Derivation Languages and Syntactical Categories

JOHNSON M. HART

Department of Computer Science, University of Kentucky, Lexington, Kentucky 40506

Derivations in unrestricted phrase structure grammars are represented in terms of a string called a derivation word. The derivation word gives a very compact representation of the canonical (leftmost) derivation. An algebra of derivation words is developed to give a concrete realization to the categorical treatment of derivations due to Hotz. In particular, derivation composition and juxtaposition are defined for derivation words, and effective techniques are given for obtaining the domain and codomain functions. All of the algorithms can be performed in linear time and space.

1. INTRODUCTION

Several authors, such as Griffiths (1968), Hotz (1966), and Loeckx (1970), have investigated the nature and structure of derivations from general phrase structure grammars. Various points of view and notations have been used, but the categorical treatment of Hotz appears to be the most successful to this author. [See Schnorr (1969) for a summary of Hotz's results.]

Hotz regards derivations as "morphisms" from one string to another, obtained by free use of composition and juxtaposition of derivations. "Canonical derivations" are introduced to represent an equivalence class of derivations which are not essentially distinct.

It is now natural to inquire as to how one can represent the canonical derivation in a compact way without resorting to the notation of Griffiths or explicitly writing out the derivation in terms of composition and juxtaposition. Furthermore, such a representation should allow for efficient algorithmic processing of derivations so that one can obtain the composition and juxtaposition of derivations, the domain and codomain functions, and the canonical derivation.

This goal is realized through the use of the "derivation languages" of the author (Hart, 1975). Each "derivation word" in the derivation language represents the canonical derivation and allows for direct algorithmic processing, as will be shown in the subsequent sections. Linear bounds on computation time and space are given.
That the derivation language treatment is equivalent to the other treatments of derivations is obvious, but it is important to establish efficient algorithmic techniques for representing and processing derivations, as is done here. It should also be noted that the results reduce to well-known results in the special case of context-free grammars. Extensive examples are given to illustrate the developments.

2. Representation of Phrase Structure Derivations

A phrase structure grammar (PSG) is a system $G = (V, \Sigma, P, S)$ where $V$ is a finite set of symbols called the alphabet or vocabulary of the grammar, $\Sigma \subseteq V$ is the set of terminal symbols, $S \in V - \Sigma$ is the start symbol, and $P$ is a finite set of production rules of the form

$$\pi: v \rightarrow w,$$

where $w, v \in V^*$. $\pi$ is the name of the production rule.

If $uvw \in V^+$ and $\pi: v \rightarrow w \in P$, we write $uvw \Rightarrow_G uwx$ to show that the second string can be derived from the first. The $G$ subscript, denoting the grammar, is omitted if no ambiguity can result. $\Rightarrow$ is a relation between strings, and $\ast$ is used to denote the reflexive transitive closure of this relation. Then, the phrase structure language (PSL) generated by $G$, denoted $L(G)$, is

$$L(G) = \{w \in \Sigma^* | S \Rightarrow w\}.$$

These definitions and the notation, or variants thereof, are standard in the literature. For instance, see Hopcroft and Ullman (1969) for details.

If $x \Rightarrow y$ in some PSG, the representation of the derivation by a sequence of words such that

$$x = x_1 \Rightarrow x_2 \Rightarrow x_3 \Rightarrow \cdots \Rightarrow x_n = y$$

is ambiguous since the actual productions used and the location in the string of their application are not specified. Also, this description of a derivation distinguishes between derivations which are not essentially different. Griffiths (1968) calls two derivations similar if one can be obtained from the other by a trivial rearrangement of the sequence of words and productions. This same notion is used by Hotz (1966) [see also Schnorr (1969)], and the idea of a canonical derivation naturally arises.
In this section, we show a very compact representation of derivations which generalizes the derivation languages developed by the author (Hart, 1975). The representation of derivations will be by means of a linear representation of the syntactical graphs of Loeckx (1970), and this representation of a derivation will be by means of a derivation word. Every derivation in a Griffiths or Hotz similarity class has exactly the same derivation word, and the canonical form of the derivation, in Hotz's notation, follows directly. Certain algebraic compositions of derivations as defined by Hotz are also defined for derivation words.

If $G = (V, \Sigma, P, S)$, assume $P = \{\pi_1, \pi_2, \ldots, \pi_k\}$ is the set of production rule names. If $\pi_i \in P$ with $\pi_i: a_1a_2 \cdots a_m \rightarrow b_1b_2 \cdots b_n$ for some $m \geq 1$, $n \geq 0$ (the $a$'s and $b$'s are in $V$), we say that the head stratification of $\pi_i$, $H(\pi_i)$, is $m$, and the tail stratification of $\pi_i$, $T(\pi_i)$, is $n$. For each $a \in V$, set $H(a) = T(a) = 1$, so that the alphabet $V \cup P$ becomes a doubly stratified alphabet in the sense of Gorn (1962). Some further definitions are needed.

**Definition 2.1.** Let $G = (V, \Sigma, P, S)$ be a PSG. The head sum of $x \in (V \cup P)^*$, $S_h(x)$, is defined as:

1. $S_h(\varepsilon) = 0$,
2. $S_h(ax) = S_h(x) - 1$ if $a \in V$ and $S_h(x) > 0$,
3. $S_h(ax) = S_h(x) + H(a)$ if $a \in P$, and
4. $S_h(x)$ is undefined in all other cases.

