Theory of Computation

Tutorial - Normal Forms

Cesare Spinoso-Di Piano

Plan for today

1. Production simplification

2. Normal forms

Production simplification

λ-productions and nullable variables

Definition. A λ-**production** is any production of the form

$$A \to \lambda$$

Definition. A nullable variable is any variable such that

$$A \Rightarrow^* \lambda$$

Removing λ-productions and nullable variables

Let $G = (V, T, S, P)$ be any CFG such that $\lambda \notin L(G)$. Then we can always convert G to an equivalent grammar $G' = (V', T', S', P')$ with no λ-productions or nullable variables.

1. Identify all nullable variables V_N.
2. For each production P in G

$$A \to x_1 x_2 \ldots x_m, x_i \in T \cup V$$

   put P in G'.
3. Add to G' every possible production resulting from replacing some nullable variable x_j by λ (except for $A \to \lambda$).

Example

Example. Suppose G has the following productions

$$S \to aABC$$
$$A \to \lambda | BC$$
$$B \to bb | C$$
$$C \to \lambda | c$$

The nullable variables are A, B, C. The simplified grammar is

$$S \to aABC | aAB | aBC | aAC | aA | aB | aC | a$$
$$A \to BC | B | C$$
$$B \to bb | C$$
$$C \to c$$

Exercise

Exercise. Eliminate all $\lambda-$productions from

$$S \to AaB|aaB$$
$$A \to \lambda$$
$$B \to bbA|\lambda$$

Unit production and removing them

Definition. Any production of the form $A \rightarrow B$ from a grammar $G = (V, T, S, P)$ where $A, B \in V$ is a **unit-production**.

Given a grammar $G = (V, T, S, P)$ with unit-productions convert it to an equivalent grammar $G' = (V', T', S', P')$ without any unit-productions as follows:

1. Remove any productions $A \rightarrow A$.
2. Find all variables $A, B \in V$ such that $A \Rightarrow^* B$.
3. Put all the non-unit productions in P'
4. For every pair A, B identified in step 1., if $B \rightarrow x_1 | ... | x_n$ are productions in P', then add $A \rightarrow x_1 | ... | x_n$ to P'.

Example

Example. Suppose the following productions rules are found in G

$$S \rightarrow A|bBb|C$$
$$A \rightarrow a|bC|C$$
$$B \rightarrow b$$
$$C \rightarrow abA|CC|B$$

Removing all unit-productions and adding productions accordingly gives the following grammar

$$S \rightarrow bBb|a|bC|abA|CC|b$$
$$A \rightarrow a|bC|abA|CC|b$$
$$B \rightarrow b$$
$$C \rightarrow abA|CC|b$$

Exercise

Exercise. Eliminate all unit-productions from

$$S \to a|aA|B|C$$
$$A \to aB|\lambda$$
$$B \to Aa$$
$$C \to cCD$$
$$D \to ddd|Cd$$

Removing useless variables

Definition. Given a grammar $G = (V, T, S, P)$, a variable $A \in V$ is **useful** if it appears in derivation of at least one string w in $L(G)$. Otherwise, the variable is **useless** (either unreachable or non-generating).

Definition. A production $x \to y$ is **useless** if it contains any useless variable.

Theorem. Removing useless productions from G produces an equivalent grammar G'.

Example

Example. Suppose the grammar G has the following productions:

$$S \to A|C$$
$$A \to aA|\lambda$$
$$B \to bA$$
$$C \to aCA$$

Notice that B (unreachable) and C (non-generating) are useless variables, so we can remove all productions containing them and get the following grammar

$$S \to A$$
$$A \to aA|\lambda$$

Exercise

Exercise. Eliminate all useless productions from

$$S \to a|aA|B|C$$
$$A \to aB|\lambda$$
$$B \to Aa$$
$$C \to cCD$$
$$D \to ddd|Cd$$

Grammar simplification

To clean up a grammar follow the steps shown above in this order:

1. Remove nullable variables and λ-productions
2. Remove unit productions
3. Remove useless productions

Exercise

Exercise. Simplify the following grammar
$G = (\{S, A, B, C, D\}, \{a, b\}, S, P)$ with P

$$S \to aaA|aBB|DaB$$
$$A \to AaA|\lambda$$
$$B \to bB|D$$
$$C \to B$$
$$D \to \lambda|bDa$$

Normal forms

Chomsky normal form

Definition. A context-free grammar $G = (V, T, S, P)$ is in **chomsky normal form** if all of its productions are of the from

$$A \to BC \qquad\qquad \text{OR}$$
$$A \to a$$

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

Example. The following grammar $G = (V, T, S, P)$ is NOT in CNF

$$S \to BC|A$$
$$A \to aa|aAAB$$

Conversion to CNF

Theorem. Any CFG G (where $\lambda \notin L(G)$) can be converted to CNF.
To convert a grammar $G = (V, T, S, P)$ that is not in CNF to an equivalent grammar $G' = (V', T', S', P')$ in CNF:

1. Simplify G. Add production rules to P'.
2. For every terminal $x \in T$, create a rule $T_x \to x$. Add to P'. If the only problem is having productions with more than 2 variables on the right, skip this step.
3. For any production $A \to C_1 C_2 \ldots C_n$ convert it to the following $n - 1$ production rules and add them to P'.

$$A \to C_1 V_1$$
$$V_1 \to C_2 V_2$$
$$\ldots$$
$$V_{n-2} \to C_{n-1} C_n$$

Exercise

Exercise. Convert the following grammar to CNF

$$S \to bS|Aa|A|C$$
$$A \to aab|\lambda$$
$$B \to AcBB$$

Greibach normal form

Definition. A context-free grammar $G = (V, T, S, P)$ is said to be in Greibach normal form if all productions have the form

$$A \rightarrow ax$$

where $a \in T, x \in V^*$.

Example. The following grammar is NOT in GNF

$$S \rightarrow abSbS|aaS|a$$

Theorem. Every CFG G ($\lambda \notin L(G)$) can be converted into an equivalent grammar G' in GNF.