Normal Forms for Context-Free Grammars

CSE 322: Introduction to Formal Models in Computer Science
May 22, 1998

1. Putting a Context-Free Grammar in Normal Form

Definition: A context-free grammar $G = (V, \Sigma, R, S)$ is in *normal form* if and only if R contains no rules of the form

1. $A \rightarrow e$, for any $A \in V - \Sigma$, or
2. $A \rightarrow B$, for any $A, B \in V - \Sigma$.

Here is a procedure for converting a grammar G into a grammar G' such that G' is in normal form, and $L(G') = L(G) - \{e\}$. Throughout the procedure, A and B are arbitrary elements of $V - \Sigma$, and u and v are arbitrary strings in V^*.

1. (a) For every pair of rules $A \rightarrow e$ and $B \rightarrow uAv$, add a new rule $B \rightarrow uv$. Continue doing this until no new rule can be added by this procedure.
   (b) Remove all rules $A \rightarrow e$. [1]
2. (a) For every pair of rules $A \rightarrow B$ and $B \rightarrow u$, add a new rule $A \rightarrow u$. Continue doing this until no new rule can be added by this procedure.
   (b) Remove all rules $A \rightarrow B$.

2. Example

Put $G = (V, \Sigma, R, S)$ in normal form, where

$$\begin{aligned} V &= \{S, A, B, a, b\}, \\ \Sigma &= \{a, b\}, \text{ and} \\ R &= \{S \rightarrow A,\ A \rightarrow SB,\ A \rightarrow B,\ B \rightarrow aAbB,\ B \rightarrow e\}. \end{aligned}$$

(Since $e \in L(G)$, the resulting normal form grammar will generate $L(G) - \{e\}$.)

[1] If $S \rightarrow e$ is removed in this step, then $L(G) - L(G') = \{e\}$; otherwise, $L(G) = L(G')$.

1. (a) Add $A \rightarrow S,\ A \rightarrow e,\ B \rightarrow aAb$.
   Add $S \rightarrow e,\ B \rightarrow abB,\ B \rightarrow ab$.

   (b) Remove $A \rightarrow e,\ B \rightarrow e,\ S \rightarrow e$.

   At this point, the set of rules is

$$
\begin{array}{l}
\{S \rightarrow A, \\
A \rightarrow SB,\ A \rightarrow S,\ A \rightarrow B, \\
B \rightarrow aAbB,\ B \rightarrow aAb,\ B \rightarrow abB,\ B \rightarrow ab\}.
\end{array}
$$

2. (a) Add $S \rightarrow SB,\ S \rightarrow S,\ S \rightarrow B$,
   $A \rightarrow A,\ A \rightarrow aAbB,\ A \rightarrow aAb,\ A \rightarrow abB,\ A \rightarrow ab$.
   Add $S \rightarrow aAbB,\ S \rightarrow aAb,\ S \rightarrow abB,\ S \rightarrow ab$.

   (b) Remove $S \rightarrow S,\ S \rightarrow A,\ S \rightarrow B,\ A \rightarrow S,\ A \rightarrow A,\ A \rightarrow B$.

   The final set of rules is

$$
\begin{array}{l}
\{S \rightarrow SB,\ S \rightarrow aAbB,\ S \rightarrow aAb,\ S \rightarrow abB,\ S \rightarrow ab, \\
A \rightarrow SB,\ A \rightarrow aAbB,\ A \rightarrow aAb,\ A \rightarrow abB,\ A \rightarrow ab, \\
B \rightarrow aAbB,\ B \rightarrow aAb,\ B \rightarrow abB,\ B \rightarrow ab\}.
\end{array}
$$

As an example of how the equivalence works, consider the following derivation in the original grammar G: $S \Rightarrow A \Rightarrow B \Rightarrow aAbB \Rightarrow aSBbB \Rightarrow aABbB \Rightarrow aBBbB \Rightarrow aaAbBBbB \Rightarrow aaBbBBbB \Rightarrow aabBBbB \Rightarrow aabBbB \Rightarrow aabaAbBbB \Rightarrow aabaBbBbB \Rightarrow aababBbB \Rightarrow aababbB \Rightarrow aababbaAbB \Rightarrow aababbaBbB \Rightarrow aababbabB \Rightarrow aababbab$.

This is simulated in the normal form grammar by the following derivation of the same terminal string: $S \Rightarrow aAbB \Rightarrow aSBbB \Rightarrow aabBbB \Rightarrow aababbB \Rightarrow aababbab$.

3. Putting a Context-Free Grammar in Chomsky Normal Form

Definition: A context-free grammar $G = (V, \Sigma, R, S)$ is in *Chomsky normal form* if and only if every rule in R is of one of the following forms:

1. $A \rightarrow a$, for $A \in V - \Sigma$ and $a \in \Sigma$, or

2. $A \rightarrow BC$, for $A, B, C \in V - \Sigma$.

Here is a procedure for putting a normal form grammar in Chomsky normal form, without changing the language generated by the grammar. Throughout the procedure, A and

$B_1, B_2, \ldots, B_m$ are nonterminals, and $X_1, X_2, \ldots X_m$ are arbitrary elements in V.

1. For each terminal symbol a, add a new nonterminal C_a and a new rule $C_a \to a$.
2. Let $A \to X_1 X_2 \cdots X_m$ be a rule, with $m \geq 2$. For each $1 \leq i \leq m$, if X_i is a terminal symbol a, replace X_i in the right hand side of the original rule by C_a.
3. Let $A \to B_1 B_2 \cdots B_m$ be a rule, with $m \geq 3$. Add new nonterminals $D_1, D_2, \ldots, D_{m-2}$, and replace the rule $A \to B_1 B_2 \cdots B_m$ by the rules

$$A \to B_1 D_1,\ D_1 \to B_2 D_2,\ \ldots, D_{m-3} \to B_{m-2} D_{m-2},\ D_{m-2} \to B_{m-1} B_m.$$

4. Example

Put $G = (V, \Sigma, R, S)$ in Chomsky normal form, where

$$\begin{aligned} V &= \{S, A, a, b\}, \\ \Sigma &= \{a, b\}, \text{ and} \\ R &= \{S \to aAb,\ A \to aAbS,\ A \to b\}. \end{aligned}$$

Notice that G is already in normal form.

The result of steps 1 and 2 is $G' = (V', \Sigma, R', S)$, where

$$\begin{aligned} V' &= \{S, A, C_a, C_b, a, b\}, \\ \Sigma &= \{a, b\}, \text{ and} \\ R' &= \{S \to C_a A C_b,\ A \to C_a A C_b S,\ A \to b,\ C_a \to a,\ C_b \to b\}. \end{aligned}$$

The result of step 3 is $G'' = (V'', \Sigma, R'', S)$, where

$$\begin{aligned} V'' &= \{S, A, C_a, C_b, D_1, E_1, E_2, a, b\}, \\ \Sigma &= \{a, b\}, \text{ and} \\ R'' &= \{S \to C_a D_1,\ D_1 \to A C_b,\ A \to C_a E_1,\ E_1 \to A E_2,\ E_2 \to C_b S, \\ &\quad A \to b,\ C_a \to a,\ C_b \to b\}. \end{aligned}$$

G'' is in Chomsky normal form.