On the covering of syntax-directed translations for context-free grammars

R K SHYAMASUNDAR
National Centre for Software Development and Computing Techniques,
Tata Institute of Fundamental Research, Bombay 400 005

MS received 13 February 1978

Abstract. A necessary and sufficient set of conditions is obtained that relates any
two context-free grammars \( G_1 \) and \( G_2 \) with the property that whenever \( G_2 \) left—or right—covers \( G_1 \), the syntax-directed translations (SDT's) with underlying grammar \( G_1 \) is a subset of those with underlying grammar \( G_2 \). Also the case that \( G_2 \) left—or right—covers \( G_1 \) but the SDT's with underlying grammar \( G_2 \) is not a subset of the
SDT's with underlying grammar \( G_1 \) is considered; in this case an algorithm is
described to obtain the syntax-directed translation schema (SDTS) with underlying
grammar \( G_2 \) to the given SDTS with underlying grammar \( G_1 \), if it exists.

Keywords. Chomsky normal form; context-free grammar; covering of grammars;
semantics; syntax-directed translation schema.

1. Introduction

Context-free grammars have been used extensively for describing the syntax of pro-
gramming languages and natural languages. Parsing algorithms for context-free
grammars consequently play a useful role in the implementation of compilers and
interpreters for programming languages. Many of the parsing algorithms require
that the grammar under consideration be in some normal form or have some special
property such as not having left recursion for use with top-down parsing techniques.

We can consider a grammar \( G_2 \) to be similar from the point of view of parsing,
with respect to \( G_1 \) if \( L(G_2) = L(G_1) \) and we can express the left and/or right parse of a
sentence generated by \( G_1 \) in terms of its parse in \( G_2 \). In such a case, we say that \( G_2 \)
covers \( G_1 \); hence the parses in \( G_1 \) can be recovered from those in \( G_2 \) by suitable mapping.
Similar grammars play an important role in the construction of translators.
Different grammars for the same language, which are similar, can be used at different
stages of translation depending upon the requirements of the particular task.

The importance of syntax-directed translation is dealt with in detail by Aho and
Ullman [1] and Lewis and Stearns [5]. Intuitively, a syntax-directed translation
schema (SDTS) is simply a grammar in which translation elements are attached to
each production. Whenever a production is used in the derivation of an input sen-
tence, the translation element is used to help complete a portion of the output sentence
associated with the portion of the input sentence generated by that production.

An interesting problem relating the covering of grammars and the SDTS of the
grammars is the following: (cf. Aho and Ullman [1]).
Problem: Is it true that whenever $G_2$ left—or right—covers $G_1$, every SDTS with $G_1$ as underlying grammar is equivalent to an SDTS with $G_2$ as underlying grammar? The answer to this is in the negative (cf. Aho and Ullman[1]).

In this paper we give a necessary and sufficient set of conditions relating $G_1$ and $G_2$ so that whenever $G_2$ left—or right—covers $G_1$, the SDT with underlying grammar $G_1$ is a subset of the SDT with underlying grammar $G_2$. Also an algorithm is described to obtain the SDTS with underlying grammar $G_2$ equivalent to the given SDTS with underlying grammar $G_1$ if it exists.

1.1. Syntax-directed translations

Definition 1.1: SDTS is a 5-tuple,

$$T = (V_N, V_T, \triangle, R, S),$$

where $V_N$ is a finite set of nonterminal symbols, $V_T$ is a finite input alphabet, $\triangle$ is a finite output alphabet, $S$ is a distinguished nonterminal in $V_N$, the start symbol and $R$ is a finite set of rules of the form $A \rightarrow a, \beta$, where $a$ is in $(V_N \cup V_T)^*$, $\beta$ is in $(V_N \cup \triangle)^*$, and the nonterminals in $\beta$ are a permutation of the nonterminals in $a$.

Let $A \rightarrow a, \beta$ be a rule. To each nonterminal of $a$ there is associated an identical nonterminal of $\beta$. If a nonterminal $B$ appears only once in $a$ and $\beta$, then the association is obvious. If $B$ appears more than once, we use integer subscripts to indicate the association. This association is an intimate part of the rule.

A translation form of an SDTS, $T$, is defined as follows:
(i) $(S, S)$ is a translation form and the two $S$'s are said to be associated.
(ii) If $(aA\beta, a'\beta')$ is a translation form, in which the two explicit instances of $A$ are associated, and if $A \rightarrow \gamma, \gamma'$ is a rule in $R$, then $(a\gamma \beta, a'\gamma' \beta')$ is a translation form. The nonterminals of $\gamma$ and $\gamma'$ are associated in the translation form exactly as they are associated in the rule. The nonterminals of $a$ and $\beta$ are associated with those of $a'$ and $\beta'$ in the new translation form exactly as in the old. The association will again be indicated by superscripts when needed, and this association is an essential feature of the translation form.

If the translation forms $(aA\beta, a'\beta')$ and $(a\gamma \beta, a'\gamma' \beta')$, together with their associations, are related as above, then we write $(aA\beta, a'\beta') \rightarrow_T (a\gamma \beta, a'\gamma' \beta')$.

Definition 1.2: The translation defined by $T$, denoted $\tau(T)$, is the set of pairs

$$\{ (x, y) | (S, S) \rightarrow^* (x, y), x \in V^* \text{ and } y \in \Delta^* \}$$

If $T$ is an SDTS, then $\tau(T)$ is called SDT.

Definition 1.3: An SDTS is semantically unambiguous if there are no two distinct rules of the form $A \rightarrow a, \beta$ and $A \rightarrow a, \gamma$.

