FORMAL TRANSLATIONS AND THE CONTAINMENT PROBLEM FOR SZILARD LANGUAGES

H.P.Kriegel and H.A.Maurer

Institut für Angewandte Informatik und
Formale Beschreibungsverfahren
Universität Karlsruhe
D-75 Karlsruhe 1, Postfach 6380, Fed.Rep.Germany

Abstract:

One of the methods used for defining translations is the socalled syntax-directed translation scheme which can be interpreted as a pair of rather similar grammars with the productions working in parallel. Because of the similarity of the grammars each of the two grammars "fits" the other in the sense that for each derivation process in one grammar leading to a terminal word the corresponding derivation process in the other grammar also leads to a terminal word. For many practical applications it suffices to consider the case that one of the grammars fits the other, but not necessarily conversely. Investigating this idea, translations are obtained which are more powerful than the syntax-directed. It is shown that one can determine whether a given grammar fits another given grammar. As a by-product, it is established that the containment problem for Szilard languages is decidable.

I. MOTIVATION AND DEFINITIONS

The concept of transforming certain sequences of symbols into other sequences of symbols is of crucial importance in many areas of computer science. Consider e.g. a programming language such as ALGOL 60. A compiler for ALGOL 60 supposedly transforms a given ALGOL 60 program - and such a program is nothing but a sequence of symbols, after all - into another sequence of symbols, namely the corresponding machine-language or assembly-language program. Or consider a commercial environment in which certain data files are to be restructured in a specified manner: again this is a situation which can be understood as a transformation of sequences of symbols.

One possibility for defining transformations of sequences of symbols is the notion of (formal) translation.

Definition 1:

A (formal) translation T is a set of pairs (x,x') of words x and x' over some alphabets Σ and Σ'. Intuitively, if (x,x') is element of a translation T, then x is the given input word and x' the desired output word. The set of all input words of T is called the domain of T and defined by $\mathrm{dom}(T)=\{x \mid (x,x')\in T \text{ for some } x'\}$. The range of T is the set of all output words of T and defined by $\mathrm{ran}(T)=\{x' \mid (x,x')\in T \text{ for some } x\}$.

For a family τ of translations $\mathrm{dom}(\tau)=\{\mathrm{dom}(T)|T\in\tau\}$ and $\mathrm{ran}(\tau)=\{\mathrm{ran}(T)|T\in\tau\}$.

Notation:

A context-free grammar, called grammar,for short, is denoted by a quadrupel $G=(N,\Sigma,P,S)$ where N is the set of nonterminals,Σ is the set of terminals, $N\cap\Sigma=\phi$, P is the set of productions and S the starting symbol.

In order to assign labels to the productions in P, we consider a set of labels Lab_P and a surjective mapping λ: $Lab_P \to P$. If $l\in Lab_P$ is one of the labels of the production $A\to\alpha$ then we write $A\overset{l}{\to}\alpha\in P$.The usual relation $\Rightarrow^*$ for derivations in G is extended in the following way:

(1) $\alpha_1 \overset{\varepsilon}{\Rightarrow}^*\alpha_2$, if $\alpha_1=\alpha_2\in(N\cup\Sigma)^*$

(2) $\alpha_1 \overset{l}{\Rightarrow}^*\alpha_2$, if $\alpha_1=\beta A\gamma,\alpha_2=\beta\alpha\gamma$
and $A\overset{l}{\to}\alpha\in P$

(3) $\alpha_1 \overset{dl}{\Rightarrow}^*\alpha_2$, if $\alpha_1 \overset{d}{\Rightarrow}^*\alpha_3 \overset{l}{\Rightarrow}^*\alpha_2$
where $d\in Lab_P^*$, $l\in Lab_P$

We often abbreviate $\alpha_1 \overset{d}{\Rightarrow}^*\alpha_2$ to just $\alpha_1 \overset{d}{\Rightarrow}\alpha_2$ and call the word d in Lab_P^* the controlword of this derivation. The controlword indicates in which sequence the productions are applied, but not at which place. Thus, different derivations may have the same controlword.

