Theoretical Computer Science 17 (1982) 43–54
North-Holland Publishing Company

STRUCTURE PRESERVING ELIMINATION OF NULL PRODUCTIONS FROM CONTEXT-FREE GRAMMARS*

Esko UKKONEN
Department of Computer Science, University of Helsinki, Finland

Communicated by A. Salomaa
Received March 1979
Revised September 1980

Abstract. A method is presented for the elimination of null productions from a context-free grammar in such a way that the resulting grammar covers the original one, that is, the languages generated are the same and parses in the original grammar are homomorphic images of equivalent parses in the covering grammar. We show that in a natural subclass of covers, called the class of compatible covers, the method is best possible in the sense that a compatible cover with no null productions can be produced by the method if and only if such a cover exists. The use of the method in transformations for obtaining a cover in Greibach normal form is finally analyzed

1. Introduction

A context-free grammar is said to cover another grammar if both grammars generate the same language and if the parses in the covered grammar are homomorphic images of parses in the covering grammar [1, 2, 5]. This covering concept has its motivation in the field parsing: a parser for the covering grammar (this grammar may be in some 'easily' parsable subclass of context-free grammars) can be modified to a parser for the covered grammar by combining the calculation of the cover homomorphism with the output function of the parser.

This paper is concerned with transformations for producing covering grammars in certain normal forms. Our main result is a transformation for the elimination of the null productions (ε-productions) from a context-free grammar in such a way that the resulting ε-free grammar is a cover. The standard method for such an elimination [1] does not produce a cover.

It turns out that our transformation is not completely as general as possible because there are some grammars which clearly have an ε-free cover but for which our transformation does not work. In such cases the ε-free cover always has, however, from the practical point of view the undesirable feature that a parse in the covering grammar and the corresponding covered parse in the original grammar may consume the input string in entirely different ways. To exclude such pathological

* Some of the results of this paper were presented at the 4th GI Conference on Theoretical Computer Science, Aachen, March 26–28, 1979.

0304-3975/82/0000–0000/$02.75

covers we will define a natural subclass of covers which we call the compatible covers. We show that for producing a compatible ε-free cover our method is as general as possible. In the rest of the paper we briefly analyze some combined transformations which use our new method and some well-known methods for obtaining a compatible cover in Greibach normal form.

2. Notation and preliminary definitions

Our notations are mainly as in [1]. All definitions and notation not explicitly given are as in this reference.

A *(context-free) grammar* is denoted by a quadruple $G=(N, \Sigma, P, S)$ where N, Σ, P and S denote the nonterminals, the terminals, the productions and the start symbol of G, respectively. Unless otherwise stated we use the notational conventions that the letters A, B, C denote nonterminals in N, the letters a, b, c denote terminals in Σ, and the letters X, Y, Z denote members in $N \cup \Sigma$. Terminal strings in Σ^* are denoted by $u, v, \ldots, z$, and strings in $(N \cup \Sigma)^*$ by $\alpha, \beta, \gamma, \delta$, and the empty string by ε. A production of the form $A \rightarrow \varepsilon$ is called a *null production* or an *ε-production*. The length of a string s is denoted by $\lg(s)$.

Derivations in G are defined in the normal way. For all α, β in $(N \cup \Sigma)^*$, a *derivation step* in G is denoted by $\alpha A\beta \Rightarrow_G \alpha\gamma\beta$ if $A \rightarrow \gamma$ is in P. In particular, we write $\alpha A\beta \Rightarrow_{G,L} \alpha\gamma\beta$ for a *leftmost derivation step* and $\alpha A\beta \Rightarrow_{G,R} \alpha\gamma\beta$ for a *rightmost derivation step* in G, if α or β is in Σ^*, respectively. If i is the reference number of the production $A \rightarrow \gamma$, denoted $i.\, A \rightarrow \gamma$, we also write $\alpha A\beta \Rightarrow^{i}_{G,L} \alpha\gamma\beta$ or $\alpha A\beta \Rightarrow^{i}_{G,R} \alpha\gamma\beta$. A *leftmost derivation* $\alpha_1 \Rightarrow^{i_1}_{G,R} \alpha_2 \Rightarrow^{i_2}_{G,L} \cdots \Rightarrow^{i_{n-1}}_{G,L} \alpha_n$ is abbreviated by $\alpha_1 \Rightarrow^{\pi}_{G,L} \alpha_n$ where $\pi = i_1 i_2 \cdots i_{n-1}$. Notation $\alpha_1 \Rightarrow^{\pi}_{G,R} \alpha_n$ for a *rightmost derivation* is analogous. If $S \Rightarrow^{\pi}_{G,L} \alpha$, then π is a *left parse* (*l-parse*) of α in G, and if $S \Rightarrow^{\pi}_{G,R} \alpha$, then the reverse of π, denoted $\bar{\pi}$, is a *right parse* (*$\bar{r}$-parse*) of α in G.

