LEFT-FITTING TRANSLATIONS

H.P. Kriegel, Th. Ottmann
University of Karlsruhe, D-7500 Karlsruhe, Western-Germany

One rather natural method for defining translations is by specifying a pair of grammars generating the translation. If for each leftmost derivation d in the input grammar generating an input word x the "corresponding" derivation d' in the output grammar generates an output word x', we call this grammar pair left-fitting. This concept is motivated by the usual parsing algorithms yielding leftmost derivations and by the fact that the left-fitting translations are more powerful than the syntax-directed translations. It is shown that it is decidable whether or not a given pair of context-free grammars is left-fitting or not essentially using the fact that the set of left derivation trees of a context-free grammar is semilinear. By means of certain structure properties of left-fitting translations, it is shown that they form a proper hierarchy in their so called intercalation number.

1. Introduction

One usually defines a language, i.e. a set of words, by specifying a grammar which generates exactly the words of the language. In a similar way a pair of grammars can be used to define a translation, i.e. a set of pairs of words. Before giving the precise definition of the translation generated by a pair of grammars we recall some notions on context free (c.f.) grammars: A c.f. grammar $G = (N,\Sigma,P,S)$ consists of two disjoint finite sets N and Σ of nonterminals and terminals, respectively, a start variable $S \in \Sigma$, and a finite set of productions $P \subseteq N \times (N \cup \Sigma)^*$. A production $(A,\alpha) \in P$ is usually written as $A \to \alpha$. If $p = A \to \alpha \in P$, the application of p to a word $\beta A \gamma$ yields $\beta\alpha\gamma$, in symbols $\beta A\gamma \Rightarrow^p \beta\alpha\gamma$. For each sequence d of productions, $d = p_1,p_2,\ldots,p_n$, $n \geq 1$, write $\alpha_0 \Rightarrow^d \alpha_n$ instead of $\alpha_0 \Rightarrow^{p_1} \alpha_1 \Rightarrow^{p_2} \Rightarrow \ldots \Rightarrow^{p_n} \alpha_n$ and call d a derivation in G. The derivation d is called terminal if $S \Rightarrow^d x$ where $x \in \Sigma^*$. As usual $L(G) = \{x \in \Sigma^* \mid S \Rightarrow^d x$, for some derivation d$\}$ denotes the language generated by G.

For convenience it is assumed that each production $A \to \alpha$ contains each variable at most once in α and furthermore that grammars are always reduced, that means they contain no useless symbols. Let $G = (N,\Sigma,P,S)$ and $G' = (N',\Sigma',P',S')$ be two grammars with a one-one correspondence of their productions. For each production p in P let p' in P' be the corresponding production.

For each derivation d in G (at most) one corresponding derivation d' in G' can be associated. Apply the sequence d' of corresponding productions such that the following conditions, hold: If $S \Rightarrow^{d_1} \gamma \Rightarrow^p \beta \Rightarrow^{d_2} x$ and $S' \Rightarrow^{d_1'} \gamma' \Rightarrow^{p'} \beta' \Rightarrow^{d_2'} x'$ are

corresponding derivations with $p = A \to \alpha$, $p' = A' \to \alpha'$ then
(i) the leftmost A in γ is replaced and
(ii) if γ' contains a A' generated at the same time as the leftmost A in γ, then that A' is replaced; otherwise the leftmost A' in γ' is replaced. This choice of positions where to apply the productions of d' rules out certain undesired pairs of derivations.

Definition 1: The pair translation T(G,G') generated by the grammar pair (G,G') is defined by

$$T(G,G') = \{(x,x') \mid S \Rightarrow^{d} x,\ S' \Rightarrow^{d'} x', (d,d') \text{ a corresponding pair of terminal derivations in G and G'}\}$$

A major problem with translations generated by grammar pairs is the fact that for a terminal derivation in one grammar the sequence of corresponding productions in the other grammar is not necessarily again a terminal derivation.
If we are interested not only in a translation T but also in its inverse T^{-1}, it is reasonable to claim

Definition 2: A grammar pair (G,G') is called agreeable if for each terminal derivation d in G the sequence d' is a terminal derivation in G' and vice versa. A translation is called agreeable if it is generated by an agreeable grammar pair (G,G').

Using a result from PENTTONEN (1974) it is shown in KRIEGEL (1976) that the family of syntax-directed translations equals the family of agreeable translations.
In many applications one is only interested in the translation T, but not in its inverse T^{-1}.
Therefore it is sufficient to claim that for each inputword we can generate at least one outputword. This leads to:

Definition 3: A grammar pair (G,G') is called fitting if for each terminal derivation d in G the sequence d' is a terminal derivation in G'. A translation is called fitting if it is generated by a fitting grammar pair.

In KRIEGEL and MAURER (1976) it is shown that the problem whether a given grammar pair is fitting or not and the equivalent containment problem for Szilard languages "$Sz(G) \subseteq Sz(G')$" are decidable. The properties of fitting translations are investigated in KRIEGEL (1976).
The definition of fitting is not very realistic. Given a grammar pair (G,G') and an inputword $x \in L(G)$ we parse x yielding a derivation d such that $S \Rightarrow^{d} x$. But the usual parsing algorithms for an arbitrary context-free grammar G and for a given $x \in L(G)$ do not yield all derivations d such that $S \Rightarrow^{d} x$, but yield i.g. one special

derivation, usually a leftmost derivation d such that $S \overset{\ell}{\Rightarrow}{}^{d} x$. Applying only terminal leftmost derivations in G the property fitting is too strong. The adequate definition is

Definition 4: A grammar pair (G,G') is called left-fitting if for each terminal leftmost derivation d in G the sequence d' is a terminal derivation in G', i.e. $S \overset{\ell}{\Rightarrow}{}^{d} x$ where $x \in \Sigma^*$ implies $S' \Rightarrow^{d'} x'$ and $x' \in \Sigma'^*$.
A translation T is called left-fitting if there is a left-fitting grammar pair (G,G') such that

$$T = T_{\ell}(G,G') = \{(x,x') \in T(G,G') \mid S \overset{\ell}{\Rightarrow}{}^{d} x \text{ for some } d\}$$

By definition each fitting grammar pair is left-fitting, but there are left-fitting grammar pairs which are not fitting, as the following example shows. So claiming a grammar pair to be fitting is really too strong.

