CPS 140 - Mathematical Foundations of CS
Dr. S. Rodger
Section: Transforming Grammars (handout)

Methods for Transforming Grammars

We will consider CFL without ϵ. It would be easy to add ϵ to any grammar by adding a new start symbol S_0,

$$S_0 \rightarrow S \mid \epsilon$$

Definition: A production of the form $A \rightarrow Ax$, $A \in V$, $x \in (V \cup T)^*$ is *left recursive.*

Example Previous expression grammar was left recursive.

$$\begin{array}{l} E \rightarrow E+T \mid T \\ T \rightarrow T*F \mid F \\ F \rightarrow I \mid (E) \\ I \rightarrow a \mid b \end{array}$$

A top-down parser would want to derive the leftmost terminal as soon as possible. But in the left recursive grammar above, in order to derive a sentential form that has the leftmost terminal, we have to derive a sentential form that has other terminals in it.

Derivation of a+b+a+a is:

$E \Rightarrow E+T \Rightarrow E+T+T \Rightarrow E+T+T+T \stackrel{*}{\Rightarrow} a+T+T+T$

We will eliminate the left recursion so that we can derive a sentential form with the leftmost terminal and no other terminals.

Theorem (Removing Left recursion) Let G=(V,T,R,S) be a CFG. Divide productions for variable A into left-recursive and non left-recursive productions:

$$\begin{array}{l} A \rightarrow Ax_1 \mid Ax_2 \mid \ldots \mid Ax_n \\ A \rightarrow y_1 | y_2 | \ldots | y_m \end{array}$$

where x_i, y_i are in $(V \cup T)^*$.

Then G'=(V$\cup${Z}, T, R', S) and R' replaces rules of form above by

$$\begin{array}{l} A \rightarrow y_i | y_i Z, \ i=1,2,\ldots,m \\ Z \rightarrow x_i | x_i Z, \ i=1,2,\ldots,n \end{array}$$

Example:

$E \rightarrow E+T|T$ becomes

$T \rightarrow T*F|F$ becomes

Now, Derivation of a+b+a+a is:

Useless productions

S $\rightarrow$ aB | bA
A $\rightarrow$ aA
B $\rightarrow$ Sa | a
C $\rightarrow$ cBc | a

Theorem (useless productions) Let G be a CFG. Then $\exists$ G' that does not contain any useless variables or productions s.t. L(G)=L(G').

To Remove Useless Productions:

Let G=(V,T,R,S).

I. Compute V_1={Variables that can derive strings of terminals}

1. $V_1=\emptyset$
2. Repeat until no more variables added
   - For every A$\in$V with A$\rightarrow x_1 x_2 \ldots x_n$, $x_i \in (T^* \cup V_1)$, add A to V_1
3. R_1 = all productions in R with symbols in $(V_1 \cup T)^*$

Then $G_1=(V_1,T,R_1,S)$ has no variables that can't derive strings.

II. Draw Variable Dependency Graph

For A $\rightarrow$ xBy, draw A$\rightarrow$B.

Remove productions for V if there is no path from S to V in the dependency graph. Resulting Grammar G' is s.t. L(G)=L(G') and G' has no useless productions.

Theorem (remove ϵ productions) Let G be a CFG with ϵ not in L(G). Then $\exists$ a CFG G' having no ϵ-productions s.t. L(G)=L(G').

To Remove ϵ-productions

1. Let V_n = {A | $\exists$ production A$\rightarrow\epsilon$ }
2. Repeat until no more additions
   - if B$\rightarrow A_1 A_2 \ldots A_m$ and $A_i \in V_n$ for all i, then put B in V_n
3. Construct G' with productions R' s.t.
   - If A$\rightarrow x_1 x_2 \ldots x_m \in$R, $m \geq 1$, then put all productions formed when x_j is replaced by ϵ (for all $x_j \in V_n$) s.t. |rhs| ≥ 1 into R'.

Example:

S $\rightarrow$ Ab
A $\rightarrow$ BC | Aa
B $\rightarrow$ b | ϵ
C $\rightarrow$ cC | ϵ

Definition Unit Production

$$A \rightarrow B$$

where A,B $\in$V.

Consider removing unit productions:

Suppose we have

A $\rightarrow$ B   becomes
B $\rightarrow$ a | ab

But what if we have

A $\rightarrow$ B   becomes
B $\rightarrow$ C
C $\rightarrow$ A

Theorem (Remove unit productions) Let G=(V,T,R,S) be a CFG without ϵ-productions. Then $\exists$ CFG G'=(V',T',R',S') that does not have any unit-productions and L(G)=L(G').

To Remove Unit Productions:

1. Find for each A, all B s.t. A $\stackrel{*}{\Rightarrow}$B (Draw a dependency graph)
2. Construct G'=(V',T',R',S') by
   (a) Put all non-unit productions in R'
   (b) For all A$\stackrel{*}{\Rightarrow}$B s.t. B$\rightarrow y_1|y_2|\ldots y_n \in$ R', put A$\rightarrow y_1|y_2|\ldots y_n \in$ R'

Theorem Let L be a CFL that does not contain ϵ. Then $\exists$ a CFG for L that does not have any useless productions, ϵ-productions, or unit-productions.

Proof

1. Remove ϵ-productions
2. Remove unit-productions
3. Remove useless productions

Note order is very important. Removing ϵ-productions can create unit-productions! QED.

Definition: A CFG is in Chomsky Normal Form (CNF) if all productions are of the form

$$A \rightarrow BC \quad \text{or } A \rightarrow a$$

where A,B,C$\in$V and a$\in$T.

Theorem: Any CFG G with ϵ not in L(G) has an equivalent grammar in CNF.

Proof:

1. Remove ϵ-productions, unit productions, and useless productions.
2. For every rhs of length > 1, replace each terminal x_i by a new variable C_j and add the production $C_j \to x_i$.
3. Replace every rhs of length > 2 by a series of productions, each with rhs of length 2. QED.

Example:

$$\begin{array}{l} S \to CBcd \\ B \to b \\ C \to Cc \mid e \end{array}$$

Definition: A CFG is in Greibach normal form (GNF) if all productions have the form

$$A \to ax$$

where $a \in T$ and $x \in V^*$

Theorem For every CFG G with ϵ not in L(G), $\exists$ a grammar in GNF.