Covering of Grammars

Let $G_1 = (V_{N_1}, V_T, P_1, S_1)$ and $G_2 = (V_{N_2}, V_T, P_2, S_2)$ be context-free grammars (CFGs) such that $L(G_1) = L(G_2)$. 
Definition 1.4: We say that \( G_2 \) left-covers \( G_1 \) if there is a homomorphism \( h \) from \( P_2 \) to \( P_1 \) such that

(i) If \( S_2 \xrightarrow{\pi} w \), then \( S_1 \xrightarrow{h(\pi)} w \), and

(ii) For all \( \pi \) such that \( S_1 \xrightarrow{\pi} w \) there exists \( \pi' \) such that \( S_2 \xrightarrow{\pi'} w \) and \( h(\pi') = \pi \).

Terms and symbols not specifically defined here are employed with their usual meaning as in Aho and Ullman [1].

Definition 1.5: Terminating property

We say that \( G_2 \) and \( G_1 \) satisfy the terminating property whenever

(i) \( G_2 \) left-or right-covers \( G_1 \),

(ii) \( V_{N_2} = V_{N_1}^{(1)} \cup V_{N_2}^{(2)} \)

where \( V_{N_1}^{(1)} = \{ A \in V_{N_1} \mid l:A \rightarrow \alpha, \text{is in } P_2, \text{such that for all rules in } P_2 \text{ with } A \text{ as the left-side}, h(i) \neq \lambda \} \), and \( V_{N_2}^{(2)} = V_{N_2} - V_{N_2}^{(1)} \),

i.e., \( V_{N_1}^{(1)} \cap V_{N_2}^{(2)} = \phi \), and

(iii) If \( A \in V_{N_2}^{(1)} \) then if \( A \xrightarrow{\pi_1} x \in V_* \) then

\[ B \xrightarrow{\pi_2, h(\pi)} x' \in V_* \text{ where } \pi_2 = h(\pi_1). \]

In the following we assume that the grammars \( G_2 \) and \( G_1 \) are proper and non-redundant and hence the associated SDTS with the underlying grammars \( G_2 \) and \( G_1 \) will be free of useless nonterminals.

Proposition: If \( G_2 \) left—or right—covers \( G_1 \) (both \( G_2 \) and \( G_1 \) are non-redundant) and \( i:B \rightarrow x_1 B x_2 \) where \( x_1, x_2 \in V_* \) , is in \( G_2 \), then \( h(i:B \rightarrow x_1 B x_2) \neq \lambda \).

Proof: We have \( S_2 \xrightarrow{\pi} x \) where \( \pi \) is a sequence of productions (i.e., \( \pi = \pi_1 \pi_2 \ldots \pi_1 \pi_{i-1} \ldots \pi_r \)) and \( x \in V_* \), then by definition of covering, we should have

\[ S_1 \xrightarrow{h(\pi)} x. \]

* In a CFG, \( G = (V_N, V_T, P, S) \) a symbol \( X \) in \( V_N \cup V_T \) is said to be inaccessible if \( X \) does not appear in any sentential form.

A nonterminal is said to be useless if it cannot generate a terminal string.

A CFG is said to be cycle-free if there is no derivation of the form \( A \xrightarrow{\star} A \).

A CFG is said to be proper, if it is cycle-free and free of useless symbols.

A CFG is said to be nonredundant if it is proper and free of inaccessible symbols.
Now, consider $S_1 \xrightarrow{\pi} x'$ where $\pi' = \pi_1 \ldots \pi_{t+1} \ldots \pi_r$ and $x' \in V_T^*$. Then, $S_1 \xrightarrow{\pi} x'$ since $h(\pi') = h(\pi)$. This contradicts the definition of covering and hence the proposition follows.

**Section 1.2**

Let $G_2 = (V_{N_2}, V_T, P_2, S_2)$ left-or right-cover, $G_1 = (V_{N_1}, V_T, P_1, S_1)$. Without loss of generality, it can be assumed that $V_{N_1} \cap V_{N_2} = \emptyset$. Let $I$ and $J$ denote the set of labels of the production rules of $G_2$ and $G_1$ respectively.

Let $i : A \rightarrow a$ be a production rule belonging to $G_2$. In this section, corresponding to the nonterminal $A$ occurring on the leftside of the $i$th rule associate a set of strings over $(V_{N_1} \cup V_T)^*$ through the following steps: (In the following, associate a string with the $i$th rule and associate a string with the nonterminal occurring on the leftside of the $i$th rule, are used interchangeably, as it will be clear from the context).

**Step 1:** To begin with, associate with the nonterminal occurring on the leftside of the $i$th rule in $G_2$ for which $h(i) = j$, $j \neq \lambda$, the nonterminal belonging to $V_{N_1}$ occurring on the left side of the $j$th rule in $G_1$. This is done for all $i \in I$ such that $h(i) \neq \lambda$. Form the set,

$$H = \{ A \in V_{N_2} \mid h(i : A \rightarrow a) = j, j \neq \lambda \}.$$

**Step 2:** Consider the terminal rules in $G_2$

(a) Let $h(i) = \lambda$. In this case associate with the $i$th rule in $G_2$, the string belonging to $V_T^*$ occurring on the rightside of the $i$th rule.

(b) Let $h(i) = j$ and $j \neq \lambda$. In this case, in addition to the string belonging to $V_T^*$, appearing on the rightside of the $i$th rule, the nonterminal belonging to $V_{N_1}$ occurring on the leftside of the $j$th rule in $G_1$ is associated with the $i$th rule in $G_2$.

