derivation language

Canonical name	DerivationLanguage
Date of creation	2013-03-22 18:58:15
Last modified on	2013-03-22 18:58:15
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	13
Author	CWoo (3771)
Entry type	Definition
Classification	msc 68Q42
Classification	msc 68Q45
Synonym	Szilard language

Background

Let $G = (\Sigma, N, P, \sigma)$ be a formal grammar. A pair (W_1, W_2) of words over Σ is said to correspond to the production $p \to q$ if

$$W_1 = u_1 p v_1 \qquad \text{and} \qquad W_2 = u_2 q v_2$$

for some words u_i, v_j over Σ. We also say that (W_1, W_2) is a derivation step, and write $W_1 \to W_2$.

Recall that a derivation in G is a finite sequence of words

$$W_1, W_2, \ldots, W_n$$

over Σ such that $W_i \Rightarrow W_{i+1}$, for $i = 1, \ldots, n-1$. The derivation is also written

$$W_1 \Rightarrow W_2 \Rightarrow \cdots \Rightarrow W_n.$$

The derivation above has $n-1$ steps. Zero-step derivations are also permitted. These are just words over Σ.

The reflexive transitive closure of $\Rightarrow$ is $\Rightarrow^*$. Thus, $V \Rightarrow^* W$ means that there is a derivation starting with V and ending with W. There may be more than one derivation from V to W.

If we consider each production as a "letter" in the alphabet P, then the above derivation can be represented by a "word" over P in the following manner: the "word" is formed by taking concatenations of the "letters", where concatenations correspond to successive applications of productions in P:

$$[p_1 \to q_1][p_2 \to q_2] \cdots [p_{n-1} \to q_{n-1}].$$

Derivation language is thus a certain collection of derivation words, formally defined below.

Definitions

Treating productions as "letters" is really nothing more than labeling each production with some symbol. Formally, call a *labeling* of an alphabet P a surjection $f : F \to P$, where F is some alphabet. For any $p \in P$, a label for p is an element $x \in F$ such that $f(x) = p$. We will only be interested in injective labeling (hence a bijection) from now on.

Definition. Suppose we are given a labeling f of the set P of productions in the grammar G. Given a derivation $D : W_1 \Rightarrow W_2 \Rightarrow \ldots \Rightarrow W_n$, a *derivation word* U for D is defined inductively as follows:

1. if $n = 1$, then the empty word $U := \lambda$,
2. if $n = 2$, then $U := x \in F$ is a label for a production that $W_1 \Rightarrow W_2$ corresponds to,
3. if $n > 2$, then $U := U_1 U_2$, where
   - U_1 is a derivation word for the derivation $W_1 \Rightarrow \cdots \Rightarrow W_i$,
   - U_2 is a derivation word for the derivation $W_i \Rightarrow \cdots \Rightarrow W_n$.

If U is a derivation word for derivation D, let us abbreviate this by writing $f[U] = D$. Note that we are not applying the labeling f to U, it is merely a notational convenience.

A derivation word is sometimes called a *control word*, for it defines or controls whether and how a word may be derived from another word. Note that any $W_1 \Rightarrow^* W_2$ may correspond to several distinct derivations, and hence several distinct derivation words. Also, the same derivation word may correspond to distinct derivations.

For example, let G be a grammar over two symbols (and) with productions $\sigma \to \lambda$, $\sigma \to (\sigma)$, and $\sigma \to \sigma\sigma$ (G generates the parenthesis language $\mathbf{Paren_1}$) Label the productions as a, b, c respectively. Then the derivation $\sigma \Rightarrow^* (()())$ correspond to derivation words *bcbbaa* and *bcbaba*. Notice that $\sigma\sigma \Rightarrow (\sigma)\sigma$ and $\sigma\sigma \Rightarrow \sigma(\sigma)$ both correspond to the derivation word b.

Definition. The *derivation language* of a grammar $G = (\Sigma, N, P, \sigma)$ is the set of all derivation words for derivations on words generated by G. In other words, consider the labeling $f : F \to P$. The derivation language of G is the set

$$\{U \in F^* \mid f[U] \text{ is a derivation of the form } \sigma \Rightarrow^* u \text{ for some } u \in N^*\}.$$

The derivation language of G is also called the *Szilard language* of G, named after its inventor, and is denoted by $\mathrm{Sz}(G)$.

For example, let G be the grammar over a the letter a, with productions given by $\sigma \to \sigma$, $\sigma \to a$. If the productions are labeled b, c, then $\mathrm{Sz}(G) = \{b^n c \mid n \geq 0\}$. Note that $L(G) = \{a\}$. Likewise, if G' is the grammar over a, with productions $\sigma \to A\sigma$, $A \to \lambda$, and $\sigma \to a$, labeled p, q, r respectively, then $L(G') = \{a\}$. However, $\mathrm{Sz}(G')$ is quite different from $\mathrm{Sz}(G)$:

$$\begin{aligned}\mathrm{Sz}(G') = \{u \in F^* \mid \; & u = vrw, \text{ where } v \in \{p,q\}^*, w \in \{q\}^*, \\ & \#_u(p) = \#_u(q) \text{ and } \#_x(p) \geq \#_x(q) \text{ for all } x \leq u\}\end{aligned}$$

where

- $F = \{p, q, r\}$,
- $\#_u(r)$ means the number of occurrences of r in word u,
- $v \leq u$ means that v is a prefix of u.

Remarks. Let G be a formal grammar.

- Some properties of $\mathrm{Sz}(G)$:
  1. $\mathrm{Sz}(G)$ is always context-sensitive.
  2. If G is regular, so is $\mathrm{Sz}(G)$.
  3. if G is context-free, $\mathrm{Sz}(G)$ may not be; in fact, for any context-free language L, there is a context-free grammar G such that $L = L(G)$ but $\mathrm{Sz}(G)$ is not context-free.
  4. There exists a context-free language L such that $\mathrm{Sz}(G)$ is not context-free for any grammar G generating L.
- However, if one considers the subset $\mathrm{Sz}_L(G)$ of $\mathrm{Sz}(G)$ consisting of all derivation words corresponding to leftmost derivations, then $\mathrm{Sz}_L(G)$ is context-free.
- It is an open question that, given any (context-sensitive) language L, whether there is a grammar G such that $L = \mathrm{Sz}(G)$.

References

[1] A. Salomaa, *Formal Languages*, Academic Press, New York (1973).

[2] G. E. Révész, *Introduction to Formal Languages*, Dover Publications (1991).