Example 1: Consider the grammar pair (G,G') where

$$G = (\{S,A,B\}, \{a,b\}, P,S),\ G' = (\{S',A',B'\}, \{0,1\},P',S')$$

and P,P' as follows:

$p_1 : S \to AB$	$p_1' : S' \to A'$
$p_2 : A \to aA$	$p_2' : A' \to 0A'$
$p_3 : A \to a$	$p_3' : A' \to B'$
$p_4 : B \to bB$	$p_4' : B' \to 1B'$
$p_5 : B \to b$	$p_5' : B' \to 1$

(G,G') is left-fitting, because for any terminal leftmost derivation p_1,p_2^n,p_3,p_4^m,p_5 where $n,m \geq 0$ also $p_1',p_2'^n,p_3',p_4'^m,p_5'$ is a terminal derivation in G'. But for the terminal derivation (not leftmost!) p_1,p_5,p_3 the sequence of corresponding productions p_1',p_5',p_3' is not a terminal derivation in G'. Therefore (G,G') is left-fitting but not fitting.
Many translations of practical interest can be generated by left-fitting grammar pairs such as the translation T_{dup} which duplicates each word x in an arbitrary context-free language to xx. This translation is built in in many translations describing certain inversions of data files (e.g. duplicating names).

Example 2: Consider the translation

$T_{Dup} = \{(x,xx) \mid x \in L_1\}$, $L_1 \in \underline{CF}$, where CF denotes the family of c.f. languages.
Let $G_1 = (N_1,\Sigma_1,P_1,S_1)$ be a context-free grammar such that $L(G_1) = L_1$ and let (G,G') be the grammar pair where $G = (N_1 \cup \{S\} \cup \{\textcircled{p} \mid p = A \to \alpha \in P_1\}, \Sigma_1,P,S)$

$$G' = (\{\overline{A} \mid A \in N_1\} \cup \{\bar{\bar{A}} \mid A \in N_1\} \cup \{S'\},\Sigma_1,P',S')$$

and P,P' as follows:

$S \to S_1$ $\qquad S' \to \overline{S}_1 \overline{\overline{S}}_1$

$A_1 \to (p)$ $\qquad \overline{A}_1 \to \overline{\alpha}_1$ $\qquad \forall P : A_1 \to \alpha_1 \in P_1$

$(p) \to \alpha_1$ $\qquad \overline{\overline{A}}_1 \to \overline{\overline{\alpha}}_1$

Here $\overline{\alpha}$ is obtained from α by replacing each variable A in α by $\overline{A}$, $\overline{\overline{\alpha}}$ is obtained in the analogous way.
Obviously (G,G') is fitting and therefore left-fitting and $T_\ell(G,G') = T_{Dup}$, but $T(G,G') \neq T_{Dup}$. Realize that the restriction to leftmost derivations in the input grammar G is necessary for this example.

2. Left sentential forms

We will show in 3. that for a grammar pair (G,G') it is decidable whether (G,G') is left-fitting or not. Moreover we will derive some structure properties of left-fitting translations. For this purpose we use a theorem on (derivation trees of) left sentential forms which can be considered as a kind of Parikh's theorem for left sentential forms.

Let $G = (N,\Sigma,P,S)$ be a context-free grammar.
Let $T_L(G)$ denote the set of all trees associated with leftmost derivations in G which are not terminal derivations. For short the trees in $T_L(G)$ are called left derivation trees.
The set of left sentential forms of G can be defined by
$S_L(G) = \{\text{frontier}(t) \mid t \in T_L(G)\}$ where frontier(t) denotes the string obtained by concatenating the leaves of t from left to right.
For each two subsets $U_1 \subseteq N$ and $U_2 \subseteq N-\{S\}$ and for each $A \in N$ we define a set of trees

$$T^A(U_1,U_2) = \{t \in T_L(G) \mid t \text{ has the root S, frontier}(t) = xA\alpha \text{ for some } x \in \Sigma^*, \alpha \in (N \cup \Sigma)^*, \text{ and conditions (1a), (1b) hold}\}$$

(1a) On the left of the path connecting the root with the leftmost leaf labelled A there occur exactly the variables in the set U_1

(1b) On the path connecting the root with the leftmost leaf labelled A there occur exactly the variables in $\overline{U}_2 = U_2 \cup \{S\}$.

The label of the root but not the label of the leaf is counted to the set of variables occurring on a path from the root to a leaf. Define
$L^A(U_1,U_2) = \{\text{frontier}(t) \mid t \in T^A(U_1,U_2)\}$.
For the following definitions let us consider an arbitrary but fixed $T^A(U_1,U_2)$. Define $u_1 = \#(U_1)$ and $u_2 = \#(\overline{U}_2)$ where $\#(M)$ denotes the number of elements of the set M.

