Introduction to Computer Science Theory

Alvin Lin

August 2017 - December 2017

Context Free Grammars

A **context-free grammar** is a 4-tuple $G = (V, \Sigma, R, S)$ where

- V is a finite set of variables (non-terminal symbols).
- Σ is a finite set of terminal symbols (terminals).
- R is the finite set of grammar rules (productions). Each rule is of the form $A \to \alpha$, where $A \in V$ and $\alpha \in (V \cup \Sigma)^*$.
- $S \in V$ is the start symbol.

Here is a CFG that generates $\{a^i b^i \mid i \geq 0\}$: $G = (V, \Sigma, R, S)$ where:

- $V = \{S\}$
- $\Sigma = \{a, b\}$
- $R = \{S \to \epsilon, S \to aSb\}$

$aaabbb$ is in the language generated by G because you can transform S into $aaabbb$ by a finite number of applications of the rules from R.

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$$

This is called a derivation of $aaabbb$. Notationally, we will use $\to$ in the rules and $\Rightarrow$ in the application of the rule in a string.

Definition

$L(G)$ (the language generated by G) is defined as $\{x \in \Sigma^* \mid S \stackrel{*}{=}_G x\}$. A language L is a context free language (CFL) if there is a context-free grammar G such that $L = L(G)$. These grammars are called context-free because if $A \to \gamma$ is a rule, then you can apply the rule to string $\alpha A \beta$, no matter what α and β are.

Example

$\{x \in \{a, b\}^* \mid x \text{ is a palindrome}\}$. $G = (V, \Sigma, R, S)$:

- $V = \{S\}$
- $\Sigma = \{a, b\}$
- $R = \{S \to \epsilon, S \to aSa, S \to bSb, S \to a, S \to b\}$

Derivation Trees

To specify what it means for two derivations to essentially be the same, we will look at derivation trees. A derivation tree is a tree representation of a derivation. Two derivations are essentially the same if and only if they correspond to the same derivation tree. Consider the grammar $G = (G, \Sigma, R, S)$

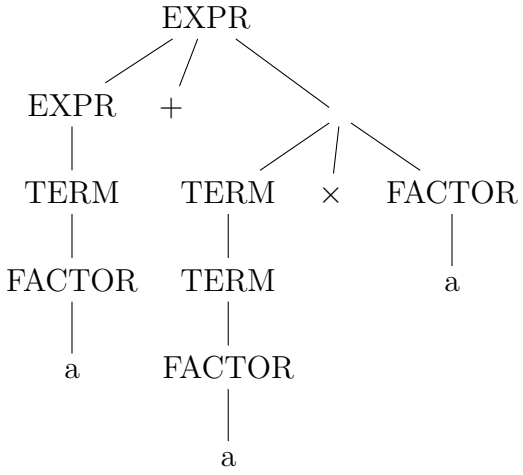
- $V = \{EXPR, TERM, FACTOR\}$
- $\Sigma = \{a, +, x, (,)\}$
- $R = \{$

$$\begin{aligned} EXPR &\to EXPR + TERM \\ EXPR &\to TERM \\ TERM &\to TERM \times FACTOR \\ TERM &\to FACTOR \\ FACTOR &\to (EXPR) \\ FACTOR &\to a \end{aligned}$$

  $\}$

- $S = EXPR$

The parse tree for the string $a + a \times a$ is as follows:



Ambiguity

A context-free grammar G is ambiguous if and only if there is a string $x \in L(G)$ such that x has two or more distinct derivation trees. $x + x \times x$ is ambiguous since order of precedence is ambiguous. Some languages are inherently ambiguous, but we are often able to find unambiguous grammars for certain context-free languages.

Machines for Context-Free Languages

Regular languages are exactly the languages accepted by finite automatons. Analogously, context-free languages are exactly the languages accepted by pushdown automaton.

Pushdown Automata

A pushdown automaton is the equivalent of adding a stack to a nondeterministic finite automaton. Each move of a pushdown automaton will be determined by the current state, the next input symbol, and the symbol on top of the stack. Each move consists of changing the state and either popping off the top character of the stack or pushing another character onto the stack. If the stack is empty, the pushdown automaton reaches a stuck state if it attempts to read. Pushdown automata start in an initial state and have a special start symbol on the stack. They accept a string x if and only if the pushdown automaton can get from the initial state to an accepting

state after processing all of x. A pushdown automaton is a sextuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where:

- Q is a finite set of states.
- Σ is a finite set called the input alphabet.
- Γ is a finite set called the stack alphabet.
- δ is the transition function from $Q \times \Sigma_\epsilon \times \Gamma_\epsilon$ to $2^{Q \times \Gamma_\epsilon}$.
- $q_0 \in Q$ is the initial state.
- $F \subseteq Q$ is the set of accepting states.

A pushdown automaton $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ accepts $w = w_1 w_2 \dots w_n$ where $w_1 w_2 \dots w_n \in \Sigma_\epsilon$, if there is a sequence of states $r_0, r_1, \dots, r_n \in Q$ and strings $s_0, s_1, \dots, s_n \in \Gamma^*$ such that:

- $r_0 = q_0$ and $s_0 = \epsilon$
- $\forall i \in \{0, \dots, n-1\}((r_{i+1}, b) \in \delta(r_i, w_{i+1}, a))$ where $s_i = at$ and $s_{i+1} = bt$ for $a, b \in \Gamma_\epsilon$ and $t \in \Gamma^*$
- $r_n \in F$

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech