\alpha \Rightarrow^{\varepsilon} \alpha$, *for* $\alpha \in (V \cup T)^{*}$, *and*

$\alpha \Rightarrow^{\pi r} \gamma$, *if* $\alpha \Rightarrow^{\pi} \beta$, $\beta \Rightarrow^{r} \gamma$ *for* $\pi \in P^{*}$, $r \in P$, $\alpha,\beta,\gamma \in (V \cup T)^{*}$.

Parser *of a grammar G,*

$^{\forall}x \in T^{*}$, *if* $x \in L(G)$ ***syntactic structure****(parse tree),*
otherwise say NO.

Is the parser for G is ***deterministic****?*

Not always!

It is ***undecidable*** *whether G is* ***ambiguous*** *or not.*

If G is ***unambiguous****, the parser for G may be* ***deterministic or not****.*
If G is ***ambiguous****, the parser for G is* ***nondeterministic****.*
If the parser for G is ***deterministic****, G is* ***unambiguous****.*

	parser	***structure***
regular	***deterministic***	***linear***
context-free	***nondeterministic***	***hierarchical***

Deterministic parsing** of **context-free languages

*If $S \Rightarrow_{lm}^{\pi} x$ for $x \in T^*$, $\pi \in P^*$ is called the **left parse** of the sentence x.*

*If $S \Rightarrow_{rm}^{\pi} x$ for $x \in T^*$, $\pi^R \in P^*$ is called the **right parse** of the sentence x.*

parse tree** ⇔ **left parse** ⇔ **right parse** ⇔ **syntactic structure

***left(right) parser**: **left(right)** parse*

*LL(k): **Left-to-right scan** in **Leftmost derivation** with k-lookahead symbols*
*LR(k): **Left-to-right scan** in **Rightmost derivation** with k-lookahead symbol*

Left parser *same direction in scan and derivation*
***normal order, top-down parsing**(LL parsing)*
Right parser *different direction in scan and derivation*
***reversed order, bottom-up parsing**(LR, LALR, SLR parsing)*

Removing ambiguity in the grammar

*Assume that precedence of * is higher than that of +, and*
*+ and * are left associative.*

G_2: $E \rightarrow E + T \mid T * F \mid a \mid (E)$
$T \rightarrow T * F \mid a \mid (E)$
$F \rightarrow a \mid (E)$

G_3: $E \rightarrow E + T \mid T$
$T \rightarrow T * F \mid F$
$F \rightarrow a \mid (E)$

$A \rightarrow B$ *is called* ***unit production****, if* $A, B \in V$.

*A **context free language** L is said to be **inherently ambiguous**,*
*if **every** cfg G .ə. L = L(G) is **ambiguous**.*

*$L = \{a^n b^n c^m d^m \mid n, m \geq 1\} \cup \{a^n b^m c^m d^n \mid n, m \geq 1\}$ is **inherently ambiguous**.*

Consider G: $S \rightarrow AB \mid C$
$A \rightarrow aAb \mid ab$
$B \rightarrow cBd \mid cd$
$C \rightarrow aCd \mid aDd$
$D \rightarrow bDc \mid bc$

and the sentence $a^n b^n c^n d^n$.

$S \Rightarrow_{lm} AB \Rightarrow_{lm} aAbB \Rightarrow_{lm}^{*} a^{n-1}Ab^{n-1}B \Rightarrow_{lm} a^n b^n B$
$\Rightarrow_{lm} a^n b^n cBd \Rightarrow_{lm}^{*} a^n b^n c^{n-1}Bd^{n-1} \Rightarrow_{lm} a^n b^n c^n d^n.$

$S \Rightarrow_{lm} C \Rightarrow_{lm} aCd \Rightarrow_{lm}^{*} a^{n-1}Cd^{n-1} \Rightarrow_{lm} a^n Dd^n$
$a^n bDcd^n \Rightarrow_{lm}^{*} a^n b^{n-1}Dc^{n-1}d^n \Rightarrow_{lm} a^n b^n c^n d^n.$