For each $Y \in U_1$ and $z \in U_2$ we define sets $T_1^Y(U_1)$ and $T_2^Z(U_1,U_2)$ of derivation trees and left derivation trees, respectively, as follows:

$T_1^Y(U_1)$ = {t|t is a tree with root Y and frontier wYz (for some $w,z \in \Sigma^*$) associated with a derivation d in G where $Y \Rightarrow^d wYz$, such that conditions (2a) and (2b) hold}.

(2a) Each variable occurring in t is in U_1

(2b) t contains no path on which a variable $x \in U_1$ occurs more than u_1+1 times.

$T_2^Z(U_1,U_2)$ = {t|t is a tree with root Z and frontier $yZ\beta$ (for some $y \in \Sigma^*$, $\beta \in (N \cup \Sigma)^*$) associated with a leftmost derivation d in G where $Z \overset{\ell}{\Rightarrow}^d yZ\beta$, such that conditions (1a'), (1b'), (3a) and (3b) hold.

Conditions (1a') abd (1b') result from (1a) and (1b) by replacing "A" by "Z" and "exactly" by "no other variables than".

(3a) No subtree of t with the root on the left of the path connecting the root of t with the leftmost leaf labelled Z contains a path with more than u_1+1 occurrences of the same variable $X \in U_1$.

(3b) On the path connecting the root with the leftmost leaf labelled Z no variable $X \in \overline{U}_2$ occurs more than u_1+u_2+2 times.

The structure of trees defined so far is shown in Fig. 1:

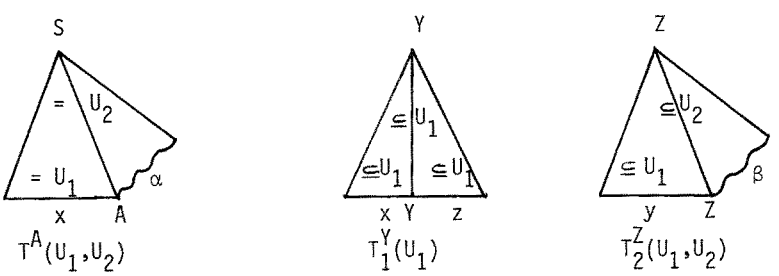

Fig. 1

Obviously $T_1^Y(U_1)$ and $T_2^Z(U_1,U_2)$ are finite sets of trees. For each left derivation tree t with frontier $xA\alpha$ let $\text{frontier}_N(t)$ denote the sequence of variables obtained from α by eliminating all terminals in α.

Define $H_2(H_1,U_2)$ = {$\text{frontier}_N(t)$|$t \in T_2^Z(U_1,U_2)$ for some $Z \in \overline{U}_2$}. Obviously $H_2(U_1,U_2)$ is finite.

For each tree $t \in T^A(U_1,U_2)$, for each (occurrence of) Y on the left of the path connecting the root of t with the leftmost leaf labelled A and for each $t' \in T_1^Y(U_1)$ a tree $\sigma_1(t,t')$ called 1-substitution of t' in t is defined as follows:

$\sigma_1(t,t')$ is obtained from t by replacing (the occurrence of) Y by the tree t'. Ob-

serve that $\sigma_1(t,t') \in T^A(U_1,U_2)$. In a similar way for each tree $t \in T^A(U_1,U_2)$, for each (occurrence of) $Z \in \overline{U}_2$ on the path connecting the root of t with the leftmost leaf labelled A, and for each $t' \in T_2^Z(U_1,U_2)$ a tree $\sigma_2(t,t')$ called the 2-substitution of t' in t is defined as follows: $\sigma_2(t,t')$ is obtained from t by replacing (the occurrence of) Z by the tree t'. Observe that $\sigma_2(t,t') \in T^A(U_1,U_2)$.

Trees obtained by 1-substitution and 2-substitution are shown in Fig. 2

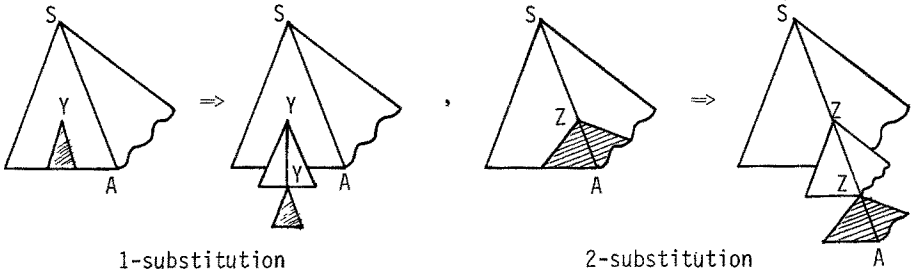

Fig. 2

We call a set $T \subseteq T^A(U_1,U_2)$ linear if there is a tree $t \in T^A(U_1,U_2)$ such that T is the smallest set of trees which contains t and is closed under 1-substitution and 2-substitution. A finite union of linear sets of trees is called semilinear.

We can prove in a tedious but straightforward way using arguments similar to those used in the proof of PARIKH's theorem (1966):

Theorem 1: For each context-free grammar G the set $T_L(G)$ of left derivation trees is semilinear.

As far as it is necessary for understanding Theorem 2 (the decidability of the property left-fitting) a rough sketch of the proof of Theorem 1 is given.

Obviously $T_L(G) = \bigcup_{\substack{A \in N \\ U_1 \subseteq N \\ U_2 \subseteq N-\{S\}}} T^A(U_1,U_2)$ holds.