**Step 3:** Consider the nonterminal rule in $G_2$

(a) Form the set,

$$H_1 = \{ i \in I \mid h(i) = \lambda \}.$$

First, choose a rule from $H_1$ that has a right side consisting of nonterminals of the following two types only:

1. nonterminals (a subset of $V_{N_2}$) belonging to $H$.
2. nonterminals that have been already processed in step 2.

The existence of such a rule is assured from the fact that $G_2$ covers $G_1$, and from the fact that $G_1$ and $G_2$ are nonredundant grammars.

Associate with such a rule (say $i$th rule) strings over $(V_{N_2} \cup V_T)^*$ obtained by replacing each nonterminal occurring on the right-side of the $i$th rule, by its respective associated sets.
Next, consider the rule belonging to $H_1$ whose right-side consists of nonterminals that are already processed; associate with this rule the set of strings obtained by replacing each nonterminal by its respective associated sets.

Repeat this step for all members belong to $H_1$.

(b) Consider the set of nonterminal rules for which $h(i) = j, j \neq \lambda$.

Choose a rule that has in its right-side, nonterminals that have been already processed and associate a set of strings with this rule as indicated earlier.

Repeat this step till all the rules are exhausted.

From the above procedure, a set of strings over $(V_{N_1} \cup V_T)^n$ has been associated with every rule. Note that if $i_1: A \rightarrow a$ and $i_2: A \rightarrow \beta$ are two rules of $G_2$, then two different sets will be associated with $A$, one corresponding to rule $i_1$ and the other to rule $i_2$.

The following example illustrates the procedure.

**Example 1.1:** Let $G_1$ be the grammar

$$G_1: 1: S' \rightarrow 0. S' \rightarrow 1. 2: S' \rightarrow 0 1$$

and let $G_2$ be the Chomsky normal form (CNF) grammar equivalent to $G_1$

$$G_2: 1: S \rightarrow AB; 2: S \rightarrow AC; 3: B \rightarrow SC; 4: A \rightarrow 0; 5: C \rightarrow 1.$$ 

It can be observed that $G_2$ left-covers $G_1$ under the homomorphism $h(1) = 1$, $h(2) = 2$, and $h(3) = h(4) = h(5) = \lambda$.

From step 1, the first rule is associated with $S'$ and the second rule also is associated with $S'$.

From step 2(a), rule 4 is associated with 0 and rule 5 is associated with 1.

From step 3(a), rule 3 is associated with $S'1$.

From step 3(b), rule 2 is associated with 0 1 and rule 1 is associated with 0$S'1$.

**Section 1.3**

In the following, a sequence of production rules belonging to $I^*$, called the parse set, is associated with each member of the set of strings associated with each production rule of $G_2$. The meaning of parse set is made clear by the following example:

Consider example 1.1.

Rule 5 is associated with the string 1. Since the string 1 associated with this rule is the right-side of the rule 5, associate a parse $p_5$ (the suffix indicates the rule number). For convenience, the production number is represented as a suffix of $p$.

Consider rule 3. The string associated with this is $S'1$. Let us trace the steps involved during the process of association of the string $S'1$ with rule 3.

1. Use rule 3.
2. Substitute for $C$ the associated string (the corresponding parse associated with this string in $p_3$).
3. $S$ is replaced by $S'$. This step does not involve the application of any production rule $G_2$. It is obtained from the homomorphism $h$ of $I$ onto $J$. 

Therefore we associate a parse $\pi_3\pi_6$ with the string $0S'1$ associated with rule 3.

Consider rule 1 of $G_2$, (i.e., $S \rightarrow AB$). The string $0S'1$ is associated with this rule.

The sequence of steps associating $0S'1$ with rule 1 is given below:

- $S \rightarrow AB$; the parse associated with $S$ is $\pi_1$.
- $A$ is replaced by ‘0’; the parse associated with $A$ is $\pi_4$.
- $B$ is replaced by $S'1$; the parse associated with $B$ is $\pi_5\pi_6$. Hence, associate $\pi_1\pi_4\pi_5\pi_6$ $0S'1$ associated with rule 1.

In this manner associate a parse set with each element of the associated string of each production rule in $G_2$.

Now, obtain the following sets:

For those rules in $G_2$ for which $h(i) = j$, $j \neq \lambda$ obtain the set of associated strings with rule $i$. Let it be represented by $q_{ij}$. Let $p_{ij}$ be the corresponding parse set associated with the members of $q_{ij}$.

Let $p_j = \{p_{1,j}, p_{2,j}, \ldots, p_{k,j}\}$ and

$q_j = \{q_{1,j}, q_{2,j}, \ldots, q_{k,j}\}$ where $h(i_r) = j$ for $r = 1, 2, \ldots, k$ and $p_{i_r,j}$ is the parse set associated with rule $i_r$, for which $h(i_r) = j$.

Let $P = \{p_1, p_2, \ldots, p_m\}$, where $m$ is the number of production rules in $G_2$.

