Lecture 6: Context-Free Grammar

Instructor: Ketan Mulmuley
Scriber: Yuan Li

January 22, 2015

1 Context-Free Grammar vs Regular Expression

Recall that a CFG $G = (V, T, P, S \in V)$, where V is the set of vertices, T the set of terminals, P the set of production rules, and S the start symbol. All the production rules are of the form $A \to \alpha \in (V \cup T)^*$, where $A \in V$.

Proposition 1.1. *Every regular language has a context-free grammar.*

Every regular language has a regular expression. A formal proof is to convert it to a context-free grammar by induction. However, the proof idea is best illustrated by an example. Let L be a regular language of expression

$$\overbrace{(\overbrace{\overbrace{(0+1)}^{V}}^{U_1})^* + \overbrace{1}^{U_2}}^{T})^* \underbrace{00}_{R_2}.$$

R_1

It has a context-free grammar is as follows.

$$
\begin{aligned}
S &\to R_1 R_2 \\
R_1 &\to \epsilon \mid T R_1 \\
T &\to U_1 \mid U_2 \\
U_1 &\to \epsilon \mid V U_1 \\
V &\to 0 \mid 1 \\
U_2 &\to 1 \\
R_2 &\to 00
\end{aligned}
$$

On the other hand, not every context-free language is regular. For example, $L = \{0^n 1^n : n \geq 0\}$ is a CFL but not regular. Therefore, the set of regular languages is *strictly* contained in the set of context-free languages.

2 Derivation Tree

Definition 2.1. *Given a CFG G, a* derivation tree *is a tree such that*

- *each vertex has a label in $V \cup T \cup \{\epsilon\}$.*
- *root is labelled with S (start symbol).*
- *leaf vertex is terminal $\in T$.*
- *if a node has label A and the sons (from left to right) have labels $x_1, \ldots, x_k$, then $A \to x_1 \ldots x_k$ is a production rule in G.*
- *if a vertex is marked with ϵ, then it is a leaf and the only son of the father.*

Definition 2.2. *The* yield *of a derivation tree is a string obtained by reading the leaves from the left to right.*

If $A \Rightarrow B$ is a production then $xAy \Rightarrow xBy$, where $x, y \in (V \cup T)^*$. Let $\Rightarrow_G^*$ be the *reflexive transitive closure* of $\Rightarrow_G$.

Proposition 2.3. *Let G be a CFG. Then $S \Rightarrow_G^* \alpha$, where α is sentential form, if and only if there is a derivation tree with yield α.*

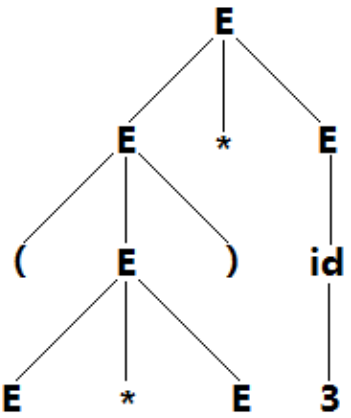

Figure 1: yield is $(E * E) * 3$

The proof is an easy standard induction, which is omitted.

Definition 2.4. *The* leftmost derivation tree *is a derivation tree such that at each stage expand the leftmost leaf (that can be expanded). The* rightmost derivation tree *is defined similarly.*

Definition 2.5. *We call CFG G* ambiguous *if some word in $L(G)$ has two different derivations. We call CFL L* (inherently) ambiguous *if every CFG G with $L = L(G)$ is ambiguous.*

For example, the following *grammar* is ambiguous

$$E \to E + E \mid E * E \mid (E) \mid \text{id}.$$

Because $x + y + z$ has two different derivations, namely, $(x + y) + z$ and $x+(y+z)$. However, the *language* is *not* ambiguous, which has the following nonambiguous grammar:

$$\begin{aligned} E &\to A + E \mid A * E \mid A \\ A &\to \text{id} \mid (E) \end{aligned}$$

Exercise 2.6 (*)**.** *Prove there exists a CFL L such that for any CFG G with $L = L(G)$, G is ambiguous. In words, prove there exists an inherently ambiguous CFL.*

3 Simplification of Grammar

Given a context-free grammar, we hope to transform the grammar into a normal form. That is, given CFG G, the goal is change G into a new grammar G' with $L(G) = L(G')$ such that G' has some desirable properties.

First, we can assume that each terminal or variable in G appears in the derivation of some word.

Definition 3.1. *A terminal or variable is* useless *if it does not occur in any derivation tree of any word in L.*

Now, let us design an algorithm to throw away all useless terminals and variables, which consists of top-down search and bottom-up search.

For the top-down search,

- first draw a graph with vertices in $V \cup T$ such that there is an edge from A to B if and only if there is a production rule in G of the form $A \to xBy$, where $x, y \in (V \cup T)^*$, and
- then do BFS on this graph starting at s to mark all variables that occur in some sentential form, and
- throw away all unmarked vertices.

For the bottom-up search,

- draw a similar graph on $V \cup T$ with arrow from $A \in V$ to $B \in V \cup T$ if there is a production rule of the form $A \to xBy$, and
- mark all terminals in this graph, and
- in each stage, if there is some unmarked vertex A with the production rule $A \to \alpha \in (V \cup T)^*$, in which each symbol in α is marked, then mark A, and
- throw away all unmarked variables at the end.

Now, let us eliminate productions of the form $A \to \epsilon$, called ϵ-productions.

Proposition 3.2. *Let G be a CFG. If $L = L(G)$, then $L \setminus \{\epsilon\}$ has a CFG with no useless symbols and ϵ-productions.*

Proof. Throw away all useless symbols using previous procedure. Call $A \in V$ *nullable* if $A \Rightarrow_G^* \epsilon$. Identify all nullable symbols:

- first mark all A's with production of the form $A \Rightarrow \epsilon$.
- (bottom-up search) if at each stage, there is an unmarked A with production of the form $A \Rightarrow \alpha \in V^*$ such that each variable in α is marked, then mark A.

After identifying all nullable variables, let us eliminate all productions of the kind $A \Rightarrow \epsilon$, and replace each production $B \to X_1 X_2 \cdots X_k$ by the set of productions obtained by removing *some* (all possible) set of nullable symbols in $X_1, \ldots, X_k$ (But do not take away all X_i's if every X_i is nullable). □

Finally, let us eliminate all productions of the form $A \to B$, called *unit productions*.

Proposition 3.3. *Every CFL L without ϵ is generated by some grammar with no useless symbols, ϵ-productions, or unit productions.*

Proof. Throw away all useless symbols and ϵ-productions as we discussed. Determine all $A, B \in V$ such that $A \Rightarrow_G^* B$ using graph theory search. If $A \Rightarrow_G^* B$, then for every production of the form $B \Rightarrow \alpha$, add production $A \Rightarrow \alpha$. □

4 Chomsky Normal Form

Definition 4.1. *CFG G is said to be in Chomsky normal form (CNF) if all productions in G are of the form*

$$A \to BC \text{ or } A \to a,$$

where B, C are variables, and a is a terminal.

Theorem 4.2. *Every CFL L without ϵ has a grammar G with $L = L(G)$ in Chomsky normal form.*

Let us continue the proof next time.