Since the number of the sets U_1,U_2 and of the variables is finite, it suffices to show that each of the sets $T^A(U_1,U_2)$ is semilinear.

For each two subsets $U_1 \subseteq N$ and $U_2 \subseteq N-\{S\}$ and for $A \in N$ the set $T_0^A(U_1,U_2) \subseteq T^A(U_1,U_2)$ is defined as follows:

$$T_0^A(U_1,U_2) = \{t \in T^A(U_1,U_2) \mid t \text{ fulfills the conditions (3a') and (3b')}\}$$

Conditions (3a') and (3b') are obtained from (3a) and (3b) by replacing Z by A.

Define $H^A(U_1,U_2) = \{\text{frontier}(t) \mid t \in T_0^A(U_1,U_2)\}$. Clearly $T_0^A(U_1,U_2)$ and $H^A(U_1,U_2)$ are finite.

It can be shown that $T^A(U_1,U_2)$ equals the finite union of sets of trees each of which is the smallest set of trees which contains a tree $t \in T_0^A(U_1,U_2)$ and is closed under 1-substitution and 2-substitution. By Definition each $T^A(U_1,U_2)$ is semilinear and therefore $T_L(G)$ is semilinear.

2. The Decidability of the left-fitting Problem

Let (G,G') where $G = (N,\Sigma,P,S)$, $N = \{A_1,\ldots,A_n\}$, $S = A_1$ and $G' = (N',\Sigma',P',S')$, $N' = \{A_1',\ldots,A_{n'}'\}$, $S' = A_1'$ be a pair of c.f. grammars and let the 1-1 correspondence of productions be given by a mapping f from P onto P'.

Since terminals have no influence on leftmost derivations (in G) and neither terminals nor the position of variables have any influence on arbitrary derivations (in G'), we define a c.f. grammar G_ε and the vector form $\overline{G}'$ of G' as follows:

Convention: $G_\varepsilon = (N,\Sigma_\varepsilon,P_\varepsilon,S)$ where $\Sigma_\varepsilon = \phi$ and $P_\varepsilon = \{p_\varepsilon = A \to N(\alpha) \mid p = A \to \alpha \in P\}$ and $N(\alpha)$ is obtained from $\alpha \in (N \cup \Sigma)^*$ by erasing all occurrences of terminals in α.
$\overline{G}' = (\overline{N}',\overline{\Sigma}',\overline{P}',\overline{S}')$ where $\overline{\Sigma}' = \phi$ $\overline{N}' = \{e_1',\ldots,e_{n'}'\}$, (where e_i' denotes the vector of n' nonnegative integers which has the integer 1 at position i, the integer 0 at all other positions) $\overline{S}' = e_1'$ and $\overline{P}'$ consists exactly of all vector forms of productions of P': If $p' = A_j' \to \alpha'$ is a production of P', then
$p' = e_j' \to \psi'(\alpha') = (\#_{A_1'}(\alpha'),\ldots,\#_{A_{n'}'}(\alpha'))$ is its vector form, where $\#_{A_i'}(\alpha')$ denotes the number of occurrences of the variable A_i' in the word α'.

Let V_+' denote the set of all ordered n'-tupels of nonnegative integers. The concept of "derivation" can be transferred in a natural way to the vector form of a c.f. grammar:
For a production $p' = e_j' \to u' \in \overline{P}'$, $u' \in V_+'$ and vectors $v',w' \in V_+'$, $v' \Rightarrow^{p'} w'$ holds, if there exist words $\beta',\gamma' \in (N' \cup \Sigma')^*$ and a production $p' = A_j' \to \alpha' \in P'$ such that $\beta' \Rightarrow^{p'} \gamma'$, $\psi'(\alpha') = u'$, $\psi'(\beta') = v'$ and $\psi'(\gamma') = w'$.

If $p = A \to \alpha$ and $p' = A' \to \alpha'$ are corresponding productions in (G,G'), then $p_\varepsilon = A \to N(\alpha)$ and $p' = \psi'(A') \to \psi'(\alpha')$ are corresponding productions in $(G_\varepsilon,\overline{G}')$. Thus we have transferred the correspondence between sequences of productions of G and G' to a correspondence between sequences of productions of G_ε and $\overline{G}'$.

The questions whether (G,G') is a left-fitting grammar pair can now be reduced to the question whether for any terminal leftmost derivation d in G_ε the corresponding sequence d' of productions of $\overline{G}'$ yields the vector o' (the n'-tupel of o's). We call $(G_\varepsilon,\overline{G}')$ left-fitting if for any terminal leftmost derivation d, $A_1 \overset{\ell}{\Rightarrow}{}^{d} \varepsilon$ implies $e_1' \Rightarrow^{d'} o'$. Then it follows immediately:

Lemma 1: $(G_\varepsilon,\overline{G}')$ is left-fitting iff (G,G') is left-fitting.

Definition 5: Let $y \in N^*$ and d be a leftmost derivation in G_ε such that $y \overset{\ell}{\Rightarrow}{}^{d} \varepsilon$. Then d is called cycle-free if in the forest of left derivation trees associated to d (with the variables of y as roots) there is no path on which the same variable occurs more than once.

Hence for any left sentential form of G_ε there exists only a finite number of cycle-free terminal leftmost derivations in G_ε.

Definition 6: Let $d' = p'_1,\ldots,p'_k$ be any sequence of productions in $\overline{P}'$ such that $p'_i = e'_{j_i} \to u'_i$, $1 \leq i \leq k$.