We let $\Rightarrow^*_G, \Rightarrow^*_{G,L}, \Rightarrow^*_{G,R}$ represent the reflexive-transitive closures, and $\Rightarrow^+_G, \Rightarrow^+_{G,L}, \Rightarrow^+_{G,R}$ represent the transitive closures of relations $\Rightarrow_G, \Rightarrow_{G,L}, \Rightarrow_{G,R}$, respectively. In all these notations we may omit the subscript G when it is clear from the context.

The *language* generated by G is $L(G)=\{w \in \Sigma^* \mid S \Rightarrow^*_G w\}$. Similarly, for a string $\alpha \in (N \cup \Sigma)^*$ the language is $L(\alpha)=\{w \in \Sigma^* \mid \alpha \Rightarrow^*_G w\}$. The *degree of ambiguity* of a string $w \in L(G)$ is the number of its different left parses in G. When the degree of ambiguity equals one, w is *unambiguous*. If $A \Rightarrow^+_G A\beta$ or $A \Rightarrow^+_G \beta A$ for some $A \in N$, $\beta \in (N \cup \Sigma)^*$, then G is said to be *left-recursive* or *right-recursive*, respectively.

Grammar G is *cycle-free* if there is no derivation of the form $A \Rightarrow^+_G A$ for any A in N, and G is *ε-free* if it has no ε-productions except possibly the production $S \rightarrow \varepsilon$ in which case S does not appear on the right-hand side of any production of G. Finally, grammar G is in *Greibach normal form* if G is ε-free and each non-ε-production of G is of the form $A \rightarrow a\alpha$ with A in N, a in Σ, α in N^*.

All grammars in this paper are *reduced*, that is, every nonterminal of a grammar is used in some derivation of a terminal string from the start symbol.

Let us now recall the grammatical covering concepts usually called the right, left, right-to-left and left-to-right covering ($\bar{r}|\bar{r}$-covering, $l|l$-covering, $\bar{r}|l$-covering, $l|\bar{r}$-covering, for short). Different versions of the following definition have appeared e.g. in [1, 2, 5].

Let $G = (N, \Sigma, P, S)$, $G' = (N', \Sigma, P', S')$ be grammars, and let $x|y$ denote $\bar{r}|\bar{r}$, $l|l$, $\bar{r}|l$, or $l|\bar{r}$. Grammar G' is said to *$x|y$-cover* grammar G if there is a homomorphism $h: P'^* \to P^*$ such that for all strings w in Σ^*:

(i) if π' is an x-parse of w in G', $h(\pi')$ is a y-parse of w in G, and

(ii) if π is a y-parse of w in G, there exists an x-parse π' of w in G' such that $h(\pi') = \pi$.

Grammar G' *faithfully* $x|y$-covers grammar G if it $x|y$-covers G and moreover, in condition (ii) the x-parse π' always is unique for each y-parse π [4]. Thus a faithful cover preserves the degree of ambiguity.

It follows immediately from the definition that if G' $x|y$-covers G, then $L(G') = L(G)$. In addition, the cover relations are transitive: if G'' $x|y$-covers G' and G' $y|z$-covers G, then G'' $x|z$-covers G.

3. Elimination of the null productions

The standard elimination method [1] of the null productions from an arbitrary context-free grammar does not always produce a cover. It is, however, possible to modify the method such that it yields a cover. The class of grammars for which the new method is valid is now a proper subclass of context-free grammars.

