Context-Free Grammars

- Dragon book Ch. 4.1 – 4.3

Outline

Introductory grammar theory

- Phase structure grammar
- Derivation and derivation tree
- Ambiguous grammar
- The Chomsky grammar hierarchy

Phrase Structure Grammars

- A *production* is written **α** → β or **α** ::= β
  - describes hierarchical structure of a language
  - Ex: if–else statements of C: if (expr) stmt else stmt

    stmt → if(*expr*) *stmt* else *stmt*

- A *phrase structure grammar* (PSG) G is a quadruple (N, T, P, S)
  - N : finite set of *Nonterminals*
  - T : finite set of *Terminals* (i.e., tokens)
  - P : Productions of the form **α** → β, where **α** must contain at least one nonterminal
  - S : Start symbol: S ∈ N

An Example of a PSG

- G1 = ({A,S}, {0,1}, P, S) where P is:
  - S $\rightarrow$ 0A1
  - 0A $\rightarrow$ 00A1
  - A $\rightarrow$ ε
  - EX: S $\Rightarrow$ 0A1 $\Rightarrow$ 00A11 $\Rightarrow$ 0011
  - What is the language of this grammar?

$$\{0^n1^n | n \geq 1\}$$

Notational Conventions

- Nonterminals: A, B, C, .., <stmt>
- Terminals: a, b, c, +, ..
- Strings of grammar symbols: α, β, …

  $\alpha = (N \cup T)^*$
- Strings of terminals: x, y, z, …

  $x = T^*$

Derivation, Sentence, Language

- If $\gamma\alpha\delta$ is a string in $(N \cup T)^*$ and $\alpha \rightarrow \beta$ is a production in G, then we say $\gamma\alpha\delta$ directly derives $\gamma\beta\delta$ and writes $\gamma\alpha\beta \Rightarrow \gamma\beta\delta$
  - $\Rightarrow^+$: one or more derivations
  - $\Rightarrow^*$: zero or more derivations
- If $S \Rightarrow^* \alpha$ then α is called a sentential form of G
- If $S \Rightarrow^* x$ then x is called a sentence of G
- The language generated by G, written L(G), is
  - $L(G) = \{x \mid x \in T^* \text{ and } S \Rightarrow^* x\}$
    - Ex: $L(G1) = \{0^n1^n \mid n \geq 1\}$

The Chomsky Hierarchy

- Type 0 – Unrestricted grammars
  - Any $\alpha \rightarrow \beta$
- Type 1 – Context sensitive grammar (CSG)
  - For all $\alpha \rightarrow \beta$, $|\alpha| \leq |\beta|$
  - Ex: G_2 = ({S,B,C}, {a,b,c}, P, S) and P is

$$
\begin{aligned}
S &\rightarrow aSBC \\
S &\rightarrow abC \\
CB &\rightarrow BC \\
bB &\rightarrow bb \\
bC &\rightarrow bc \\
cC &\rightarrow cc
\end{aligned}
$$

What is L(G2)?

$\{a^n b^n C^n \mid n \geq 1\}$

The Chomsky Hierarchy

- Type 2 – Context-free grammars (CFG)
  - For all $\alpha \rightarrow \beta$, $\alpha \in N$ (i.e., $A \rightarrow \beta$)
  - P of G_3 is: $E \rightarrow E + E | E * E | (E) | num | id$
    - What are derivations for **id + num * id** ?

      E → E + E → **id** + E → **id** + E * E → **id + num** * E → **id + num * id**
- Type 3 – Right or left linear grammars
  - Right-linear if all productions are of the form $A \rightarrow x$ or $A \rightarrow xB$
  - Left-Linear if all productions are of the form $A \rightarrow x$ or $A \rightarrow Bx$
  - Regular if all are $A \rightarrow a$ or $A \rightarrow aB$; Equivalent to regular languages
    - what is a grammar for **(a|b)*abb**?

      S → **a**S, S → **b**S, S → **a**B, B → **b**C, C → **b**D, D → ϵ
- In our compiler context, a grammar means the CFG.

Derivation and Derivation Tree

- Leftmost derivation and rightmost derivation
  - $E \to E + E$
    $\to id + E$
    $\to id + E * E$
    $\to id + num * E$
    $\to id + num * id$
- Parse (Derivation) Trees:
  Graphical Representation for a derivation
  - Internal nodes: Nonterminal
  - Leaves: Terminals

Ambiguous Grammars

- A grammar that produces more than one parse tree for some sentence
  - Produces more than one leftmost or more than one rightmost derivation
  - Eliminating ambiguity
  - Sometimes we use ambiguous grammars with *disambiguating* rules for simplicity of parsing

Inherently Ambiguous Languages

No unambiguous grammar that accepts it

- $L = \{0^i1^j2^k \mid i = j \text{ or } j = k;\ i, j, k \geq 0\}$
- One CFG for L is