The value of d', in symbols z(d'), is defined by

$$z(d') = \sum_{i=1}^{k} (e'_{j_i} - u'_i)$$

Clearly, if d' is a derivation in $\overline{G}'$ such that $v' \Rightarrow^{d'} o'$, then $z(d') = v' \in V'_+$.

Intuitively, for any terminal leftmost derivation d in G_ε z(d') indicates the number of occurrences of variables of N' which must be available to ensure that d' is a terminal derivation in G'.
For any A_i, $1 \leq i \leq n$, we define the set $g_\ell(A_i)$ as follows:

$$g_\ell(A_i) = \{z(d') \mid A_i \overset{\ell}{\Rightarrow}{}^{d} \varepsilon,\ d \text{ cycle-free}\}$$

g_ℓ is extended to N^* by defining $g_\ell(\varepsilon) = \{o'\}$ and for any $y \in N^+$

$$g_\ell(y) = \{v' \mid \exists w'_1 \in g_\ell(A_1),\ldots, \exists w'_n \in g\ (A_n) \text{ where}$$

$$v' = \sum_{i=1}^{n} \#_{A_i}(y)\ w'_i\}$$

Instead of $g_\ell(y) = \{w'\}$ we write $g_\ell(y) = w'$.

g_ℓ has the following properties:

Lemma 2: Let $G_\varepsilon = (\{A_1,\ldots,A_n\},\phi,P_\varepsilon,A_1)$ and $\#(g_\ell(A_i)) = 1$ for all i, $1 \leq i \leq n$. Then (1) - (4) hold:

(1) $\#(g_\ell(y)) = 1$ and $g_\ell(y) = \sum_{i=1}^{n} \#_{A_i}(y)\ g_\ell(A_i)$ for all $y \in N^+$.

(2) $g_\ell(y) = g_\ell(v) + g_\ell(w)$ for all $y,v,w \in N^+$ such that $\psi(y) = \psi(v) + \psi(w)$.

Here ψ denotes the Parikh-mapping which maps a word $x \in N^*$ onto the vector

$(\#_{A_1}(x),\ldots,\#_{A_n}(x))$.

(3) For any cycle-free leftmost derivation d and any $y \in N^+$ such that $y \overset{\ell}{\Rightarrow}^d \varepsilon$, $z(d') = g_\ell(y)$ holds.
(4) $\psi(w) = \psi(w')$ implies $g_\ell(w) = g_\ell(w')$ for any $w, w' \in N^+$.

Proof: Obvious

We can now give necessary and sufficient conditions for $(G_\varepsilon,\overline{G}')$ and therefore for (G,G') to be left-fitting. These conditions are decidable.

Theorem 2: $(G_\varepsilon,\overline{G}')$ is left-fitting iff the conditions (1) - (3) hold:

(1) $\#(g_\ell(A_i)) = 1$ for all i, $1 \leq i \leq n$.
(2) $g_\ell(A_1) = e_1'$
(3) Let $H = \bigcup_{\substack{A \in N \\ U_1 \subseteq N,\ U_2 \subseteq N-\{A_1\}}} H^A(U_1,U_2)$

and $H_2 = \bigcup_{\substack{U_1 \subseteq N \\ U_2 \subseteq N-\{A_1\}}} H_2(U_1,U_2)$

Then for any left sentential form $v \in H$ and any production $p \in P_\varepsilon$ (left-)applicable to v, $v \overset{\ell}{\Rightarrow}^p w$ implies $g_\ell(v) \Rightarrow^{p'} g_\ell(w)$.

Furthermore for any $v \in H$ and any $r \in H_2$, $g_\ell(v) \in V_+'$ and $g_\ell(r) \in V_+'$ hold.

Remarks: Observe that Lemma 2 and condition (1) imply that $\#(g_\ell(v)) = \#(g_\ell(w)) = 1$ and $g_\ell(w) \in V_+'$ hold. Therefore $g_\ell(v) \Rightarrow^{p'} g_\ell(w)$ is well-defined.

Proof: Part I: Suppose the conditions (1) - (3) hold. Lemma 2 and condition (1) imply $\#(g_\ell(y)) = 1$ for any $y \in N^*$.
It will be shown that for any left sentential form $y \in S_L(G_\varepsilon)$ and any $p \in P_\varepsilon$ such that $y \overset{\ell}{\Rightarrow}^p u$, $g_\ell(y) \Rightarrow^{p'} g_\ell(u)$ holds where $g_\ell(y), g_\ell(u) \in V_+'$.

Let $y \in S_L(G_\varepsilon)$ be a left sentential form, $p \in P_\varepsilon$ be a production such that $y \overset{\ell}{\Rightarrow}^p u$ and let A be the leftmost nonterminal in y. Then there is a left derivation tree t_y with root S whose frontier is y. Let U_1 be the set of variables occuring in the subtrees on the left of the path S-A and let $U_2 \cup \{S\}$ be the set of variables occuring on the path S-A. Then the considered left derivation tree t_y is in $T^A(U_1,U_2)$ and y is in $L^A(U_1,U_2)$.
By Theorem 1 there is a left derivation tree t_o in $T_o^A(U_1,U_2)$ with frontier $v \in H^A(U_1,U_2)$ such that $v \overset{\ell}{\Rightarrow}^p w$ (because A is again the leftmost nonterminal).

t_y is obtained from t_o by iterated 1- and 2-substitution of trees from $T_1^Y(U_1)$ for any $Y \in U_1$ and from $T_2^Z(U_1,U_2)$ for any $Z \in \overline{U}_2$, respectively. Since 1-substitution does not influence the frontier of t_o there are positive integers $k, n_1,\ldots,n_k$ and trees $t_1,\ldots,t_k \in \bigcup_{Z \in \overline{U}_2} T_2^Z(U_1,U_2)$, such that the frontier y of t_y is obtained from the frontier v of t_o by inserting $r_i = \text{frontier}_N(t_i) \in H_2(U_1,U_2)$ exactly n_i times, $1 \leq i \leq k$. Thus we have $\psi(y) = \psi(v) + \sum_{i=1}^{k} n_i\psi(r_i)$.