**Definition 1.6**: Define two homomorphisms $h_1$ and $h_2$ as follows:

$$h_1 : (V_{N_1} \cup V_T) \rightarrow V_{N_1}$$

$$h_1(x) = \begin{cases} 
\lambda & \text{if } x \in V_T \\
\lambda & \text{if } x \in V_{N_1} 
\end{cases}$$

Now $h_2$ is defined for strings associated with the production rules of $G_2$. The strings associated with each production are strings over $(V_{N_1} \cup V_T \cup \overline{V}_N)^*$ where

$$\overline{V}_N = \{A \in V_{N_2} \mid \text{for all rules with } A \text{ as the subject such that, } h(A \rightarrow a) = \lambda\}.$$  

[Here note that for $A \rightarrow x_1Ax_2x_3$, $x_2 \in V_T^*$, $h(A \rightarrow x) \neq \lambda$].

$$h_2 : (V_{N_1} \cup V_T \cup V_N) \rightarrow V_{N_1} \cup \{\lambda\}$$

$$h_2(x) = \lambda \text{ if } x \in V_T, \lambda \text{ if } x \in \overline{V}_N \text{ and } x \text{ if } x \in V_{N_1}.$$  

Before obtaining the required conditions we prove the following lemma:

**Lemma 1.1**: If $h(i : A \rightarrow a) = j$, $j \neq \lambda$, then $h_1(a)$ matches with at least one element of $h_2(q_j)$.

**Proof**: Consider a rule in $G_2$, $i : A \rightarrow a$ such that $h(i) = j$ (let $j : B \rightarrow \gamma$).

Consider an element associated with the $i$th rule in $G_2$, and let $h(i) = j$, $j \neq \lambda$. From the process of association given earlier, it is clear that the parse associated with every element of $q_j$ has at least one production rule belonging to $G_2$ which does not map to $\lambda$. 
Let an element associated with the $i$th rule in $G_9$ be represented by,

$$a_1 A_1 a_2 \ldots A_m a_{m+1},$$

where $h_3(a_i) = \lambda$, for all $1 \leq i \leq m+1$, and $A_1, \ldots, A_m$ are nonterminals in $V_{N_9}$ such that $P_9$ contains at least one rule with $A_i$ as the subject for which the mapping $h$ is nonempty.

We have,

$$A \rightarrow^{G_9} a_1 A_1 a_2 \ldots A_m a_{m+1}. \tag{1}$$

Consider the derivation of a terminal string from $A$ such that

$$A \rightarrow_{\pi_1, \pi_1} x_i \in V_T^* \text{ where } h(\pi_{A_i}) \neq \lambda.$$

Since by the definition of covering,

$$A \rightarrow_{G_9} x \in V_T^*, \text{ implies}$$

$$B \rightarrow_{G_9} x' \in V_T^*,$$

there should be a nonterminal in $\gamma$ corresponding to every $A_i$ in (1). [Here 'correspondence' is understood in the following way: If $h(i) = j$ and $i : A \rightarrow a$ and $j : B \rightarrow \gamma$, then $B$ corresponds to $A$]

Hence the lemma follows.

Section 1.3.1

Let $M_j = \{x \in p_{i_j} \mid h(i) = j, \text{ if the } j\text{th rule in } G_1 \text{ is of the form } j : A' \rightarrow a' \text{ then the element in } q_{i_j} \text{ corresponding to } x, \text{ say } \beta, \text{ satisfies the property, } h_3(\beta) = h_1(a')\}.$

Let $M = \{M_1, M_2, \ldots, M_m\}$, where $m$ is the number of rules in $G_1$.

Example 1.2: Consider example 1.1. The grammar $G_1$ has the production rules, 1: $S' \rightarrow 0S'1$ and 2: $S' \rightarrow 01$. It may be observed that the right side of rule 1 in $G_1$, viz., $0S'1$, matches with the string associated with rule 1 of $G_2$ in the sense of lemma 1.1. Similarly, the right side of rule 2 in $G_2$, viz., $01$, matches with the string associated with rule 2 of $G_2$; in fact, in this case the strings are identical.

Section 1.4

In this section, we obtain a sufficient set of conditions for the problem cited in section 1.

Remark 1.1: Consider a rule $A \rightarrow BCD \tag{1}$

and two other rules: $F \rightarrow BE \tag{2}$

and $E \rightarrow CD \tag{3}$
Here, $A, B, C, D, E$ and $F$ are all nonterminals. Now, the direct derivation of rule 1 can be represented by

$\begin{array}{c}
\text{A} \\
\text{B} \quad \text{C} \\
\text{D}
\end{array}$

and corresponding derivation with rule 2 and 3 together can be represented by

$\begin{array}{c}
\text{F} \\
\text{B} \quad \text{E} \\
\text{C} \quad \text{D}
\end{array}$

The frontiers of these two trees are identical. Now, consider a translation, $CDB$, associated with rule 1; the corresponding translation can be achieved by associating the translations $EB$ and $CD$ respectively, with rules 2 and 3. However, if we associate a translation $CBD$ with rule 1, then it is impossible to obtain the same translation from the set of rules 2 and 3.

**Theorem 1.1:** If $G_2$ left-or right-covers $G_1$ satisfying the terminating property (definition 1.5), then the following conditions form a sufficient set of conditions for the problem cited in section 1.

(i) each member of $M_i$, $1 \leq i \leq m$ ($m$ is the number of rules in $G_i$), consists of at most one multiple-length production* (MLP) rule and

(ii) $M_i \cap M_j$ ($i \neq j$) consists of at most of only simple length production (SLP) rules for all $i \neq j$, so that the SDTS (semantically unambiguous) with the underlying grammar $G_1$ is a subset of the SDTS with underlying grammar $G_2$.

**Proof:** Consider condition (i)

Since the elements in $M_i$ satisfy condition (i), a typical structure for an element in $M_i$ will have the form shown below:

$\begin{array}{c}
\text{A} \\
\text{B} \\
\text{A_1} \quad \text{A_2} \ldots \quad \text{A_n}
\end{array}$

Let $a$ be the element of the rule $D \to a$ in $G_1$ corresponding to this element in $M_i$ that satisfies lemma 1.1.