The tail sum of $x \in (V \cup P)^*$, $S_t(x)$, is defined as:

1. $S_t(\varepsilon) = 0$,
2. $S_t(xa) = S_t(x) - 1$ if $a \in V$ and $S_t(x) > 0$,
3. $S_t(xa) = S_t(x) + T(a)$ if $a \in P$, and
4. $S_t(x)$ is undefined in all other cases.

Note that if $S_h$ and $S_t$ are defined, they are nonnegative. If $S_h(x) = 0$, then $x = ax'\pi$ for some $a \in V$, $x' \in (V \cup P)^*$, and $\pi \in P$, or else $x = \varepsilon$. If $S_t(x) = 0$ and $x \neq \varepsilon$, then either $x = \pi x'a$ as above or $x = x'\pi$ for some $\pi \in P$ with $T(\pi) = 0$ and $x'$ with $S_t(x') = 0$.

**Definition 2.2.** Let $G = (V, \Sigma, P, S)$ be a PSG. A string $\alpha \in (V \cup P)^*$ is said to have domain $a_1 \cdots a_m$, written $D_0(\alpha) = a_1 \cdots a_m$, if and only if $\alpha$ can be written as

$$\alpha = a_1\psi_1a_2\psi_2 \cdots \psi_{m-1}a_m\psi_m,$$
with $a_1 \cdots a_m \in V$ and $S_t(\psi_1) = S_t(\psi_2) = \cdots = S_t(\psi_m) = 0$. $\alpha$ has codomain $b_1 \cdots b_n$, written $D_1(\alpha) = b_1 \cdots b_n$, if $\alpha$ can be written as

$$\alpha = \xi_0 b_1 \xi_1 b_2 \cdots \xi_{n-1} b_n,$$

with $b_1,\ldots, b_n \in V$ and $S_h(\xi_0) = S_h(\xi_1) = \cdots S_h(\xi_{n-1}) = 0$.

The definition of $D_0$ and $D_1$ is precise, for a word $\alpha$ has at most one such factorization. Also, if $D_0(\alpha) = x$ and $D_0(\beta) = y$, then $D_0(\alpha\beta) = xy$. Likewise, if $D_1(\alpha) = x$ and $D_1(\beta) = y$, then $D_1(\alpha\beta) = xy$. Thus, $D_0$ and $D_1$ are homomorphisms where they are defined.

**Definition 2.3.** Let $G = (V, \Sigma, P, S)$ be a PSG. A derivation word with domain $x \in V^*$ and codomain $y \in V^*$ is defined recursively as follows.

1. If $\alpha = a_1 a_2 \cdots a_m \in V^*$ ($m \geq 0$), then $a_1 \cdots a_m$ is a derivation word with domain and codomain $\alpha = a_1 \cdots a_m$ ($D_0(\alpha) = D_1(\alpha) = \alpha$).

2. If $\alpha \in (V \cup P)^*$ is a derivation word with $\alpha = \alpha_1 \alpha_2 \alpha_3$, $D_1(\alpha_1) = u$, $D_1(\alpha_2) = a_1 \cdots a_m$, $D_1(\alpha_3) = v$, and if $\pi : a_1 \cdots a_m \to b_1 \cdots b_n \in P$, then $\beta = a_1 \alpha_2 \pi b_1 \cdots b_n$ is a derivation word with $D_0(\beta) = x$ and $D_1(\beta) = ub_1 \cdots b_nv$.

3. Nothing else is a derivation word unless its being so follows from (1) and (2).

This definition is precise, and it is easy to check (using recursion) that the derivation words have the indicated domain and codomain as defined by Definition 2.2.

Denote by $[x, y]$ or $[x, y]_G$ the set of all derivation words $\alpha$ such that $D_0(\alpha) = x$ and $D_1(\alpha) = y$. In this case, we also say that $\alpha$ is a derivation word from $x$ to $y$.

If $x = x_1 \Rightarrow_G x_2 \Rightarrow_G \cdots \Rightarrow_G x_n = y$ for some PSG $G$ then we can generate a corresponding derivation word in $[x, y]_G$ by repeatedly applying Step 2 of Definition 2.3. It will turn out that the actual sequence of derivation steps can be recovered from the derivation word, except that the sequence obtained may differ from the original in inessential differences in the order in the sequence. In fact, the derivation word will represent an entire equivalence class of derivations [in the sense of Griffiths (1968) or Hotz (1966)].

Hotz observes that strings in $V^*$ can be regarded as the objects of a category and that $[x, y]$ is the set of morphisms from object $x$ to object $y$. We will not make use of category theory in what follows, except to use the above observation to justify the use of the words "domain" and "codomain".
Example 2.1. Let \( G = (V, \Sigma, P, S) \) be a PSG with \( V = \{A, B, C\} \) and
\[ P = \{ \pi_1: AB \rightarrow CAB, \pi_2: C \rightarrow BA, \pi_3: BAA \rightarrow CBA \} . \]
\( S \) and \( \Sigma \) are not relevant to the example. Then
\[
\begin{align*}
AB & \in [AB, AB], \\
AB\pi_1CAB & \in [AB, CAB], \\
AB\pi_1CAB\pi_1CAB & \in [AB, CCAB], \\
\alpha_1 = AB\pi_1CAB\pi_1C\pi_2BAAAB & \in [AB, CBAAB].
\end{align*}
\]
With \( \psi_1 = \epsilon \) and \( \psi_2 = \pi_1CAB\pi_1C\pi_2BAAAB \), we have \( S_\epsilon(\psi_1) = S_\epsilon(\psi_2) = 0 \) so that \( D_0(\alpha_1) = AB \) as claimed. With \( \xi_0 = AB\pi_1, \xi_1 = AB\pi_1C\pi_2, \) and \( \xi_2 = \xi_3 = \xi_4 = \epsilon \), we have \( S_\xi(\xi_i) = 0 \) and \( \alpha_1 = \xi_0C\xi_1B\xi_2A\xi_3A\xi_4A \). Thus, \( D_1(\alpha_1) = CBAAB \).