Choose $r \in N^*$ such that $\psi(r) = \sum_{i=1}^{k} n_i\psi(r_i)$.

By Lemma 2(2) it follows that $g_\ell(r) = \sum_{i=1}^{k} n_i g_\ell(r_i).g_\ell(r_i) \in V_+'$ by condition (3) and $n_i > 0$ for all i, $1 \leq i \leq k$, imply $g_\ell(r) \in V_+'$.

Let us summarize that $y \overset{\ell}{\Longrightarrow}^p u$, $v \overset{\ell}{\Longrightarrow}^p w$ and $\psi(y) = \psi(v) + \psi(r)$ hold. This implies $\psi(u) = \psi(w) + \psi(r)$.

$v \in H^A(U_1,U_2)$ by condition (3) implies $g_\ell(v) \Rightarrow^{p'} g_\ell(w)$. Since $g_\ell(v), g_\ell(w) \in V_+'$ by condition (3) and $g_\ell(r) \in V_+'$ as well, $g_\ell(v) + g_\ell(r) \in V_+'$ and $g_\ell(w) + g_\ell(r) \in V_+'$.

Thus $g_\ell(v) + g_\ell(r) \Rightarrow^{p'} g_\ell(w) + g_\ell(r)$ is well-defined, i.e., 2-substitution of the trees t_i, $1 \leq i \leq k$, in the tree t_o does not decrease any components of the g_ℓ-vectors of the frontiers of the trees.

By $\psi(y) = \psi(v) + \psi(r)$ and $\psi(u) = \psi(w) + \psi(r)$ and Lemma 2(2) it follows that

$$g_\ell(y) = g_\ell(v) + g_\ell(r) \text{ and } g_\ell(u) = g_\ell(w) + g_\ell(r).$$

Therefore $g_\ell(y) \Rightarrow^{p'} g_\ell(u)$ holds.

Thus $A_1 \overset{\ell}{\Longrightarrow}^{p_1} x_1 \overset{\ell}{\Longrightarrow}^{p_2} x_2 \overset{\ell}{\Longrightarrow}^{p_3} \ldots \overset{\ell}{\Longrightarrow}^{p_n} \varepsilon$ implies

$g_\ell(A_1) \Rightarrow^{p_1'} g_\ell(x_1) \Rightarrow^{p_2'} g_\ell(x_2) \Rightarrow^{p_3'} \ldots \Rightarrow^{p_n'} g_\ell(\varepsilon)$ for any terminal leftmost derivation $p_1,\ldots,p_n$. Observe that $g_\ell(A_1) = e_1'$ by condition (2) and $g_\ell(\varepsilon) = o'$ by definition.

Therefore $A_1 \overset{\ell}{\Longrightarrow}^{p_1,\ldots,p_n} \varepsilon$ implies $e_1' \Rightarrow^{p_1',\ldots,p_n'} o'$ for any terminal leftmost derivation $p_1,\ldots,p_n$. Consequently $(G_\varepsilon,\overline{G}')$ is left-fitting.

Part II: Suppose $(G_\varepsilon,\overline{G}')$ is left-fitting. It will be shown that conditions (1) - (3) hold.

(1) Suppose that there is an A_i, $1 \leq i \leq n$, such that $\#(g_\ell(A_i)) > 1$. Then there is a $v \in H^{A_i}(U_1,U_2)$ for some $U_1 \subseteq N$, $U_2 \subseteq N-\{S\}$ such that $\#g_\ell(v)) > 1$. Thus there exist $v_1', v_2' \in g_\ell(v)$ where $v_1' \neq v_2'$. Then there are cycle-free leftmost derivations

d_1 and d_2 such that $v \overset{\ell}{\Rightarrow}^{d_1} \varepsilon$, $v \overset{\ell}{\Rightarrow}^{d_2} \varepsilon$, $v_1' = z(d_1')$ and $v_2' = z(d_2')$. Since $v \in S_L(G_\varepsilon)$ there is a leftmost derivation d such that $A_1 \overset{\ell}{\Rightarrow}^{d} v \overset{\ell}{\Rightarrow}^{d_1} \varepsilon$ and $A_1 \overset{\ell}{\Rightarrow}^{d} v \overset{\ell}{\Rightarrow}^{d_1} \varepsilon$. $(G_\varepsilon,\overline{G}')$ left-fitting implies $e_1' \Rightarrow^{d'} v' \Rightarrow^{d_1'} o'$ and $e_1' \Rightarrow^{d'} v' \Rightarrow^{d_2'} o'$. Consequently $v' = z(d_1')$ and $v' = z(d_2')$ hold. By assumption we have $z(d_1') \neq z(d_2')$, a contradiction.

(2) Since G_ε is reduced there is a cycle-free leftmost derivation d such that $A_1 \overset{\ell}{\Rightarrow}^{d} \varepsilon$. $(G_\varepsilon,\overline{G}')$ left-fitting implies $e_1' \Rightarrow^{d'} o'$. By definition, $e_1' = z(d')$. Since $\#g_\ell(A_1)) = 1$ according to condition (1), $g_\ell(A_1) = z(d') = e_1'$ holds.

