Context-free Languages (CFL)

Context Free Grammar (CFG)

- $G = (V, \Sigma, R, S)$
  - V: finite set of variables (nonterminals),
  - Σ: finite set of terminals. $V \cap \Sigma = \emptyset$,
  - $S \in V$: a special start symbol,
  - R: finite set of productions of form $A \rightarrow \alpha$, where $A \in V, \alpha \in (V \cup \Sigma)^*$.
- Conventions
  - $a, b, c, d \cdots \in \Sigma$
  - $A, B, C, D \cdots \in V$
  - $\alpha, \beta, \gamma \in (V \cup \Sigma)^*$

Language of a Grammar

- $G = (V, \Sigma, R, S)$
- Define **yield**, $\alpha_1 \Rightarrow \alpha_2$: if $(A \rightarrow \beta) \in R$, $uAv \Rightarrow u\beta v$.
- Define $\alpha_1 \Rightarrow^* \alpha_2$: if $\alpha_1 = \alpha_2$ or $\exists \beta_1, \beta_2, \cdots, \beta_k$ such that
  $\alpha_1 \Rightarrow \beta_1 \Rightarrow\Rightarrow \beta_2 \Rightarrow \cdots \Rightarrow \beta_k \Rightarrow \alpha_2$
  "A finite # of application of productions"
- The language of a grammar G, $L(G)$
  $L(G) = \{w \in \Sigma^* | S \Rightarrow^* w\}$
  All strings of terminals derivable from S
- L is a CFL iff $\exists$ CFG G such that $L = L(G)$
- Parse tree: a tree structured representation of a derivation.

Designing CFG

- Union of simpler CFGs

  For strings of form X or Y,

  use $S \rightarrow A|B$, where A derives X and B derives Y.

- Repeated symbols

  $A \rightarrow aA|a$

- Equal number of symbols

  $A \rightarrow bAc|\epsilon$

Ambiguity

- A grammar can generate the same string in different ways.
- Leftmost derivation:
  replace the leftmost variable at every step.
- Each parse tree has a unique leftmost derivation.
- A string is derived *ambiguously* in G if it has two or more leftmost derivations.
- G is *ambiguous* if it generates some strings ambiguously.
- A language L is inherently ambiguous if
  $\forall G$ such that $L(G) = L$, G is ambiguous.

Normal Forms

- Write grammars in simpler forms.
- Eliminate
  - ϵ-productions
  - unit productions
  - useless symbols
- Examples
  - Chomsky normal form (CNF)
  - Greibach Normal Form (GNF)

Chomsky Normal Form (CNF)

- Every production is of the form

$$A \to BC, \qquad A \to a$$

  where $a \in \Sigma$, $A, B, C \in V$, $B \neq S, C \neq S$.
  Also $S \to \epsilon$.

- Theorem: Any CFL is generated by a CFG in CNF.
  Proof: Convert any G into CNF.

Convert a CFG G into CNF

- Add a new start symbol
- Eliminate ϵ-productions
- Eliminate unit productions
- Eliminate useless symbols
- Convert rules into proper form
- Examples