Example 2.2. Let \( G \) be as in the previous example. Then
\[
\begin{align*}
CBAAB & \in [CBAAB, CBAAB], \\
C\pi_2BABAAB & \in [CBAAB, BABABAAB], \\
C\pi_2BAB\pi_1CABAAB & \in [CBAAB, BCABABAAB], \\
\alpha_2 = C\pi_2BAB\pi_1CABAAB\pi_3CBAB & \in [CBAAB, BCACBABA].
\end{align*}
\]
To show the domain and codomain of \( \alpha_2 \), we parenthesize the \( \psi \) and \( \xi \) strings so as to show the two functions.
\[
\begin{align*}
\alpha_2 = C(\pi_2BA)B(\pi_1CAB)A( )A(\pi_3CBA)B( ); D_0(\alpha_2) = CBAAB; \\
\alpha_2 = (C\pi_2)B(AB\pi_1)C( )A(BAA\pi_3)C( )B( )A( )B; D_1(\alpha_2) = BCACBABA.
\end{align*}
\]
Note 1. In both of the previous examples a similar analysis yields the domains and codomains of the derivation words.

Note 2. If \( G \) is a context-free grammar, a derivation word of \( G \) becomes the familiar prefix representation of a derivation tree (with the addition of the production names).

Note 3. If \( G = (V, \Sigma, P, S) \) is a PSG, then \( L(G) = \{ y \in \Sigma^* \mid \exists \alpha \in [S, y]_G \} \).

In Note 2, the relationship between context-free derivation trees and derivation words is pointed out. In the general case, we can relate derivation words to the so-called syntactical graphs of Loeckx (1970). Figure 1 shows the syntactical graphs of derivation words \( \alpha_1 \) and \( \alpha_2 \) (in the preceding examples).
Fig. 1. Syntactical graphs of the derivation words in Examples 2.1 and 2.2. (a) Syntactical graph of the derivation word $\alpha_1 = AB\pi_1 CAB\pi_1 C\pi_2 BAAB$. (b) Syntactical graph of the derivation word $\alpha_2 = C\pi_2 BAB\pi_1 CABAA\pi_2 CBAB$.

The graphs are self-explanatory. The derivation words can be obtained by following the paths to the left as far as possible (reading off the node labels) but not entering a node labeled by a production name until all of the nodes above it have been listed. This technique is formally established by the author in a previous paper (Hart, 1973a).

If $\alpha \in (V \cup P)^*$, a necessary, but not sufficient, condition that $\alpha$ be a derivation word in $[x, y]$ is that $D_0(\alpha) = x$ and $D_1(\alpha) = y$. The lemma that follows gives a necessary and sufficient condition that $\alpha$ be in $[x, y]$ for some words $x$ and $y$. A definition is necessary first.

**Definition 2.4.** Let $G = (V, \Sigma, P, S)$ be a PSG. If a word $\alpha \in (V \cup P)^*$ can be written as $\alpha = \alpha_1 \alpha_2 \pi b_1 \cdots b_n \alpha_3$ such that $\alpha_3 \in V^*$, $D_1(\alpha_3) = a_1 \cdots a_m$, and $\pi \in P$ is the rule $\pi: a_1 \cdots a_m \rightarrow b_1 \cdots b_n$, then $\alpha$ right reduces to $\alpha_1 \alpha_2 \alpha_3$. We write $\alpha \rightarrow_R \alpha_1 \alpha_2 \alpha_3$.

Right reduction is performed by finding the right-most occurrence of a production name in the word. If the production name can be made to correspond with symbols on the left and right, the reduction can take place. Note that if $\alpha \in [D_0(\alpha), D_1(\alpha)]$ and $\alpha \rightarrow_R \beta$, then $\beta \in [D_0(\beta), D_1(\beta)]$ with $D_0(\beta) = D_0(\alpha)$. Also, $D_1(\beta) \rightarrow_G D_1(\alpha)$. If $\alpha \rightarrow_R \beta_1$ and $\alpha \rightarrow_R \beta_2$, then $\beta_1 = \beta_2$.
Lemma 2.1. Let $G = (V, \Sigma, P, S)$ be a PSG. Then $\alpha \in [x, y]_G$ if and only if $D_0(\alpha) = x$, $D_1(\alpha) = y$, and there exists a sequence of derivation words $\alpha_1, \alpha_2, \ldots, \alpha_p$ ($p \geq 1$) such that $\alpha = \alpha_1 \vdash_R \alpha_2 \vdash_R \cdots \vdash_R \alpha_p = x$.

The lemma is proved by showing that right reduction is the inverse of the operation used to create derivation words. A proof of a similar (but more limited) lemma is given in Hart (1974), but we have given the lemma here for later use in Section 3.

Example 2.3. Use the derivation words $\alpha_1$ and $\alpha_2$ from the previous examples. We show the reduction of $\alpha_1$ and $\alpha_2$. At each point, the $a_i$, $\pi$, and $b_i$ are underlined.