(3) (a) Suppose that there is a $v \in H$ such that $g_\ell(v) \notin V_+'$. Then $g_\ell(v) = z(d_1')$ for some cycle-free leftmost derivation d_1 such that $v \overset{\ell}{\Rightarrow}^{d_1} \varepsilon$. Since $v \in S_L(G_\varepsilon)$ there is a leftmost derivation d such that $A_1 \overset{\ell}{\Rightarrow}^{d} v$.

$(G_\varepsilon,\overline{G}')$ left-fitting implies $e_1' \Rightarrow^{d'} v' \Rightarrow^{d_1'} o'$. By definition $z(d_1') = v' \in V_+'$, a contradiction to the assumption $z(d_1') \notin V_+'$.

(3) (b) Suppose that there is a $r \in H_2$ such that $g_\ell(r) \notin V_+'$. Let r be in $H_2(U_1,U_2)$. Choose A and some $v \in H^A(U_1,U_2)$. By (3)(a) $g_\ell(v) \in V_+'$. Then there is a nonnegative integer n such that $g_\ell(v) + ng_\ell(r) \notin V_+'$. By Theorem 1 there is a left sentential form $y \in L^A(U_1,U_2)$ such that $\psi(y) = \psi(v) + n\psi(r)$. By Lemma 2(2) it follows that $g_\ell(y) = g_\ell(v) + ng_\ell(r)$ and $g_\ell(y) \notin V_+'$.
Since $y \in S_L(G_\varepsilon)$, $g_\ell(y) \notin V_+'$ can be led to a contradiction in an analogous way as in (3)(a).

(3) (c) Let $v \in H$ and $p \in P_\varepsilon$ be a production such that $v \overset{\ell}{\Rightarrow}^{p} w$.

Consider the case $w \neq \varepsilon$.

Since $v \in S_L(G_\varepsilon)$ and G_ε is reduced, there exist a leftmost derivation d_1 and a cycle-free leftmost derivation d_2 such that $A_1 \overset{\ell}{\Rightarrow}^{d_1} v \overset{\ell}{\Rightarrow}^{p} w \overset{\ell}{\Rightarrow}^{d_2} \varepsilon$. Since $\#(g_\ell(A_i)) = 1$ for all i, $1 \leq i \leq n$, by condition (1), $g_\ell(w) = z(d_2')$ holds by Lemma 2(3).

$(G_\varepsilon,\overline{G}')$ left-fitting implies $e_1' \Rightarrow^{d_1'} v' \Rightarrow^{p'} w' \Rightarrow^{d_2'} o'$. By definition of $z(d_2')$ we have $w' = z(d_2') = g_\ell(w)$. By the same argument as above, there exists a cycle-free derivation d_3 such that $A_1 \overset{\ell}{\Rightarrow}^{d_1} v \overset{\ell}{\Rightarrow}^{d_3} \varepsilon$ and $z(d_3') = g_\ell(v)$. $(G_\varepsilon,\overline{G}')$ left-fitting implies $e_1' \Rightarrow^{d_1'} v' \Rightarrow^{d_3'} o'$ and $v' = z(d_3') = g_\ell(v)$. Consequently $g_\ell(v) \Rightarrow^{p'} g_\ell(w)$ holds if $w \neq \varepsilon$.

Consider the case $w = \varepsilon$.

Then there is a leftmost derivation d_1 such that $A_1 \overset{\ell}{\Rightarrow}^{d_1} v \overset{\ell}{\Rightarrow}^{p} \varepsilon$. $(G_\varepsilon,\overline{G}')$ left-fitting implies $e_1' \Rightarrow^{d_1'} v' \Rightarrow^{p'} o'$. By definition of z(p') and Lemma 2(3) we have $g_\ell(v) = z(p') = v'$ and thus $g_\ell(v) = v' \Rightarrow^{p'} o' = g_\ell(\varepsilon)$. This concludes the proof of Theorem 2. □

Theorem 3: It is decidable whether a grammar pair (G,G') is left-fitting or not.

Proof: Given a grammar pair (G,G'), we transform it to $(G_\varepsilon,\overline{G}')$. Since the conditions (1) - (3) in Theorem 2 are decidable, we can decide whether $(G_\varepsilon,\overline{G}')$ and therefore (G,G') is left-fitting or not. □

Obviously the time complexity of the decision algorithm given by Theorem 2 is exponential in the number of nonterminals of G, because this already holds for #(H) in condition (3).
Concerning grammars interesting for practical applications, GÜNTHER (1976) has shown in his Master's Thesis that their behaviour is much better than exponential. The aim of this thesis was to realize the declaration of string procedures in higher programming languages by means of left-fitting grammar pairs. The implementation was carried out in PL/I.
In the next section some properties of left-fitting translations are investigated.

4. Properties of left-fitting translations

Definition 7: For a translation T the domain of T is defined by $dom(T) = \{x \mid (x,x') \in T \text{ for some } x'\}$ and the range of T is defined by $ran(T) = \{x' \mid (x,x') \in T \text{ for some } x\}$.
For a family $\mathfrak{T}$ of translations $dom(\mathfrak{T}) = \{dom(T) \mid T \in \mathfrak{T}\}$ and $ran(\mathfrak{T}) = \{ran(T) \mid T \in \mathfrak{T}\}$.
Let LFT denote the family of left-fitting translations and CF denote the family of context-free languages. Then by Definition 4 and by Example 2
($T_{Dup} \notin$ SDT, because $ran(T_{Dup}) = \{xx \mid x \in L_1\}$, $L_1 \in$ CF, is i.g. not a context-free language) we have:

Theorem 4: dom(LFT) = CF, SDT $\subsetneq$ LFT.
dom(LFT) = CF ensures that parsing algorithms for context-free grammars can be applied.