Let $a = x_1 \ D_1 \ x_2 \ ... \ x_r \ D_r \ x_{r+1}$, where $x_i \in V_T$, and $D_i \in V_{N_i}$.

*A production is said to be an MLP if the number of symbols occurring on the right-hand-side of the production is more than one; a production is said to be an SLP if there is only one symbol on the right-hand side.
SDT for context-free grammars

Then

\[ h_1(x_1 \ D_1 \ x_2 \ \ldots \ D_r \ x_{r+1}) = h_2(A_1 \ A_2 \ \ldots \ A_n) \]

i.e., \( D_1 \ D_2 \ \ldots \ D_r = A_i \ \ldots \ A_{i'} \)

where \( i' = i + 1 = r. \)

Now the translations that can be associated with the rule \( D \rightarrow a \) in \( G_1 \) corresponds to the permutations of \( D_1, D_2, \ldots, D_r. \) To get a corresponding translation in \( G_2, \) we have to obtain a corresponding permutation of \( A_1, \ldots, A_{i'}. \) This is always possible since \( A_i, \ldots, A_{i'} \) are direct descendants of the node \( B \) (except for chains). Thus all translations defined* by,

\[ x'_1 \ D_{p(1)}' \ x'_2 \ D_{p(2)}' \ \ldots \ x'_{r} \ D_{p(r)}' \ x'_{r+1} \]

associated with the rule \( D \rightarrow a \) in \( G_1, \) can be achieved by associating the corresponding translations with \( B. \)

Consider condition (ii) assuming that condition (i) holds.

Case (a): Let \( (M_i \cap M_j) = \phi. \) In this case it is obvious that there is no chance of two distinct translations being associated with any rule of \( G_2. \)

Hence sufficiency holds in this case.

Case (b): Let \( (M_i \cap M_j) \) consist of SLP rules. Here again the translation associated with \( G_2 \) will be unambiguous; this is illustrated below by considering the typical tree structure shown below:

```
  B
 /\  \
|  |
X  X  \ldots X
 |  |
A  A  \ldots A
```

Now, since the translation associated with the rules in \( G_2 \) is associated with the node \( B \) (subject of MLP's) and not with the subject of SLP's there is no chance of ambiguous translation being associated with any rule.

In deriving the trees of the SDT's with underlying grammar \( G_2 \) we essentially replace a node and its direct descendants in the tree of \( G_1 \) by a corresponding tree structure and thus the equivalence of the two SDT's follows since \( G_2 \) left—or right—covers \( G_1, \) and \( G_2 \) and \( G_1 \) satisfy the terminating property.

Hence the theorem follows.

*\( p \) is a permutation mapping.
**Section 1.5**

In this section the necessary set of conditions are developed by refining conditions (i) and (ii) of theorem 1.1.

*Claim 1*: Condition (i) of theorem 1.1 can be replaced by the condition that each member of $M_i$, $1 \leq i \leq m$ satisfies.

(a) The condition (i) specified in theorem 1.1, or

(b) The condition that each member of $M_i$, $1 \leq i \leq m$ consists of one MLP rule [say $A \rightarrow a$] for which $|h_2(a)| \geq 2$ and many MLP rules for which $|h_2(a)| \leq 1$.

*Proof*: A typical tree structure corresponding to an element satisfying the above condition will have the form shown below:

![Tree Diagram](attachment:image.png)

Now consider the tree structure after pruning all the leaves whose image under $h_2$ is λ. After this operation, the tree structure will take the form shown below:
This tree structure has the same form as discussed in theorem 1.1. Hence the claim follows.

Note: Without any ambiguity, we will be in a position to associate the translation with the node which has descendants more than or equal to two after taking the image of the leaves under the mapping $h_2$. [i.e., $|h_2(\alpha)| \geq 2$]: note that an element of $M_1$ does not consist of more than one element of the form $\pi_i$ for which

$h(\pi_i) \neq \lambda$.

Claim 2: Condition (i) can be replaced by the condition that for each element in $M_1$, all the distinct translations that can be effected on a corresponding element of $G_1$ can be obtained by the corresponding translations on the elements of $M_1$.

Consider a parse structure corresponding to an element in $M_1$. Let the frontier of the tree be represented by $X_1 X_2, ..., X_n$. Let $h_2(X_1 X_2, ..., X_n) = X_i, ..., X_i'$. Let the corresponding element in $G_1$ be $a$ [i.e., the right side of the rule $h(i)$]. In that case if $a = x_1 D_1 x_2, ..., D_r x_{r+1}$ then from lemma 1.1 we have

$h_1(a) = D_1, ..., D_r = h_2(X_1 X_2, ..., X_n) = X_i, ..., X_i'$

and $i' - i + 1 = r$.

Translation on $D_1, ..., D_r$ corresponds to the permutations of $D_1, ..., D_r$. What claim 2 says is that all distinct permutations of $D_1, ..., D_r$ can be obtained for $X_i, ..., X_i'$. The proof of claim 2 follows from theorem 1.1. However, the condition is illustrated below:

Let $h(i) = j$ and $j: D \to a$ in $G_1$. Let $h_1(a) = A_1 A_2 A_1$. Let the corresponding element in $M_1$ have the parse structure shown below:

```
X
/   \
|   |
B_1 B_2 ... B_i+1 B_n
/   \
|   |
Y
/   \
|   |
D_1 ... D_r
```

Now from lemma 1.1 it follows that,

$h_2(B_1 B_2 ... B_i D_1 ... D_r B_{i+2} ... B_n)$
consists of only three elements, say $B_i D_j D_k$. Then, it will have the structure

Now $A_1 A_2 A_3$ and $B_i D_j D_k$ are in correspondence i.e., the first $A_1$ corresponds to $B_i$, $A_2$ corresponds to $D_j$ and second $A_1$ corresponds to $D_k$. The possible translations that can be associated with $A_1 A_2 A_3$ are the permutations of $A_1 A_2 A_3$, i.e., $A_1 A_2 A_3$, $A_2 A_1 A_3$, $A_2 A_1 A_3$ and that is all: these can be achieved by the set of translations

(a) $X \rightarrow B_i B_n, B_i B_n$ and $B_n \rightarrow D_j D_k, D_j D_k$.