$\alpha_1 = AB\pi_1CAB\pi_2C \pi_2BA AB$

$\vdash_R AB\pi_1C AB \pi_1CAB$

$\vdash_R AB \pi_1CAB$

$\vdash_R AB$

$\alpha_2 = C\pi_2BAB\pi_1CA BAA \pi_3CBA B$

$\vdash_R C\pi_2B AB \pi_1CAB AAB$

$\vdash_R C \pi_3BAB AAB$

$\vdash_R CBAAB.$

Derivation words as defined here are strings of characters over the alphabet $V \cup P$. These derivation words correspond to two-dimensional structures, the syntactical graphs. In order to obtain the graphical interpretation of a derivation, it is necessary to use the syntactical stratification of the alphabet and perform successive right reductions of the word until no production names remain. Derivation words also allow easy extraction of the derived word and the original word. The language over $V \cup P$ of derivation words is recursive and very easy to parse.

3. OPERATIONS ON DERIVATIONS AND CANONICAL DERIVATIONS

Hotz (1966) (see also Schnorr, 1969) considers two operations which combine derivations, juxtaposition, and composition. We consider these operations as formulated in terms of the derivation words introduced in the preceding section.
The first operation is that of juxtaposition of derivations. That is, if $x_1 \overset{x}{\Rightarrow}_G y_1$ and $x_2 \overset{y}{\Rightarrow}_G y_2$ are two derivations, then there is a derivation $x_1 x_2 \overset{x}{\Rightarrow} y_1 y_2$ obtained by using $x_1 x_2 \overset{y}{\Rightarrow} y_1 y_2$ or $x_1 x_2 \overset{y}{\Rightarrow} y_1 y_2$. That is, the two derivations are performed independently, side by side.

**Definition 3.1.** Let $G = (V, \Sigma, P, S)$ be a PSG with $\alpha_1 \in [x_1, y_1]$ and $\alpha_2 \in [x_2, y_2]$. Then the juxtaposition of $\alpha_1$ with $\alpha_2$, denoted by $\alpha_1 \times \alpha_2$, is
\[ \alpha_1 \times \alpha_2 = \alpha_1 \alpha_2, \]
where $\alpha_1 \alpha_2$ is the product of $\alpha_1$ and $\alpha_2$ in the free monoid $(V \cup P)^*$.

**Lemma 3.1.** If $\alpha_1 \in [x_1, y_1]$ and $\alpha_2 \in [x_2, y_2]$ for some PSG, $G$ then $\alpha_1 \times \alpha_2 \in [x_1 x_2, y_1 y_2]$.

**Proof.** First note that $D_0(\alpha_1) D_0(\alpha_2) = D_0(\alpha_1 \times \alpha_2) = x_1 x_2$ and that $D_1(\alpha_1) D_1(\alpha_2) = D_1(\alpha_1 \times \alpha_2) = y_1 y_2$.

There also exist sequences of derivation words $\beta_1, \beta_2, \ldots, \beta_m$ and $\gamma_1, \gamma_2, \ldots, \gamma_n$ such that
\[ \alpha_1 = \beta_1 \overset{R}{\Rightarrow} \beta_2 \overset{R}{\Rightarrow} \cdots \overset{R}{\Rightarrow} \beta_m = x_1 \]
and
\[ \alpha_2 = \gamma_1 \overset{R}{\Rightarrow} \gamma_2 \overset{R}{\Rightarrow} \cdots \overset{R}{\Rightarrow} \gamma_n = x_2. \]

Consequently,
\[ \alpha_1 \times \alpha_2 = \alpha_1 \alpha_2 = \beta_1 \gamma_1 \overset{R}{\Rightarrow} \beta_2 \gamma_2 \overset{R}{\Rightarrow} \cdots \overset{R}{\Rightarrow} \beta_m \gamma_n = x_1 x_2. \]

From Lemma 2.1, we have $\alpha_1 \times \alpha_2 \in [x_1 x_2, y_1 y_2]_G$. Q.E.D.

Next, consider composition of derivations, where if $x \overset{z}{\Rightarrow}_G y$ and $y \overset{z}{\Rightarrow}_G z$, we compose the two derivations to obtain $x \overset{z}{\Rightarrow}_G y \overset{z}{\Rightarrow}_G z$, or $x \overset{z}{\Rightarrow}_G z$. The correct way to obtain this derivation in terms of derivation words is given in the next definition.

**Definition 3.2.** Let $G = (V, \Sigma, P, S)$ be a PSG with $\alpha_1 \in [x, y]_G$ and $\alpha_2 \in [y, z]_G$. If $y = b_1 \cdots b_n (b_i \in V)$ with
\[ \alpha_1 = \xi_0 b_1 \xi_2 \cdots \xi_{n-1} b_n \quad (S_n(\xi_i) = 0, 0 \leq i \leq n - 1) \]
and
\[ \alpha_2 = b_1 \psi_1 b_2 \psi_2 \cdots b_n \psi_n \quad (S_i(\psi_i) = 0, 1 \leq i \leq n), \]
then the composition of $\alpha_2$ with $\alpha_1$, written $\alpha_2 \circ \alpha_1$, is

$$\alpha_2 \circ \alpha_1 = \xi_0 b_1 \xi_1 b_2 \xi_2 ... \xi_{n-1} b_n \psi_n.$$ 

This definition must now be shown to be the correct one, in terms of giving a derivation word which is the result of composing the two derivations. Two preliminary results are needed first.

