equivalent grammars

Canonical name	EquivalentGrammars
Date of creation	2013-03-22 17:37:17
Last modified on	2013-03-22 17:37:17
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	6
Author	CWoo (3771)
Entry type	Definition
Classification	msc 91F20
Classification	msc 68Q42
Classification	msc 68Q45
Classification	msc 03D05

Two formal grammars $G_1 = (\Sigma_1, N_1, P_1, \sigma_1)$ and $G_2 = (\Sigma_2, N_2, P_2, \sigma_2)$ are said to be *equivalent* if they generate the same formal languages:

$$L(G_1) = L(G_2).$$

When two grammars G_1 and G_2 are equivalent, we write $G_1 \cong G_2$.

For example, the language $\{a^n b^{2n} \mid n \text{ is a non-negative integer}\}$ over $T = \{a, b, c\}$ can be generated by a grammar G_1 with three non-terminals x, y, σ, and the following productions:

$$P_1 = \{\sigma \to \lambda, \sigma \to x\sigma y, x \to a, y \to b^2\}.$$

Alternatively, it can be generated by a simpler grammar G_2 with just the starting symbol:

$$P_2 = \{\sigma \to \lambda, \sigma \to a\sigma b^2\}.$$

This shows that $G_1 \cong G_2$.

An alternative way of writing a grammar G is (T, N, P, σ), where $T = \Sigma - N$ is the set of terminals of G. Suppose $G_1 = (T_1, N_1, P_1, \sigma_1)$ and $G_2 = (T_2, N_2, P_2, \sigma_2)$ are two grammars. Below are some basic properties of equivalence of grammars:

1. Suppose $G_1 \cong G_2$. If $T_1 \cap T_2 = \varnothing$, then $L(G_1) = \varnothing$.

2. Let $G = (T, N, P, \sigma)$ be a grammar. Then the grammar $G' = (T, N', P, \sigma)$ where $N \subseteq N'$ is equivalent to G.

3. If $(T_1, N_1, P_1, \sigma_1) \cong (T_1, N_1, P_2, \sigma_2)$, then $(T, N, P_1, \sigma_1) \cong (T, N, P_2, \sigma_2)$, where $T = T_1 \cup T_2$ and $N = N_1 \cup N_2$.

4. If we fix an alphabet Σ, then $\cong$ is an equivalence relation on grammars over Σ.

So far, the properties have all been focused on the underlying alphabets. More interesting properties on equivalent grammars center on productions. Suppose $G = (\Sigma, N, P, \sigma)$ is a grammar.

1. A production is said to be trivial if it has the form $v \to v$. Then the grammar G' obtained by adding trivial productions to P is equivalent to G.

2. If $u \in L(G)$, then adding the production $\sigma \to u$ to P produces a grammar equivalent to G.

3. Call a production a *filler* if it has the form $\lambda \to v$. Replacing a filler $\lambda \to v$ by productions $a \to va$ and $a \to av$ using all symbols $a \in \Sigma$ gives us a grammar that is equivalent to G.

4. G is equivalent to a grammar G' without any fillers. This can be done by replacing each filler using the productions mentioned in the previous section. Finally, if $\lambda \in L(G)$, we add the production $\sigma \to \lambda$ to P if it were not already in P.

5. Let $S(P)$ be the set of all symbols that occur in the productions in P (in either antecedents or conseqents). If $a \in S(P)$, then G' obtained by replacing every production $u \to v \in P$ with production $u[a/b] \to v[a/b]$ and $b \to a$, with a symbol $X \notin \Sigma$, is equivalent to G. Here, $u[a/b]$ means that we replace each occurrence of a in u with X.

6. G is equivalent to a grammar G', all of whose productions have antecedents in N^+ (non-empty words over N).

Remark. In fact, if we let

1. $X \to XY$,

2. $XY \to ZT$,

3. $X \to \lambda$,

4. $X \to a$

be four types of productions, then it can be shown that

- any formal grammar is equivalent to a grammar all of whose productions are in any of the four forms above

- any context-sensitive grammar is equivalent to a grammar all of whose productions are in any of forms 1, 2, or 4, and

- and any context-free grammar is equivalent to a grammar all of whose productions are in any of forms 1, 3, or 4.

References

[1] A. Salomaa *Computation and Automata, Encyclopedia of Mathematics and Its Applications, Vol. 25*. Cambridge (1985).