A derivation $S \overset{d}{\Rightarrow}\alpha_2$ is called terminal if α_2 is in Σ^* and in this case the terminal controlword d is said to generate x. A word β in $(N\cup\Sigma)^*$ is said to be a sentential form if there is a controlword d such that $S\overset{d}{\Rightarrow}\beta$. The set $L(G)=\{x\in\Sigma^*|\ S\overset{d}{\Rightarrow}x\}$ is called the language generated by G. The Szilard language of G, denoted by Sz(G), is the set of all control words of terminal derivations in G, i.e.

$$Sz(G)=\{d\in Lab_P^*\ |\ S\overset{d}{\Rightarrow}x,\ x\in L(G)\}.$$

For convenience it is assumed that grammars are always reduced, i.e. for each nonterminal $A\neq S$ there are controlwords d_1 and d_2 such that
$S\overset{d_1}{\Longrightarrow}xAz\overset{d_2}{\Longrightarrow}xyz$ for some x,y,z in Σ^*

Notation:

Throughout this paper, let $G=(N,\Sigma,P,S)$ and $G'=(N',\Sigma',P',S')$ be two reduced grammars such that $Lab_P=Lab_{P'}$. Now the relation Co from P to P' is defined by $(p,p')\in Co$ if p and p' have the same label $l\in Lab_P=Lab_{P'}$. Whenever a production p in P is applied in a derivation in G, one of the productions p' such that $(p,p')\in Co$, called a corresponding production, has to be applied in G'. For convenience, we choose $N=\{A_1,\dots,A_n\}$, $N'=\{A'_1,\dots,A'_{n'}\}$, $S=A_1$ and $S'=A'_1$.

Definition 1:

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

$T(G,G')=\{(x,x')\in\Sigma^*\times\Sigma'^*|\ S\overset{d}{\Rightarrow}x$ and $S'\overset{d}{\Rightarrow}x'$ for a terminal controlword $d\in Lab_P^*$ subject to condition (1) below$\}$

Condition (1):

If $S \overset{d_1}{\Rightarrow} \beta \overset{l}{\Rightarrow} \gamma \overset{d_2}{\Rightarrow} x\in\Sigma^*, S' \overset{d_1}{\Rightarrow} \beta' \overset{l}{\Rightarrow} \gamma' \overset{d_2}{\Rightarrow} x'\in\Sigma'^*$ and $(A \overset{l}{\rightarrow} \alpha,\ A' \overset{l}{\rightarrow} \alpha') \in Co$, then

(i) the leftmost A in β is replaced and

(ii) if β' contains an A' generated at the same time as the leftmost A in β, then that A' is replaced; otherwise the leftmost A' in β' is replaced.

The above condition rules out certain undesired pairs of terminal derivations. By determining the place where to a apply a given production, there is a unique derivation in G' for each terminal controlword $d\in Lab^*_P$. Larger examples for grammar pairs translating simple ALGOL 60 programs to equivalent assembly language programs are given in KANDZIA und LANGMAACK (1973) as well as in MAURER und SIX (1974).
A major problem with translations generated by grammar pairs is the fact that a terminal controlword in one grammar is not necessarily again a terminal controlword in the other grammar.
This leads to the introduction of agreeable grammar pairs and agreeable translations.

Definition 2:

A grammar pair (G,G') is called agreeable if Sz(G)=Sz(G') that is if each terminal controlword in one of the grammars G and G' is a terminal controlword in the other grammar. A translation is called agreeable if it is generated by an agreeable grammar pair.
PENTTONEN (1974) has shown that for two reduced context-free grammars G and G', Sz(G)=Sz(G') if and only if G and G' agree up to terminals, up to a one-one renaming of nonterminals and up to a permutation of nonterminals on the right hand side of corresponding productions, the correspondence given by a bijection from P onto P'.

Since the conditions in the above theorem are exactly those LEWIS and STEARNS (1968) and AHO and ULLMAN (1969 and 1972) used for defining syntax-directed translations, the family AT of agreeable translations equals the family SDT of syntax-directed translations and dom(SDT)=ran(SDT)=CF, where CF denotes the family of context-free languages.
For many applications, the translation process is only performed in one direction. This leads to the following

Definition 3:

A grammar pair (G,G') is called fitting if $Sz(G)\subseteq Sz(G')$ that is if each terminal controlword in G is a terminal controlword in G'. The translation T(G,G') is called fitting if it is generated by a fitting grammar pair (G,G').

II. PROPERTIES OF FITTING TRANSLATIONS

Let FT denote the family of fitting translations.