**Lemma 3.2a.** If $\alpha_1 \in [x, y]_G$ and $\alpha_2 \in [y, z]_G$ for some PSG, $G$ then

$$D_0(\alpha_2 \circ \alpha_1) = x \quad \text{and} \quad D_1(\alpha_2 \circ \alpha_1) = z.$$ 

**Proof.** First, let $x = a_1 ... a_m$, $y = b_1 ... b_n$, and $z = c_1 ... c_p$ where the $a's$, $b's$, and $c's$ are in $V$ and $m$, $n$, $p \geq 0$. Hence, we can write:

$$\alpha_1 = a_1 \psi_1 a_2 \psi_2 ... a_m \psi_m = \xi_0 b_1 \xi_1 b_2 ... \xi_{n-1} b_n,$$

$$\alpha_2 = b_1 \psi_1 b_2 \psi_2 ... \psi_{n-1} b_n \psi_n = \xi_0 c_1 \xi_1 c_2 ... \xi_{p-1} c_p,$$

where, as usual, $S_t(\psi_i) = 0$ ($1 \leq i \leq m$), $S_h(\xi_i) = S_t(\psi_i) = 0$ ($0 \leq i \leq n$), and $S_h(\xi_i') = 0$ ($0 \leq i \leq p - 1$). Now, by definition

$$\alpha_2 \circ \alpha_1 = \xi_0 b_1 \xi_1 b_2 \xi_2 ... \xi_{n-1} b_n \psi_n$$

$$= a_1 \psi_1 a_2 \psi_2 ... a_m \psi_m$$

for some strings $\psi_1', ..., \psi_m' \in (V \cup P)^*$. We wish to show that $S_t(\psi_1') = ... = S_t(\psi_m') = 0$ so that $D_0(\alpha_2 \circ \alpha_1) = a_1 ... a_m = x$.

By the construction of $\alpha_2 \circ \alpha_1$, each $\psi_i'$ is obtained by inserting certain of the $\psi$ into $\psi_i$. That is, there is an integer $n_i \geq 1$, a factorization of $\psi_i'$ as

$$\psi_i' = \psi_{i1} \psi_{i2} ... \psi_{in_i},$$

and integers $j_i$, $k_i$ such that $1 \leq j_i \leq k_i \leq n$ with

$$\psi_i'' = \psi_{i1} \psi_{i2} \psi_{j_{i+1}} ... \psi_{k_i} \psi_{in_i}.$$ 

(If $n_i = 1$, we simply have $\psi_i'' = \psi_{i1}$ with $k_i$, $j_i$ not defined.)

By the hypothesis, $S_t(\psi_i') = S_t(\psi_i) = ... = S_t(\psi_{k_i}) = 0$, and hence $D_0(\alpha_2 \circ \alpha_1) = x$.

A symmetrical argument shows that $D_1(\alpha_2 \circ \alpha_1) = z$. Q.E.D.
**Lemma 3.2b.** If $\alpha_1 \in [x, y]_G$ and $\alpha_2 \in [y, z]_G$ for some PSG, $G = (V, \Sigma, P, S)$, such that $\alpha_1 \vdash R \beta_1$ and $\alpha_2 \vdash R \beta_2$, then either

(a) $\alpha_2 \circ \alpha_1 \vdash R \beta_2 \circ \alpha_1$, or

(b) there is a derivation word $\beta'_2$ such that $\alpha_2 \circ \alpha_1 \vdash R \beta'_2 \circ \beta_1$.

**Proof.** Using the notation of Lemma 3.2a, we have

$$\alpha_2 \circ \alpha_1 = \xi_0 b_1 \xi_1 b_2 \xi_2 \cdots \xi_{n-1} b_n \psi_n \cdot$$

There are two cases to consider, corresponding to the two cases in the statement of the lemma.

**Case a.** For some $i$, $1 \leq i \leq n$, $\psi_i \neq \epsilon$ but $\xi_i = \psi_{i+1} = \cdots = \psi_{n-1} = \psi_n = \epsilon$. Therefore, we can write:

$$\alpha_1 = \xi_0 b_1 \xi_1 b_2 \cdots \xi_{i-1} b_i b_{i+1} \cdots b_n,$$

$$\alpha_2 = b_1 \psi_1 b_2 \psi_2 \cdots b_i \psi_i b_{i+1} \cdots b_n, \quad \text{and}$$

$$\alpha_2 \circ \alpha_1 = \xi_0 b_1 \psi_1 b_2 \xi_2 \cdots b_i \psi_i b_{i+1} \cdots b_n.$$

Since $S_i(\psi_i) = 0$ and $\psi_i \neq \epsilon$, we can write $\psi_i$ uniquely as $\psi_i = \psi_i \pi d_1 \cdots d_q \psi_R$ where $\pi: c_1 \cdots c_g \rightarrow d_1 \cdots d_q \in P$ is some production of $G$ and $\psi_R \in V^*$. Set $\psi'_i = \psi_i \psi_R$ (note that $S_i(\psi'_i) = 0$), and by the definition of right reduction, we must have

$$\alpha_2 \vdash R b_i \psi_i b_{i+1} \cdots b_n = \beta_2 \cdot$$

From this,

$$\beta_2 \circ \alpha_1 = \xi_0 b_1 \psi_1 \xi_2 b_2 \cdots b_i \psi_i b_{i+1} \cdots b_n,$$

where $D_1(\beta_2 \circ \alpha_1) = D_1(\beta_2)$ (from Lemma 3.2a). Consequently, by Definition 2.4,

$$\alpha_2 \circ \alpha_1 \vdash R \beta_2 \circ \alpha_1 \cdot$$

**Case b.** For some $i$ ($0 \leq i \leq n - 1$), $\xi_i \neq \epsilon$ but $\psi_{i+1} = \xi_{i+1} = \cdots = \xi_{n-1} = \psi_n = \epsilon$ so that we can write