(b) $X \rightarrow B_i B_n, B_i B_n$ and $B_n \rightarrow D_j D_k, D_k D_j$ and

(c) $X \rightarrow B_i B_n, B_n B_i$ and $B_n \rightarrow D_j D_k, D_j D_k$, respectively.

Claim 3: The condition specified by claim (2) is also necessary.

Proof: Now let an element of $M_f$ has the parse consisting of more than or equal to two MLPs for which the length of their images under $h_2$ is greater than or equal to 2. Then a typical structure of such an element has the form shown below:

Let $h_2 (A_1 A_2, \ldots, A_{n-1} B_1 B_2, \ldots, B_m)$ match with a string $a$ over $(V_N \cup V_T)^*$ which is a right side of a rule in $G_f$ corresponding to $M_f$. Let $a=x_1 D_1 x_2, \ldots, D_r x_{r+1}$.

Now $|h_2 (A_1 A_2, \ldots, A_{n-1} B_1 B_2, \ldots, B_m)| \geq 3$, and let it be equal to $D_1, \ldots, D_r$. In other words, the tree structure after obtaining the $Hh_2$ image of the eaves will have the form shown below where $i-i' \geq 1$ and $j-j' \geq 1$. 


Now from remark 1.1 it follows that if all $D_i$'s are distinct, a permutation exists, i.e., $D_{p(1)}, \ldots, D_{p(r)}$ for which corresponding elements in $G_2$ will be of the form,

$$A \rightarrow A_r \rightarrow A_{r+1} \rightarrow \ldots \rightarrow A_{r+j} \rightarrow B_{j+1} \rightarrow \ldots \rightarrow B_{j+k}$$

This type of translation cannot definitely be obtained in $G_2$ (cf. remark 1.1).

Hence the necessity of the above condition follows.

Claim 4: Condition (ii) of theorem 1.1 can be replaced by the following condition:

$M_i, 1 \leq i \leq j$ satisfy the condition of claim 3 and $M_i \cap M_j, i \neq j$, consist of MLP rules for which the length of the homomorphic image (i.e., $h_2$) is less than or equal to 1.

Proof: If $M_i \cap M_j$ consist of MLP rules for which $|h_2(a)| \geq 2$, then it should be obvious from the argument of theorem 1.1, that this gives rise to elements of the form $A \rightarrow a, y'$ and $A \rightarrow a', y$ where $y \neq y'$ with the SDTS with underlying grammar $G_2$; in that case the SDTS defined will be ambiguous. This establishes the necessity.

The sufficiency of condition of claim 4 is established in the following:

Now, let $M_i \cap M_j$ consist of MLP for which $|h_2(a)| = 1$ and $|a| \geq 1$

In this case it follows that the effective translation will be associated with the MLP rule for which $|h_2(a)| \geq 2$ and with the MLP rules (say $B \rightarrow \beta$), for which $|h_2(\beta)| \leq 1$, we associate a translation of $h_2(\beta)$ itself. Thus no contradiction results; the case $|h_2(\beta)| \leq 1$ and $|\beta| = 1$ has already been treated in theorem 1.1. Thus the sufficiency of the condition is established.

The above case analysis leads us to the following theorem.

Theorem 1.2: The conditions

(A) $M_i, 1 \leq i \leq m$, have elements consisting of at most one MLP rule or consists of one MLP rule for which $|h_2(a)| \geq 2$ and many MLP rules for which $|h_2(a)| \leq 1$ or consists of MLP rules which can effect all the distinct translations that can be effected on the rule in $G_1$ corresponding to the element considered in $M_i$. 


(B) \((M_i \cap M_j)\) for \(i \neq j\) consists only rules for which \(|h_\delta(a)| \leq 1\), are both necessary and sufficient so that the SDTS with underlying grammar \(G_\delta\) is a subset of the SDTS with the underlying grammar \(G_\delta\), whenever \(G_\delta\) left—or right—covers \(G_i\), and \(G_\delta\) and \(G_\delta\) satisfy the terminating property.

**Example 1.3**

\[
\begin{align*}
G_3 & : \quad 1: S \rightarrow 0A1 \\
    & : \quad 2: A \rightarrow 1A1 \\
    & : \quad 3: A \rightarrow 11 \\
G_4 & : \quad 1: S' \rightarrow 01A'1, 01A1 \\
    & : \quad 2: A' \rightarrow 1A'1, 1A1 \\
    & : \quad 3: A' \rightarrow 1, 1
\end{align*}
\]

Here \(G_4\) left-covers \(G_\delta\) (also \(G_1\) left-covers \(G_\delta_1\)) under the homomorphism \(h(1) = 1, h(2) = 2, h(3) = 3\). Also, \(G_4\) and \(G_\delta\) satisfy the terminating property (definition 1.5). It can be observed from the strings associated with \(G_4\), that theorem 1.1 holds in this case (The SDTS with \(G_3\) is a subset of the SDTS with \(G_\delta\)).