Theorem 1:

$\underline{SDT}\subsetneq\underline{FT}$

Proof:

By definition of agreeable translations, $\underline{SDT}\subseteq\underline{FT}$. Now consider the translation

$T=\{((abc)^n,a^nb^nc^n)|n\geq 1\}\notin \underline{SDT}$ (ran(T) is not a context-free language). T is generated by the fitting grammar pair (G,G'), where

$G=(\{A_1,\dots,A_6\},\{a,b,c\},P,A_1)$

$G'=\{A'_1,\dots,A'_4\}, \{a,b,c\},P',A'_1)$,

$Lab_P=Lab_{P'}=\{1,2,\dots,7\}$ and

productions in P	corresponding productions in P'
$(A_1 \xrightarrow{1} A_2$,	$A'_1 \xrightarrow{1} A'_2A'_3A'_4)$
$(A_2 \xrightarrow{2} aA_3$,	$A'_2 \xrightarrow{2} aA'_2\ )$
$(A_3 \xrightarrow{3} bA_4$,	$A'_3 \xrightarrow{3} bA'_3\ )$
$(A_4 \xrightarrow{4} cA_2$,	$A'_4 \xrightarrow{4} cA'_4\ )$
$(A_2 \xrightarrow{5} aA_5$,	$A'_2 \xrightarrow{5} a\ )$
$(A_5 \xrightarrow{6} bA_6$,	$A'_3 \xrightarrow{6} b\ )$
$(A_6 \xrightarrow{7} c$,	$A'_4 \xrightarrow{7} c\ )$ ■.

By definition, dom($\underline{FT}$) = $\underline{CF}$.

It can be shown easily that for a fitting grammar pair (G,G') the language ran(T(G,G')) is a matrix language. By the following theorem this inclusion is proper.

Theorem 2:

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

For the proof, a system of linear diophantine equations associated with G is used. A meaningful set of solutions of this system is considered and proved to be a semilinear set. Now a linear transformation is applied to yield the Parikh-mapping of ran(T).

Since there are matrix languages whose Parikh-mapping is not semilinear, this implies

Corrollary 3:

The family ran($\underline{FT}$) is a proper subset of the family $\mathcal{M}^{\varepsilon}$ of matrix languages: ran($\underline{FT}$)$\subsetneq\mathcal{M}^{\varepsilon}$. (The upper index ε indicates that productions $A\to\varepsilon$ are allowed).

For practically applying the concept of fitting translations, it is important to determine whether a given grammar pair is fitting or not. In an earlier report by KRIEGEL and MAURER (1974) it has been shown that this "fitting problem" and the equivalent containment problem for Szilard languages are decidable. An outline of the proof follows. It is obvious, to apply Parikh's theorem to the sentential forms of the grammar G. Since we are interested neither in the terminals nor in the position of the nonterminals, but only in the number of occurrences of the nonterminals, we characterize sentential forms in G by n-vectors whose i-th component indicates the number of occurrences of the nonterminal A_i, $1\leq i\leq n$.

Notation:

For some fixed natural number n, an n-vector is an ordered n-tupel of integers, an n_+-vector an ordered n-tupel of nonnegative integers.

0 denotes the zero-vector, e_i, $1\leq i\leq n$, the i-th unitary vector.
n-vectors are denoted by u,v,w,t,b, specially $0,e_i,1\leq i\leq n$, n'-vectors by u',v',w',t',b', specially $0',e'_i$, $1\leq i\leq n'$. Let V,V',V_+ and V'_+ denote the sets of all n-vectors, n'-vectors, n_+-vectors and n'_+-vectors, respectively.

Let the grammars $G=(N,\Sigma,P,S)$ and $G'=(N',\Sigma',P',S')$ now be in vector representation,i.e. $N=\{e_1,\dots,e_n\},\Sigma=\emptyset$, $S=e_1$ and a production in P has the form $e_i \overset{l}{\to} u$, $1\leq i\leq n$, u an n_+-vector and $l\in Lab_P$. The usual relation $\overset{d}{\Rightarrow}$, $d\in Lab_P^*$, for sentential forms is carried over to n_+-vectors. Clearly, d is a terminal controlword if $e_1 \overset{d}{\Rightarrow} 0$. An n_+-vector v is a sentential form in G if there is a controlword d such that $e_1 \overset{d}{\Rightarrow} v$. Since G is reduced, for each nonterminal e_i, $2\leq i\leq n$, there are controlwords d_1 and d_2 such that $e_1 \overset{d_1}{\Longrightarrow} e_i \overset{d_2}{\Longrightarrow} 0$.