$$\alpha_1 = \xi_0 b_1 \xi_1 b_2 \cdots b_i \xi_i b_{i+1} \cdots b_n,$$

$$\alpha_2 = b_1 \psi_1 b_2 \psi_2 \cdots \psi_{i+1} b_{i+1} \cdots b_n = (b_1 \psi_1 b_2 \psi_2 \cdots b_{i+1} \psi_{i+1}) \times (b_{i+1} \cdots b_n), \quad \text{and}$$

$$\alpha_2 \circ \alpha_1 = \xi_0 b_1 \psi_1 \xi_2 b_2 \cdots \psi_{i+1} \xi_{i+1} b_{i+1} \cdots b_n.$$
From the remarks after Definition 2.1, \( \xi_i \) can be written as
\[
\xi_i = \xi'_i \pi
\]
for some \( \pi: c_1 \cdots c_p \rightarrow b_{i+1} \cdots b_j \in P \) (if \( j = i \), then the right-hand side of \( \pi \) is \( \epsilon \)). In addition, \( D_1(\xi_0 b_1 \xi_1 b_2 \cdots b_i \xi'_i) = b_1 b_2 \cdots b_i c_1 \cdots c_p \). We then have
\[
\alpha_1 \dashv \Rightarrow \xi_0 b_1 \xi_1 b_2 \cdots b_i \xi'_i b_{i+1} \cdots b_n = \beta_1, \\
D_1(\beta_1) = b_1 \cdots b_i c_1 \cdots c_p b_{i+1} \cdots b_n, \quad \text{and} \\
D_1(\alpha_1) = b_1 b_2 \cdots b_i b_{i+1} \cdots b_n = y.
\]
Now, set
\[
\beta'_2 = b_1 \xi_1 b_2 \xi_2 \cdots b_i \xi_i c_1 \cdots c_p b_{i+1} \cdots b_n,
\]
and it is apparent that \( \beta'_2 \) is a derivation word since \( \beta'_2 = (b_1 \xi_1 \cdots b_i \xi_i) \times (c_1 \cdots c_p b_{i+1} \cdots b_n) \) and \( D_0(\beta'_2) = D_1(\beta_1) \). As a result, \( \beta'_2 \circ \beta_1 \) is defined and \( \alpha_2 \circ \alpha_1 \vdash \Rightarrow \beta'_2 \circ \beta_1 \).

As a result of Lemma 3.2b, the composition of any two derivation words (each being nontrivial so that they have right reductions) right reduces to the composition of two derivation words. We now get the principal result regarding composition.

**LEMMA 3.2.** If \( \alpha_1 \in [x, y]_G \) and \( \alpha_2 \in [y, z]_G \) for some PSG, \( G \) then \( \alpha_2 \circ \alpha_1 \in [x, z]_G \).

**Proof.** By Lemma 3.2a, we have \( D_0(\alpha_2 \circ \alpha_1) = x \) and \( D_1(\alpha_2 \circ \alpha_1) = z \). If either \( \alpha_1 \in V^* \) or \( \alpha_2 \in V^* \), the rest of the lemma is immediate.

In general, by repeated application of Lemma 3.2b, there are two sequences of derivation words (\( \beta_{11}, \beta_{12}, \ldots, \beta_{1n} \) and \( \beta_{21}, \beta_{22}, \ldots, \beta_{2n} \)) such that \( \beta_{11} = \alpha_1 \), \( \beta_{21} = \alpha_2 \) and \( \alpha_2 \circ \alpha_1 = \beta_{21} \circ \beta_{11} \vdash \Rightarrow \beta_{22} \circ \beta_{12} \vdash \Rightarrow \cdots \vdash \Rightarrow \beta_{2n} \circ \beta_{1n} = x \) with \( D_0(\beta_{2i} \circ \beta_{1i}) = x \) for all \( i, i = 1, \ldots, n \). Lemma 2.1 then gives the desired result. Q.E.D.

**EXAMPLE 3.1.** Consider the derivation words
\[
\alpha_1 = AB\pi_1 CAB\pi_1 C\pi_2 BAAB \in [AB, CBAAB]
\]
and
\[
\alpha_2 = C\pi_2 BAB\pi_1 CABAA\pi_3 CBAB \in [CBAAB, BCACBAB].
\]
Then
\[
\alpha_2 \circ \alpha_1 = AB\pi_1 C\pi_2 BAAB\pi_1 C\pi_2 B\pi_1 CABAA\pi_3 CBAB \in [AB, BCACBAB].
\]
Derivation Languages and Categories

Figure 2 shows the syntactical graph of $\alpha_2 \circ \alpha_1$ as the composition of the two graphs in Fig. 1.

For every word $x \in V^*$, the word $x$ is itself a derivation word in $[x, x]$ of length 0. $x$, regarded as a derivation word, is called $\text{id}_x$ (the identity morphism from object $x$ to object $x$).

We now establish two identities among derivation words and their operations. These identities are given by Hotz (1966) and Schnorr (1969) for derivations, but are stated and proved here in terms of the new definition of derivation words.

**Lemma 3.3.** Let $\alpha \in [x, y]$ be a derivation word for some PSG $G$. Then

$$\alpha \circ \text{id}_x = \text{id}_y \circ \alpha.$$

*Proof.* Obvious from the fact that $\text{id}_x = x$ and $\text{id}_y = y$.