The grammars \(G_3\) and \(G_\delta\) of example 1.1 also satisfy theorem 1.1. Example 1.3 is given to illustrate that the strings associated with the rules of \(G_4\) do not exactly match with the right sides of \(G_3\). In the next section we consider the practicability of the conditions derived in this section.

**Section 1.6**

Before describing the practicability of the condition we comment on the terminating property mentioned in definition 1.5.

It should be noted that many of the normal form of grammars that cover their original grammars satisfy the terminating property; e.g., grammars in two canonical form, invertible context-free grammars (cf. Gray and Harrison [4]); note that a grammar need not be covered by any grammar in Greibach normal form.

In most formal treatments of parsing, the parser must enumerate all the nodes of the parse tree. In programming practice, certain nodes of the parse tree have no semantic significance and do not need to be present in a similar grammar. For example, consider the generation tree shown in figure 1, which occurs in EULER given in Wirth and Weber [6]; see also Gray and Harrison [4].

The chain expr → lambda is typical of what happens in grammars for programming languages. Chains exist to enforce precedence among operators and to collect several categories of syntactic types.

Chain productions rarely have semantic significance. In our running example, only the following productions have nontrivial semantics.

\[
\begin{align*}
\text{expr} & \rightarrow \text{var} \leftarrow \text{expr} \\
\text{var} & \rightarrow \text{lambda} \\
\text{primary} & \rightarrow \text{var}
\end{align*}
\]
Section 1.7

In this section a general algorithm is described that yields the SDTS (semantically unambiguous) with underlying grammar $G_2$ equivalent to the given SDTS (semanti-
cally unambiguous) with underlying grammar $G_4$ (Note that $G_2$ left—or right—covers $G_1$, and $G_2$ and $G_3$ satisfy the terminating property). Thus it is not necessary that the conditions stipulated in theorems 1.1. and 1.2 should be satisfied by $G_2$ and $G_4$ if the algorithm is to be applicable.

Algorithm:

Input: Grammars $G_2$ and $G_1$ ($G_2$ left—or right—covers $G_1$) and SDTS with underlying grammar $G_1$.

Output: The SDTS with underlying grammar $G_2$ equivalent to the given SDTS with underlying grammar $G_1$, if it exists; otherwise 'NO'.

Method:

Step 1(a): Find the sets $M_i$, $1 \leq i \leq m$, where $m$ indicates the number of production rules in $G_1$.

Set $i = 1$; let $n_i$ indicate the number of elements in the set $M_i$ i.e., $h(i) = i$, $1 \leq i \leq n_i$.

Step 1(b): Set $n = 1$.

Step 2: For the $n$th element in $M_i$ where $1 \leq n \leq n_i$, construct the derivation tree. The frontier of this tree is expressed in terms of the bracket language of the grammar $G_2$ (cf. Chartres and Florentin [2]). We illustrate this through an example.

\[ G : S \rightarrow AB \]
\[ A \rightarrow a \]
\[ A \rightarrow S \]
\[ B \rightarrow b \]
\[ B \rightarrow SS \]

To obtain the bracket language the grammar is augmented as follows: For each nonterminal $A$ which has, say, $r$ different productions $A \rightarrow \phi_k$, $k = 1, 2, \ldots, r$, introduce left and right brackets $A_{l_k}$ and $A_{r_k}$, $k = 1, 2, \ldots, n$. Let $L$ and $R$ denote the sets of left and right brackets, respectively, so introduced. The $k$th production $A \rightarrow \phi$ in $G$ for $A$, now carries the subscript $k$. It actually represents $A \rightarrow A_{l_k} \phi A_{r_k}$. 
The grammar \( G \) is converted into \( G' = (V_N, V_T, S, L, R, P') \), where

\[
L = \{ S_1', A_1', A_2', B_1', B_2' \}
\]

\[
R = \{ S_1', A_1', A_2', B_1', B_2' \}
\]

The sentence \( abb \) belonging to \( L(G) \) having the structure shown corresponds to the sentence in \( L(G') \),

\[
S_1'( A_1' S_1' A_1' aA_1' B_1' bB_1' S_1' A_2' B_1' bB_1' S_1')
\]

**Step 3:** Let \( h(i_k) = i \), \( i \neq \lambda, i_k \leq n_1 \).

Let the frontier of the derivation tree of an element of \( M \) be of the form \( X', ..., X' \).
Let the next immediate leftmost symbol belonging to \( L \) be \( Y_1' \). Find the corresponding (leftmost) element belonging to \( R \) viz., \( Y_1' \). Consider the symbols from \( Y_1' \) to \( Y_1' \) as forming a single group. Next find out whether there is any symbol belonging to \( L \) beyond \( Y_1' \) (i.e., beyond the group obtained earlier). Let it be \( Z' \). Then find the corresponding \( Z' \). The sequence of symbols from \( Z' \) to \( Z' \) forms the next group. In this way, the grouping of sequences of symbols is done till \( X' \) is reached. The number of such groups obtained represents the direct descendants of \( X \), in the same order (each descendant is represented by the first symbol of each group belonging to \( L \)). The groupings obtained, form the first level of partition of the concerned frontier of the derivation tree.

**Example:**

\[
S'(A'X_1X_2A')B'(X_3X_4B')S'
\]

is grouped as

\[
S'[A'(X_1X_2A')][B'(X_3X_4B')]S'
\]

The tree corresponding to this is shown below:

Thus there are two direct descendants from node \( S \). Omitting the symbols belonging to \( L \cup R \), we can consider the string \( X_1X_2X_3X_4 \) to be partitioned to two groups consisting of \( X_1X_2 \) and \( X_3X_4 \) respectively.
Now, in the same way partition of each of the groups is obtained.