G' is given in the analogous way. Clearly, for a fitting grammar pair (G,G') in vector representation $e_1 \overset{d}{\Rightarrow} 0$ implies $e'_1 \overset{d}{\Rightarrow} 0'$. Now by Parikh's theorem it follows immediately:

Lemma 4:

The set $M=\{v\in V_+ | v$ is a sentential form in $G\}$ is semilinear.

Definition 4:

Let $d=l_1\dots l_m\in Lab_P^*$, such that $e'_{j_i} \xrightarrow{l_i} u'_i\in P'$, $1\leq j_i\leq n'$ and $1\leq i\leq m$. The value of d, in symbols z(d), is defined by

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

Clearly, if d is a terminal controlword in G' such that $v' \overset{d}{\Rightarrow} 0'$, then $z(d)=v'\in V'_+$.
For any sentential form $v\in V_+$ in G the set f(v) is defined by $f(v)=\{z(d)\in V' | v \overset{d}{\Rightarrow} 0$, d cycle-free$\}$.
Additionally, define $f(0) = \{0'\}$.
A controlword d in G such that $v \overset{d}{\Rightarrow} 0$ is called cycle-free if in no branch of an associated derivation tree any nonterminal occurs more than once.
Note that the elements of f(v) may have negative components.
An n'_+-vector $w'\in f(v)$ can be considered a nonterminal balance vector which should be generated by the same controlword as v.
Let #(S) denote the number of elements of the set S. For the following definition, we suppose $\#(f(v))=1$ for each sentential form v in G, which will turn out to be reasonable in Theorem 5.

Definition 5:

Let $E\subseteq V_+$ be a linear set in the semilinear set M and $b_0,b_1,\dots,b_k\in V_+$ be a basis of E.

E is termed well-formed, if for all $t \in V_+$ such that $t=b_0+b_i$, $1 \leqq i \leqq k$, and for $t=b_0$, $t \xRightarrow{l} w$, $l \in Lab_P$, implies $f(t) \xRightarrow{l} f(w)$.

The grammar pair (G,G') is well-formed, if M is the finite union of well-formed linear sets.

Now we can state necessary and sufficient conditions that a grammar pair (G,G') is fitting.

Theorem 5:

(G,G') is a fitting grammar pair if and only if the conditions (1)-(4) hold:

(1) $\#(f(e_i))=1$ for all i, $1 \leqq i \leqq n$

(2) $f(e_1) = \{e_1'\}$

(3) Let $E \subseteq V_+$ be a linear set in the semilinear set M and $b_0, b_1, \ldots, b_k \in V_+$ be a basis of E. Then $f(b_i)$ is an n_+'-vector for all i, $0 \leqq i \leqq k$.

(4) The grammar pair (G,G') is well-formed.

Obviously, the conditions (1)-(4) in Theorem 5 are decidable. They can be easily formulated as algorithm for deciding whether or not a given grammar pair (G,G') is fitting.

Given a grammar pair (G,G'), we use the above algorithm to test whether (G,G') is fitting. If the result is "yes", we parse a given inputword $x \in L(G)$ (e.g. with Earley's algorithm) yielding a controlword d such that $S \xRightarrow{d} x$. Observing condition (1) of definition 1 this d generates an outputword x' such that $(x,x') \in T(G,G')$.

References:

AHO,A.V. and ULLMAN,J.D. (1972) The theory of parsing, translation and compiling, Vol. I: Parsing, Prentice-Hall, Series in automatic computation, Englewood Cliffs, N.J. (1972).

AHO,A.V. and ULLMAN,J.D. (1969), Syntax-directed translations and the pushdown assembler, Journal of Computer and System Sciences3 (1969), 37-56.

AHO,A.V. and ULLMAN,J.D. (1969), Properties of syntax-directed translations, Journal of Computer and System Sciences 3 (1969), 319-334.

KANDZIA,P. und LANGMAACK,H. (1973), Informatik: Programmierung, Teubner Studienbücher Informatik, Bd. 18, Stuttgart (1973)

KRIEGEL,H.P. and MAURER,H.A. (1974), Formal translations and the containment problem for Szilard languages, Report No. 23 of the Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe

LEWIS,P.M. and STEARNS,R.E. (1968), Syntax-directed translations, Journal of the ACM 15 (1968), 271-281.

MAURER,H.A. und SIX,H.W. (1974), Datenstrukturen und Programmierverfahren, Teubner Studienbücher Informatik, Bd. 25, Stuttgart (1974)

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