**Lemma 3.4.** If $\alpha_1 \in [x_1, y_1]$, $\beta_1 \in [y_1, x_1]$, $\alpha_2 \in [x_2, y_2]$, and $\beta_2 \in [y_2, x_2]$ for some PSG $G$ then

$$(\beta_1 \circ \alpha_1) \times (\beta_2 \circ \alpha_2) = (\beta_1 \times \beta_2) \circ (\alpha_1 \times \alpha_2).$$

*Proof.* In accordance with Definition 3.2, set

$$\alpha_i = \xi_{i0} b_{i1} \xi_{i2} b_{i3} \ldots \xi_{i,n_i - 1} b_{in_i} \quad (i = 1, 2)$$

and

$$\beta_i = b_{i1} \psi_{i1} b_{i2} \psi_{i2} \ldots b_{in_i} \psi_{in_i} \quad (i = 1, 2),$$
so that the $\xi$ and $\psi$ satisfy the usual head and tail sum conditions. Then

$$(\beta_1 \circ \alpha_1) \times (\beta_2 \circ \alpha_2) = (\xi_{11} b_{11} \xi_{21} b_{12} \psi_{11} b_{13} \xi_{22} \cdots b_{1n_1} \psi_{1n_1})$$

$$\times (\xi_{21} b_{21} \psi_{21} \xi_{22} b_{22} \psi_{22} \cdots b_{2n_2} \psi_{2n_2})$$

$$= (\xi_{11} b_{11} \xi_{21} b_{12} \cdots b_{1n_1} \psi_{1n_1} \xi_{21} b_{21} \psi_{21} \xi_{22} b_{22} \cdots b_{2n_2} \psi_{2n_2})$$

$$= (b_{11} \xi_{11} b_{12} \cdots b_{1n_1} \psi_{1n_1} \xi_{21} b_{21} \psi_{21} \xi_{22} b_{22} \cdots b_{2n_2} \psi_{2n_2})$$

$$\circ (\xi_{11} b_{11} \xi_{12} \cdots b_{1n_1} \xi_{21} b_{21} \xi_{22} \cdots \xi_{2n_2} \psi_{2n_2})$$

$$= (\beta_1 \times \beta_2) \circ (\alpha_1 \times \alpha_2).$$

Q.E.D.

To prevent confusion with the definition of "derivation word" used in this paper, a "derivation sequence" will mean a sequence of words $y_1, y_2, \ldots, y_n$ such that $y_1 \Rightarrow_G y_2 \Rightarrow_G \cdots \Rightarrow_G y_n$ for a PSG $G$. Composition and juxtaposition have obvious meanings when applied to derivation sequences, and Hotz (1966) gives Lemmas 3.3 and 3.4 for derivation sequences. Here, it has been shown that these identities hold for derivation words as well. Hotz then has the following proposition.

**Proposition (Hotz, 1966).** Two derivation sequences are equivalent (in the sense of Griffiths) if and only if one can be transformed into the other by the identities of Lemmas 3.3 and 3.4.

But these two identities leave derivation words unchanged, so a derivation word yields the entire equivalence class of derivation sequences. Consequently, two derivation sequences are equivalent if and only if they yield the same derivation word or syntactical graph.

Every derivation sequence in a grammar $G$ can be written as

$$y_1 \Rightarrow_G y_2 \Rightarrow_G y_3 \Rightarrow_G \cdots \Rightarrow_G y_n$$

such that $y_i = u_i v_i x_i$ and $y_{i+1} = u_i w_i x_i$ for some strings $u_i, v_i, w_i, x_i \in V^*$ ($1 \leq i \leq n$) with $v_i \rightarrow w_i$ a production rule of $P$. The following proposition is found in Griffiths (1968) and Hotz (1966).

**Proposition.** Let $G = (V, \Sigma, P, S)$ be a PSG. Then every derivation sequence

$$y'_1 \Rightarrow_G y'_2 \Rightarrow_G \cdots \Rightarrow_G y'_n$$

is equivalent to a derivation sequence

$$y_1 \Rightarrow_G y_2 \Rightarrow_G \cdots \Rightarrow_G y_n$$
such that \( y_i = u_i v_i x_i, y_{i+1} = u_i w_i x_i \), with \( v_i \to w_i \) a production rule of \( P \) and \[
| u_i | < | u_{i+1} | + | v_{i+1} | \quad (1 \leq i \leq n - 2).
\]

| \( w \) | denotes the length of string \( w \), and \( u_n, v_n, x_n, \) and \( w_n \) are not defined.

If \( n \geq 2 \) and either \( y_1' \) or \( y_n' \) is not \( \epsilon \) (the null string), then this derivation sequence is unique and is called the "canonical derivation sequence" of \( y_1' \to G y_2' \to G \cdots \to G y_n' \) (Note that \( y_1' = y_1 \) and \( y_n' = y_n \)).

In a canonical derivation sequence, the production rule is applied as far to the left in the string as is possible.

We now show that the canonical derivation sequence is obtained directly from the derivation word or syntactical graph. In fact, the canonical derivation sequence is obtained from the sequence of right reductions of the derivation.

**Theorem 3.1.** Let \( G = (V, \Sigma, P, S) \) be a PSG. Let \( \alpha \in [x, y]_G \) be a derivation word with \( \alpha = a_0 \to_R a_{n-1} \to_R \cdots \to_R a_1 = x \) the sequence of right reductions of \( \alpha \). Then \( D_1(a_1) \to_G D_1(a_2) \to_G \cdots \to_G D_1(a_n) \) is a canonical derivation sequence.