**Note 2:** A terminating rule expressed in the bracket language will have the form $A^i b_1, \ldots, b_n A^j$; hence no such grouping is possible.

**Step 4:** Since the homomorphic image of the frontier of the derivation tree is in correspondence with the homomorphic image of the right hand side of the $i$th rule in $G_1$, the right-hand side of the $i$th rule can be partitioned in an identical way ignoring the terminal sequence.

E.g., Let $X_1 X_2 X_3 X_4$ be the frontier of the derivation tree grouped as $X_1 X_2 X_3 X_4$. Let the right-hand side of the $i$th rule in $G_1$ be $a = Y_1 Y_2 Y_3 Y_4$. The corresponding grouping in this case is $Y_1 Y_2 Y_3 Y_4$.

Let the translation associated with this be given by $i : A \to a, \beta$ (belonging to $G_3$). Now group the symbols of $\beta$ corresponding to the groups of $a$, level by level using $(h_1)$. This is illustrated through an example.

\[
\begin{align*}
\text{E.g., } & h_1(a) = X_1 X_2 X_3 X_4 X_5 X_6 \\
& h_1(\beta) = X_6 X_5 X_3 X_2 X_1 X_4
\end{align*}
\]

$a$ consists of two level groupings.

First level of groupings of $h_1(a)$: $X_1 X_2 X_3 X_4 X_5 X_6$

Corresponding groupings of $h_1(\beta)$: $X_6 X_5 X_3 X_2 X_1 X_4$

Second level of groupings of $h_1(a)$: $X_1 X_2 X_3 X_4 X_5 X_6$

Corresponding groupings of $h_1(\beta)$: $X_6 X_5 X_3 X_2 X_1 X_4$

If such a partition of $\beta$ is not possible at any of the grouping of $a$, go to step 7; otherwise the translation is associated with the rules of $G_2$, corresponding to the translation associated with the $i$th rule of $G_3$, in the following way:

Let there be $r$ first level groups in $a$. Therefore there will be $r$ first level groups in the frontier of the corresponding derivation tree. Let the groupings be represented by the first element of each group belonging to $L$ viz., the grouping of $a$ is represented by $w_1 A_1 w_2 A_2, \ldots, A_r w_{r+1}$, where $A_i$ represent the groupings $1 \leq i \leq r$ and the frontier is denoted by

\[
X_k w_1 A_1 w_2 A_2 w_3 A_3 w_4 A_4 \cdots w_r A_r w_{r+1} X_k
\]

where $X_k \in V_{N_1}$ and is the root of this tree, $\gamma_i, 1 \leq i \leq r$ belongs to $(V_{N_1} \cup V_r)^*$. Now, by the definition of translation, $\gamma$ can be grouped* as,

* $p$ is a permutation mapping.
SDT for context-free grammars

\[ \bar{w}_1 \bar{A}_{p(1)} \bar{w}_2 \bar{A}_{p(2)} \ldots \bar{A}_{p(r)} \bar{w}_{r+1} \text{, where } \bar{w}_i, 1 \leq i \leq r+1 \text{ belongs to } (V_{N_1} \cup V_T)^* \text{ and the groups } \bar{A}_{p(i)} \text{, } 1 \leq i \leq r \text{, consists of sequences over } (V_{N_1} \cup V_T)^* \text{ [since the lengths of symbols in } L \cup R \text{ are considered as } 0, \text{ we write } (V_{N_1} \cup V_T)^* \text{ instead of } (V_{N_1} \cup V_T)^*], \text{ and consists of the same nonterminals that occur in the group represented by } A_{i}, 1 \leq i \leq r \text{. Construct } \bar{A}'_{p(i)} \text{, } 1 \leq i \leq r \text{, corresponding to } \bar{A}_{p(i)} \text{, } 1 \leq i \leq r \text{ so that } \bar{A}'_{p(i)} \text{ consists of sequences of } (V_{N_2} \cup V_T)^* \text{ after renaming of the nonterminals by the corresponding mapping used for the elements } M_i. \text{ Now associate the translation,} \]

\[ \bar{w}_1 \bar{A}'_{p(1)} \ldots \bar{A}'_{p(r)} \bar{w}_{r+1} \text{ with the node } X_k \text{ (thus with the associated rule } i_k). \text{ Repeat this for each of the groups } A_i, 1 \leq i \leq r. \]

Step 5: If the given SDTS with underlying grammar \( G_1 \) has to be covered by a semantically unambiguous SDTS with underlying grammar \( G_2 \), then go to step 6(a); otherwise go to step 6(b).

Step 6(a): Find whether the translations associated with the rules are ambiguous (i.e., there are translations of the form \( A \rightarrow a, \beta; A \rightarrow a, \gamma \)); if so, go to step 7; otherwise go to step 6(b).

Step 6(b): \( n = n+1 \); if \( n \leq n_i \), go to step 2; otherwise go to step 8.

Step 7: The given SDTS with underlying grammar \( G_1 \) cannot be covered by any SDTS with underlying grammar \( G_2 \); halt.

Step 8: \( i = i+1 \); if \( i \leq m \), go to step 1(b); otherwise the SDTS obtained with \( G_2 \) as underlying grammar is the one required, i.e., the obtained SDTS with \( G_2 \) covers the given SDTS with underlying grammar \( G_1 \); stop.

Acknowledgement

The author is thankful to the referee for his suggestions for improving the clarity of the paper.

References
