CS 205
Lecture 18
23|2|21

Context Free Languages

Ex : Language of balanced parentheses $\Sigma = \{(,)\}$

( ), (( )), ( )( ), (( ))( ), ... ✓

((,)(, ( )(, x

Inductive definition

(1) $\varepsilon \in L$

(2) If $x, y \in L$ then $xy \in L$ and $(x) \in L$.

Let E be the set of correct expressions (i.e. strings of balanced parentheses)

$$E \to \varepsilon$$
$$E \to EE$$
$$E \to (E)$$

Context-Free Grammar (CFG)

A CFG is a tuple $G = (V, T, S, P)$ where

V: finite set of variables (nonterminals, syntactic categories)

T: finite set of symbols called terminals

$S \in V$: start symbol

P: finite set of productions / rules of the form $A \to \alpha$ where $A \in V$, $\alpha \in (V \cup T)^*$

Ex $G = (\underbrace{\{E\}}_{V}, \underbrace{\{(,)\}}_{T}, \underbrace{E}_{S}, \underbrace{\{E \to \varepsilon,\ E \to EE,\ E \to (E)\}}_{P})$

Ex 1 : A string w is a palindrome if $w = w^R$.

CFG for the language of palindromes $\{0,1\}$

$G = (\{S\}, \{0,1\}, S, P)$ where P.

Contains the following productions

$$S \to \varepsilon$$
$$S \to 0$$
$$S \to 1$$
$$S \to 0S0$$
$$S \to 1S1$$

Written as $\boxed{S \to \varepsilon \mid 0 \mid 1 \mid 0S0 \mid 1S1}$

Ex 2 : Language of arithmetic expressions (E) built from integers (N) and identifiers (I), using only +, and $*$

$$G = (\underbrace{\{E, I, N\}}_{V}, \underbrace{\{a, b, 0, 1, (,), +, *, -\}}_{T}, \underset{S}{E}, \underset{\text{prod.}}{P})$$

$-$ → unary

$$\boxed{\begin{aligned} E &\to I \mid N \mid E+E \mid E*E \mid (E) \\ I &\to a \mid b \mid Ia \mid Ib \\ N &\to 0 \mid 1 \mid N0 \mid N1 \mid -N \mid +N \end{aligned}}$$

e.g. $\boxed{a} + \boxed{+2}$

$$\underset{\overline{E}}{I} + \underset{\overline{E}}{N}$$

Applications :
1. Parsing of programs (in compilers, say)
2. Markup languages : HTML, XML
3. Model of software

Derivations

Ex What are the strings in the language of arithmetic expressions ?

$E \Rightarrow E * \underline{E}$ using the production $E \to E * E$

$\Rightarrow E * \underline{N}$ 〃 $E \to N$

$\Rightarrow E * -N$ 〃 $N \to -N$

$\Rightarrow \underline{Id} * -N$ 〃 $E \to Id$

$\Rightarrow a * -\underline{N}$ 〃 $Id \to a$

$\Rightarrow a * -1$ 〃 $N \to 1$

Def : Let G be a CFG.

- We say $\boxed{\alpha A \beta \Rightarrow_G \alpha \gamma \beta}$ where $A \in V$, $\alpha, \beta, \gamma \in (V \cup T)^*$ and $A \to \gamma$ is a production of G
  ("single step derivation")
  - When G is understood, we write $\Rightarrow$
  - $\overset{*}{\Rightarrow}$: zero or more derivation steps of G (reflexive-transitive closure of $\Rightarrow$)
    e.g. $E \overset{*}{\Rightarrow} a * -N$
- The language generated by CFG G is
  $L(G) = \{w \in T^* \mid S \overset{*}{\Rightarrow} w\}$, i.e.
  the set of all strings of terminals derivable from the start symbol.
  - A language L is context-free if there is a CFG G s.t. $L = L(G)$.

Ex Palindromes $G : S \to \varepsilon \mid 0 \mid 1 \mid 0S0 \mid 1S1$

Claim $L(G) = \boxed{\{w \in \{0,1\}^* \mid w = w^R\}} \to$ Pal

Proof : Need to show

(1) $Pal \subseteq L(G)$

(2) $L(G) \subseteq Pal$

We show (1) by proving by induction on $|w|$ that $w \in Pal \Rightarrow S \overset{*}{\Rightarrow} w$

We show (2) by induction on the no. of derivation steps that

$S \overset{*}{\Rightarrow} w \Rightarrow w \in Pal$.