**Proof.** For each \( i (1 \leq i \leq n - 1) \), we have \( a_{i+1} \to_R a_i \) such that \( a_{i+1} = \xi_0 a_1 \xi_1 a_2 \cdots \xi_{m-1} a_m \pi b_1 \cdots b_p \xi_m \) where \( \xi_m \in V^* \); \( a_1, \ldots, a_m, b_1, \ldots, b_p \in V \); \( \xi_0 \in (V \cup P)^* \), and \( S_h(\xi_1) = \cdots = S_h(\xi_{m-1}) = 0 \). Also, \( \pi : a_1 \cdots a_m \to b_1 \cdots b_p \in P \). By definition, \( a_i = \xi_0 a_1 \xi_1 a_2 \cdots \xi_{m-1} a_m \xi_m \). Let \( x_i = \xi_m \), \( v_i = a_1 \cdots a_m \), \( w_i = b_1 \cdots b_p \), and \( u_i \) be that string such that \( D_1(a_i) = u_i v_i x_i \). Note that \( D_1(a_{i+1}) = u_i v_i x_i \) and \( D_1(a_i) = u_i v_i x_i \) with \( D_1(a_{i+1}) \to_G D_1(a_i) \). Therefore, the sequence is a derivation sequence, and we must show that it is canonical.

\( D_1(a_i) = u_i v_i x_i \) and \( D_1(a_{i+1}) = u_i w_i x_i = u_{i+1} v_{i+1} x_{i+1} \) for \( 1 \leq i \leq n - 2 \). \( \alpha_{i+2} \) can be written as

\[
\alpha_{i+2} = \xi_0 a_1 \xi_1 a_2 \cdots a_m \pi' b_1 \cdots b_p \xi_m',
\]

with the usual conditions on the \( a', b', \pi' \), and \( \xi' \). In particular, we have \( x_{i+1} = \xi_m', \) and

\[
\alpha_{i+1} = \xi_0 a_1 \xi_1 a_2 \cdots a_m \pi' b_1 \cdots b_p \xi_m' = \xi_0 a_1 \cdots a_m \pi b_1 \cdots b_p \xi_m.
\]

Since \( a_m' \in V \), it is apparent that \( | b_1 \cdots b_p \xi_m | > | \xi_m' | \), or \( | w_i x_i | = | w_i | + | x_i | > | x_{i+1} | \). Since \( u_i w_i x_i = u_{i+1} v_{i+1} x_{i+1} \), we have

\[
| u_i | + | w_i | + | x_i | = | u_{i+1} | + | v_{i+1} | + | x_{i+1} |.
\]
The above inequality then yields \(|u_i| < |u_{i+1}| + |v_{i+1}|\). By definition, the derivation sequence is canonical, as claimed. Q.E.D.

The right-most reduction gives the canonical derivation sequence. This canonical sequence is unique and is the same for every derivation sequence in the equivalence class. The derivation word represents the entire equivalence class of derivation sequences, and this derivation word directly yields the canonical sequence by the right-reduction process.

**Example 3.2.** Consider the derivation word \(\alpha_2 \circ \alpha_1\) of Example 3.1, shown graphically in Fig. 2. In the left column, we show the right reductions, and the right column shows the \(D_f\) function of the left column. The canonical derivation sequence is then obtained by reading the right column from bottom to top. We underline the characters used in the reduction in the left column, and underline the \(u_i\) in the right column.

<table>
<thead>
<tr>
<th>Derivation Word</th>
<th>Codomain (Derived Word) and Canonical Derivation Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AB\pi_1C\pi_2BAAB\pi_1C\pi_2B\pi_1CABA\pi_2CBAB)</td>
<td>BCACBAB</td>
</tr>
<tr>
<td>(\leftarrow R AB\pi_1C\pi_2BAAB\pi_1C\pi_2B\pi_1CABA)</td>
<td>BCABAAB</td>
</tr>
<tr>
<td>(\leftarrow R AB\pi_1C\pi_2BAAB\pi_1C\pi_2BAAB)</td>
<td>BABAAB</td>
</tr>
<tr>
<td>(\leftarrow R AB\pi_1C\pi_2BAAB\pi_1CAB)</td>
<td>BACAB</td>
</tr>
<tr>
<td>(\leftarrow R AB\pi_1C\pi_2BAAB)</td>
<td>BAAB</td>
</tr>
<tr>
<td>(\leftarrow R AB\pi_1CAB)</td>
<td>CAB</td>
</tr>
<tr>
<td>(\leftarrow R AB)</td>
<td>AB</td>
</tr>
</tbody>
</table>
The \( u_i \) and \( v_i \) are also listed so that the inequality for canonical derivation sequences can be verified.

The canonical derivation sequence equivalent to an arbitrary derivation sequence can be found by first creating the derivation word corresponding to the derivation sequence, using Definition 2.3. The canonical derivation sequence is then obtained from the derivation word and its right reductions. Any two equivalent derivation sequences yield the same derivation word and syntactical graph.

The algorithms for the processing of derivation words are directly implied by the definitions. The domain of a derivation word can be computed in a single left-to-right scan of the word, using a single counter. If \( m \) is the maximum tail stratification of any of the production names, and if a derivation word is of length \( n \), then the counter used in computing the domain will certainly never exceed \( mn \). Therefore, the domain function can be computed in linearly bounded space and time. A second right-to-left pass on the derivation word serves to compute the codomain (note the complete symmetry in the definitions of the domain and codomain functions).

Similarly, one can place simple linear time and space bounds on the computation of the juxtaposition and composition of derivation words (these linear bounds are based upon the lengths of the derivation words, not on the lengths of the derived words).

Certainly right reduction (and hence determination of the canonical derivation) can be performed in an amount of space proportional to the length of a derivation word. A derivation word of length \( n \) requires at most \( n \) right reductions, and each reduction can require a scan of the word, so the amount of time required to compute the canonical derivation is bounded by a linear function of \( n^2 \).

Received: October 15, 1973; Revised: January 8, 1975

References


