MA513: Formal Languages and Automata Theory
Topic: Properties of Context-free Languages
Lecture Number 28 Date: October 17, 2011

1 Eliminating Unit Productions

Definition: A **unit production** $\mathcal{U}$ in a CFG $G = (V, T, R, S)$ is a production (i.e., member of R) of the form $\mathcal{U} : A \rightarrow B$, where A and B are both variables. The variable B is said to be A**-derivable.**

For a variable $A \in V$ of the CFG $G = (V, T, R, S)$, the Algorithm 1 find the set of A**-derivable** variables.

Algorithm 1 Derivable(G, A)

Input: CFG $G = (V, T, R, S)$ and a variable $A \in V$.
Output: Set W of A-derivable variables in the grammar G.
$W \leftarrow \emptyset, W' \leftarrow \emptyset$
for (each production $A \rightarrow B \in R$) **do**
    $W = W \cup \{B\}$
end for
while ($W' \neq W$) **do**
    $W' = W$
    for (each $C \in W'$) **do**
        for (each $C \rightarrow B \in R$ such that $B \neq A$) **do**
            $W = W \cup \{B\}$
        end for
    end for
end while
Return (W)

• If $G = (V, T, R, S)$ is a CFG with no null production, then we can design a algorithm (see Algorithm 2) to find a CFG $G_1 = (V, T, R_1, S)$ having no unit production such that $L(G_1) = L(G)$.

Algorithm 2 Elimination_of_unit_Productions(G)

Input: CFG $G = (V, T, R, S)$
Output: CFG $G_1 = (V, T, R_1, S)$ having no unit production s. t. $L(G_1) = L(G)$
$R_1 \leftarrow R$
for (each $A \in V$) **do**
    W = Derivable(G, A) /* Using the Algorithm 1 */
    for (each $B \in W$) **do**
        for (each non-unit production $B \rightarrow \alpha \in R$) **do**
            if ($A \rightarrow \alpha \notin R_1$) **then**
                $R_1 = R_1 \cup \{A \rightarrow \alpha\}$
            end if
        end for
    end for
end for
Delete all unit productions from R_1
Return ($G_1 = (V, T, R_1, S)$)

Now, we can summarize the various simplifications on grammar described so far. We want to convert any CFG G into an equivalent CFG that has no **useless symbols,** **ϵ-productions**, or **unit productions**. Some care must be taken in the order of application of the constructions. A safe order is:

1. Eliminate ϵ-productions.
2. Eliminate unit productions.
3. Eliminate useless symbols.

Theorem: If G is a CFG generating a language that contains at least one string other than ϵ, then there is another CFG G_1 such that $L(G_1) = L(G) - \{\epsilon\}$, and G_1 has no ϵ-productions, unit productions, or useless symbols.

2 Chomsky Normal Form (CNF)

In this section, we shall show that every nonempty CFL without ϵ has a grammar G in which all productions are in one of two simple forms, either:

1. $A \rightarrow BC$, where each of the A, B, and C are variables, or
2. $A \rightarrow a$, where A is a variable and a is a terminal.

Further, G has no useless symbols. Such a grammar is said to be **Chomsky Normal Form**, or CNF.

To put a grammar in CNF, start with one that satisfies the following restrictions:

1. The grammar has no ϵ-productions,
2. The grammar has no unit productions, and
3. The grammar has no useless symbols.

Every production of such a grammar is either of the form $A \to a$, which is already in a form allowed by CNF, or it has a body [1] of length 2 or more. Our tasks are to:

a) Arrange all that bodies of length 2 or more consists only of variables.

b) Break bodies of length 3 or more into a cascade of productions, each with a body consisting of two variables.

Construction of (a): For every terminal a that appears in a body of length 2 or more, create a new variable, say A. This variable has only one production $A \to a$. Now, we use A in place of a everywhere a appears in body of length 2 or more. At this point, every production has a body that is either a single terminal or at least two variables and no terminals.

Construction of (b): We break all the productions of the form $A \to B_1 B_2 \dots B_k$, for $k \geq 3$ into a group of productions with two variables in each body. We introduce $k-2$ new variables, $C_1, C_2, \dots, C_{k-2}$. The original production is replaced by the following $k-1$ productions:

$A \to B_1 C_1$, $C_1 \to B_2 C_2$, $C_2 \to B_3 C_3$, $\dots$, $C_{k-2} \to B_{k-1} B_k$

Theorem: If G is a CFG whose language contains at least one string other than ϵ, then there is a grammar G_1 in **Chomsky Normal Form**, such that $L(G_1) = L(G) - \{\epsilon\}$.

[1] If $A \to \alpha$ is a production, then the part α is said to be **body** of that production.