  $S \rightarrow AB|CD$

  $A \rightarrow 0A|\varepsilon$

  $B \rightarrow 1B2|\varepsilon$

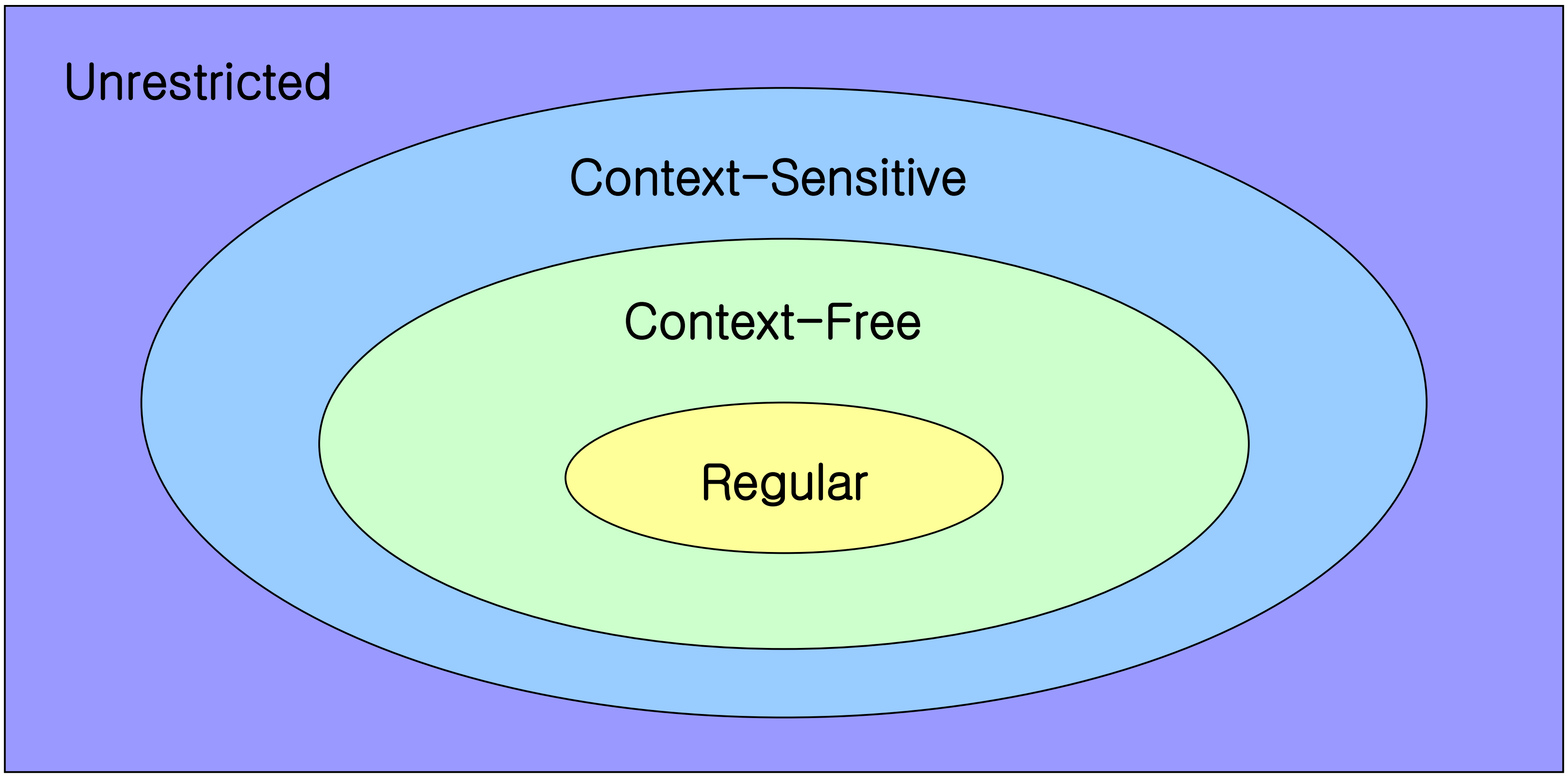
  $C \rightarrow 0C1|\varepsilon$

  $D \rightarrow 2D|\varepsilon$

  Why is this grammar ambiguous?

  - When deriving $\{0^i1^j2^k \mid i = j = k\}$

Languages and Grammar Hierarchy



Non Context-Free Languages

- $L = \{0^n1^n | n \geq 1\}$ is not regular
  - Is it context-free? Then, what is the CFG?
  - What about $\{0^n1^n2^n | n \geq 1\}$?
- Is $\{wcw | w \in (a|b)^*\}$ context free?
  - Check if a variable is used after declaration in C
    - C grammar does not specify characters in an identifier
    - Use a generic token **`id`** and rely on semantic analysis
  - What about $\{wcw^R | w \in (a|b)^*\}$?

- Is $L = \{a^n b^m c^n d^m \mid n, m \geq 1\}$ context free?
  - Checking if the number of formal parameters equals to that of actual parameters
  - C grammar cannot specify the number of parameters, so we rely on semantic analysis
    - What about $L = \{a^n b^n c^m d^m \mid n, m \geq 1\}$?
    - What about $L = \{a^n b^m c^m d^n \mid n, m \geq 1\}$?

"C" itself is not context-free, yet CFG is used for parsing and non-CFG features are handled by semantic analysis

Summary

- A phrase structure grammar and productions
- Grammar hierarchy
- Ambiguous grammar
- Non-CFG features of C