Lecture 38

Context-Free Grammars and Languages

Example of a CFG

Variables: $\{P\}$

Terminals: $\{\epsilon, 0, 1\}$

Start Symbol: P

Productions:

$$P \to \epsilon$$
$$P \to 0$$
$$P \to 1$$
$$P \to 0P0$$
$$P \to 1P1$$

We can **derive** a string of terminals by starting from the start symbol and replacing variables repeatedly using productions.

For instance,

$$P \to 1P1 \to 10P01 \to 101P101 \to 1011101$$
$$P \to 0P0 \to 00P00 \to 001P100 \to 001100$$

Note: This grammar can generate all and only palindromes.

Definition of CFG

Definition: A CFG G is represented by its four components, that is, $G = (V, T, P, S)$, where

- V is a finite set called **variables**.
- T is a finite set, disjoint from V, called **terminals**.
- P is a set of **productions**, where each production consists of:
  - A variable called the **head** of the production.
  - The production symbol $\rightarrow$.
  - A string of 0 or more terminals and variables called the body of the production.
- $S \in V$ is a **start symbol**.

Definition: **Language** of a CFG G, denoted $L(G)$, is the set of all strings of terminals that we can obtain by starting from the start symbol and repeatedly replacing the variables with the body of one of its production.

Some CFGs

Example: Design a CFG for $L = \{w \mid w \text{ is not a palindrome}\}$

Solution: **Observation:** $w = w_1 w_2 \ldots w_n$ is not a palindrome if and only if there exist an $i \in [n]$ such that $w_i \neq w_{n+1-i}$.

Grammar for L:

Variables: $\{P, A\}$ **Terminals:** $\{\epsilon, 0, 1\}$ **Start Symbol:** P

Productions: $P \to 0P0 \mid 1P1 \mid 0A1 \mid 1A0$

$A \to \epsilon \mid 0A \mid 1A$

Notes: 1. $P \to 0A1 \mid 1A0$ ensure that the generated string is not palindrome.

2. A generates all strings in $\{0,1\}^*$.

Some CFGs

Example: Design a CFG for $L = \{a^i b^j c^k \mid i, j, k \geq 0$ and $i = j$ or $j = k\}$

Solution: **Idea:** Have two productions for the starting symbol. One that ensures $i = j$ and other that ensures $j = k$.

Grammar for L:

Variables: $\{P, A, C, X, Y\}$ **Terminals:** $\{\epsilon, a, b, c\}$ **Start Symbol:** P

Productions:

$$P \to XC \mid AY$$

$$X \to aXb \mid \epsilon$$

$$Y \to bYc \mid \epsilon$$

$$C \to cC \mid \epsilon$$

$$A \to aA \mid \epsilon$$

More on CFGs

Theorem: Every regular language is a CFL.

Proof: (H.W.) ■

Of course, not all the languages are CFLs. Below are some examples of non CFLs.

- $L = \{0^n 1^n 2^n \mid n \geq 0\}$
- $L = \{0^n 1^n 0^n 1^n \mid n \geq 0\}$
- $L = \{ww \mid w \in \{0,1\}^*\}$
- $L = \{0^n \mid n \text{ is a prime number}\}$
- $\ldots$

Ambiguity in Grammar

Definition: Derivation of a string of terminals where at each step we replace the left most variable by one of its production bodies is called a **leftmost derivation**.

Example: For the grammar $E \to E + E \mid E \times E \mid (E) \mid a$

and the string $a + a \times a$ the following are two different leftmost derivations.

- $E \to E \times E \to E + E \times E \to a + E \times E \to a + a \times E \to a + a \times a$
- $E \to E + E \to a + E \to a + E \times E \to a + a \times E \to a + a \times a$

Definition: A grammar is called **ambiguous** if there is at least one string in its language that has two or more different leftmost derivation.

Definition: A context free language L is said to be **inherently ambiguous** if all its grammars are ambiguous.