For the proofs of the following theorems see KRIEGEL (1976).

For a left-fitting translation T the Parikh-mapping of the language dom(T) which is context-free is a semilinear set. This implies:

Theorem 5: Let T be a left-fitting translation. Then the Parikh-mapping of the language ran(T) is a semilinear set.

For a left-fitting translation T an analogon to the pumping lemma holding for the context-free domain can be given for the range of T essentially using Theorem 1. This structure property characterizes the range of T more precise than the semilinearity. For this purpose we need

Definition 8: For any words x and $y = a_1,\ldots,a_n$ $n \geq 1$, $a_i \in \Sigma$ for some alphabet Σ,

$1 \leq i \leq n$, left shuff(x,y) denote the set of words obtained from x by inserting all symbols a_i, $1 \leq i \leq n$, in this order.

Theorem 6: Let T be a left-fitting translation and ran(T) be an infinite language. Then there exist constants p and q and a word y' where $0 < \ell(y') < q$ such that for any $u' \in ran(T)$ where $\ell(u') > p$ and any $i \geq 1$ $shuff(u',y'^i) \cap ran(T) \neq \phi$ holds.

Corrollary 7: Let $L \subseteq \Sigma^*$ be an infinite context-free language and $c \notin \Sigma$. Let f be a mapping from Σ^* into the set of nonnegative integers such that $f(x) = 0$ iff $x = \varepsilon$. Then

$$L^{pot} = \bigcup_{x \in L} \{(xc^{f(x)})^m | m \geq 1\} \notin ran(\underline{LFT}) \quad \text{holds.}$$

Corrollary 7 can be used for verifying that a given translation is not left-fitting.

Consider the translation $T = \bigcup_{i=1}^{\infty} \{(a^i c^m, (a^i c^i)^m) | m \geq 1\}$.

T has a context-free domain and its range $ran(T) = \bigcup_{i=1}^{\infty} \{(a^i c^i)^m | m \geq 1\}$ has a semilinear character. But T is not a left-fitting translation because choosing $L = \{a^i | i \geq 1\}$ and $f(x) = \ell(x)$ for all $x \in \{a\}^*$ Corrollary 7 implies $ran(T) = L^{pot} = \bigcup_{i=1}^{\infty} \{(a^i c^i)^m | m \geq 1\} \notin ran(\underline{LFT})$.

Realize that for a left-fitting grammar pair the vector $g_\ell(A_i)$ may have negative components. This leads to

Definition 9: Let (G,G') be a left-fitting grammar pair and let $(G_\varepsilon, \overline{G}')$ be the usual transformation. For a vector $v = (v_1,\ldots,v_n)$ define $|v|_+ = \sum_{i,v_i > 0} v_i$. The intercalation number I of (G,G') is defined by $I = \max_{1 \leq i \leq n} \{|g_\ell(A_i)|_+\}$.

A left-fitting translation T has the intercalation number $m \geq 1$ if there exists a left-fitting grammar pair (G,G') with intercalation number m such that $T_\ell(G,G') = T$.

Let $\underline{LFT}_m$, $m \geq 1$, denote the family of left-fitting translations with intercalation number $m' \leq m$, $m \geq 1$. For any word x and words $y_1,y_2,\ldots,y_n$ let $shuff_w(x,y_1,y_2,\ldots y_n)$ denote the set of words obtained from x by inserting the words $y_1,y_2,\ldots,y_n$ in this order.

Then Theorem 6 can be formulated more precise as follows:

Theorem 8: Let T be a left-fitting translation with intercalation number $m \geq 1$ and let ran(T) be an infinite language. Then there exist constants p and q and words

$y_1, y_2, \ldots, y_{2k}$, where $k \leq m$ and $0 < \ell(y_1 y_2 \ldots y_{2k}) < q$ such that for any $u' \in ran(T)$ where $\ell(u') > p$ $shuff_w(u', y_1, y_2, \ldots, y_{2k}) \cap A(T) \neq \phi$ holds.

By Theorem 8 it can be shown that the left-fitting translations form a proper hierarchy in their intercalation number.

Theorem 9: $\underline{LFT}_m \subsetneq \underline{LFT}_{m+1}$ for all $m \geq 1$ and $\bigcup_{m \geq i} \underline{LFT}_m = \underline{LFT}$.

References

GÜNTHER,C. (1976), Zeichenkettenmanipulation mit Formalen Übersetzungen, (String manipulation by formal translations), Master's Thesis at the Institut für Angewandte Informatik und Formale Beschreibungsverfahren, University of Karlsruhe.

KRIEGEL,H.P. (1976), Erzeugung von Übersetzungen durch Grammatikpaare (Generation of translations by grammar pairs), Ph. D. Thesis at the Institut für Angewandte Informatik und Formale Beschreibungsverfahren, University of Karlsruhe.

KRIEGEL,H.P. and MAURER,H.A. (1976), Formal Translations and Szilard Languages, Information and Control 30(1976), 187-198.

PARIKH,R.J. (1966), On context-free languages, Journal of the ACM 13(1966), 570-581.

PENTTONEN,M. (1974), On Derivation Languages Corresponding to context-free Grammars, Acta Informatica 3(1974), 285-291.