Lecture 14

Example:

G_1: $S \to AB$

$A \to aA \mid a$

$B \to bB \mid \varepsilon$

G_2: $S \to aS \mid aB$

$B \to bB \mid \varepsilon$

$$S \Rightarrow AB \overset{k}{\Rightarrow} a^kAB \Rightarrow a^{k+1}B \overset{m}{\Rightarrow} a^{k+1}b^mB \Rightarrow a^{k+1}b^m$$

$k+1 > 0$

$\Rightarrow k \geq 0$

$$L(G_1) = \{ a^k b^m \mid k > 0, \ m \geq 0 \}$$

$$= L(a^+ b^*)$$

$$= L(G_2)$$

$\Rightarrow$ G_1 and G_2 are equivalent.

Derivation Trees

Definition: Given a CFG $G = (V, \Sigma, R, S)$, a labeled ordered tree T is called a derivation tree in G if

- the root of T is labeled by some nonterminal of G.
- a node with label A (where $A \in V$) has children $Y_1, Y_2, \ldots, Y_k$, then there must be a rule in R s.t. $A \longrightarrow Y_1 \cdot Y_2 \cdot \ldots Y_k$.

Example: G: $S \to aB \mid bA$

$A \to aS \mid bAA \mid a$

$B \to bS \mid aBB \mid b$

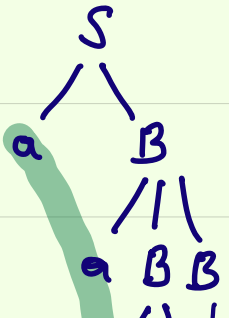
$w = aabaabb$

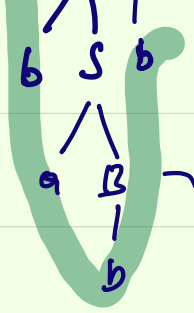
$S \Rightarrow a\underline{B} \Rightarrow aa\underline{BB} \Rightarrow aab\underline{S}B$

$\Rightarrow aabaB\underline{B}$

$\Rightarrow aababB$

$\Rightarrow aababb$





aabaabb

→ frontier of the derivation tree.

$S \Rightarrow^* w$ iff $\exists$ deriv. tree with frontier w.

* left-most derivation: Replace the nonterminal on the left-most position (at any step).

Designing CFGs

Although there exists no general algorithm to design a CFG for a given language, we use two practical methods:

1) Recursion technique
2) Decomposition technique

Recursion Technique:

Idea: Define the language recursively and extract rules from it.

Example: Set of palindromes P over $\Sigma = \{0,1\}$

Basis: $\lambda, 0, 1 \in P$

Rec. step: If $w \in P$, so are $0w0 \in P$ and $1w1 \in P$

No other rule is needed.

CFG for set of palindromes over $\Sigma = \{0,1\}$.

- Let S be a nonterminal that stands for a palindrome.

$S \to \lambda \mid 0 \mid 1 \mid 0S0 \mid 1S1$

Exercise: - CFG for even length palindromes?

- " " odd " " ?

Example: Dyck language D.. set of well-formed parentheses. $\Sigma = \{(,)\}$

e.g. $() \in D$ $((\notin D$

Basis: $\lambda \in D$

Rec. Step: If $x, y \in D$, then so are

(x) (or (y)) and $x \cdot y$ (or $y \cdot x$) $\in D$.

No other rule is needed.

e.g. $x = ()()$ $\quad y = (())$

$(x) = (()()) \in D$

$x \cdot y = ()() \cdot (()) \in D$.

$$S \rightarrow \lambda \mid (S) \mid S \cdot S$$

Example: CFG for $L = \{a^i b^j \mid i < j < 2i\}$

e.g. $ab^2 \notin L$

- $i = 0$ no j.
- $i = 1$ no j
- $i = 2 \Rightarrow 2 < j < 4 \Rightarrow j = 3$

$\Rightarrow a^2 b^3 \in L$

Basis: $a^2 b^3 \in L$

- $i = 3 \Rightarrow 3 < j < 6$

$j = 4, 5$

$\Rightarrow a^3 b^4$, $a^3 b^5$

$a^3b^4 = a \cdot a^2b^3 \cdot b$

$a^3b^5 = a \cdot a^2b^3 \cdot b^2$

- i

Rec. Step: If $w \in L$, so are awb and awb^2.

No other rule is required.

CFG G: $S \to a^2b^3 \mid aSb \mid aSb^2$

Exercise: CFG for $L = \{0^n 1^n \mid n \geq 0\}$?

Basis: ?

Rec. Step: ?

Decomposition Technique

Idea: Decompose the language into simpler ones.

Example: CFG for $L = \{a^i b^j \mid i \neq j\}$

e.g. $ab \notin L$ $ab^2 \in L$

$a^2b \in L$

$L = L_1 \cup L_2$ where

$L_1 = \{ a^i b^j \mid i < j \}$

$L_2 = \{ a^i b^j \mid i > j \}$

CFG for L_1 : $B \to aBb \mid Bb \mid b$

CFG for L_2 : $A \to aAb \mid aA \mid a$

CFG for $L_1 \cup L_2$: $S \to A \mid B$

$B \to aBb \mid Bb \mid b$

$A \to aAb \mid aA \mid a$

Correctness of CFGs *

L ... CFG G

Verify if $L = L(G)$

not in the scope of exams