Definition 1. Let $G = (N, \Sigma, P, S)$ be a grammar such that

(i) ε is unambiguous in G, and

(ii) G has no nonterminal A such that $A \Rightarrow_G^* \beta A \alpha$ where α is nonempty and $\alpha \Rightarrow_G^* \varepsilon$.

An ε-free grammar $G' = (N', \Sigma, P', S')$ and a homomorphism $h: P'^* \to P^*$ are defined for G in the following steps (1)–(3). Productions in P' are given in the form $A \to \alpha\langle\pi\rangle$ where $A \to \alpha$ is the production and $\pi \in P^*$ its image under h. If $\langle\pi\rangle$ is missing, the image is the empty string. Set N' will contain, besides S', some elements of the form $[\beta X \gamma]$ where β, γ in N^* are such that $\varepsilon \in L(\beta\gamma)$ and X in $N \cup \Sigma$ is such that $L(X) \neq \{\varepsilon\}$. For uniqueness, we write $[\beta \mathbf{X} \gamma]$ instead of $[\beta X \gamma]$.

(1) Initially, N' contains the new start symbol S'. If $S \Rightarrow_{G,L}^{\pi} \varepsilon$, add $S' \to \varepsilon\langle\pi\rangle$ to P'. If G does not generate a nonempty string, the method terminates. Otherwise add $S' \to [\mathbf{S}]$ to P' and $[\mathbf{S}]$ to N', and then repeat steps (2)–(3) until no changes are possible.

(2) For each element $[\mathbf{A}\gamma]$ in N' and for each production $j.\ A \to \alpha$ in P such that $L(\alpha) \neq \{\varepsilon\}$, add to P' all productions constructed as follows. Suppose α has a representation $\alpha = \alpha_1 X_1 \alpha_2 X_2 \cdots \alpha_n X_n \alpha_{n+1}$, $n > 0$, where $X_i \in N \cup \Sigma$ such that $L(X_i) \neq \{\varepsilon\}$, $i = 1, \ldots, n$, and $\alpha_i \in N^*$ such that $\varepsilon \in L(\alpha_i)$, $i = 1, \ldots, n+1$. For each

such a representation of α, add to P' the production $[A\gamma] \to Y_1 \cdots Y_{n-1} Y_n\langle j\rangle$ where for $i = 1, \ldots, n-1$

$$Y_i = \begin{cases} [\alpha_i X_i], & \text{if } \alpha_i \neq \varepsilon \text{ or } X_i \in N, \\ X_i, & \text{otherwise} \end{cases}$$

and

$$Y_n = \begin{cases} [\alpha_n X_n \alpha_{n+1}\gamma], & \text{if } \alpha_n\alpha_{n+1}\gamma \neq \varepsilon \text{ or } X_n \in N, \\ X_n, & \text{otherwise.} \end{cases}$$

Add to N' those elements Y_i which are not in Σ.

(3) Let $j\,.\,B \to \beta$ in P be a production such that $\varepsilon \in L(\beta)$. Then for each nonterminal $[B\alpha X\gamma]$ in N' where $\alpha, \gamma \in N^*$ and $X \in N \cup \Sigma$, add the production $[B\alpha X\gamma] \to [\beta\alpha X\gamma]\langle j\rangle$ to P' and the nonterminal $[\beta\alpha X\gamma]$ to N' if $\beta\alpha\gamma \neq \varepsilon$ or $X \in N$. Otherwise, that is, if $\beta\alpha X\gamma = X \in \Sigma$, add to P' the production $[B\alpha X\gamma] \to X\langle j\rangle$. Similarly, for each nonterminal $[XB\gamma]$ in N' where $X \in \Sigma$ and $\gamma \in N^*$, add the production $[XB\gamma] \to [X\beta\gamma]\langle j\rangle$ to P' and the nonterminal $[X\beta\gamma]$ to N', if $\beta\gamma \neq \varepsilon$. Otherwise, that is, if $X\beta\gamma = X \in \Sigma$, add the production $[XB\gamma] \to X\langle j\rangle$ to P'.

It is easy to test whether G satisfies the input conditions (i) and (ii) of the above method. Condition (i) can be tested in the same procedure which constructs a leftmost derivation π of ε for step (1). Note that the homomorphism h is uniquely defined in the method if and only if condition (i) is satisfied. Condition (ii) can be tested when new elements are added to N'. It is easily seen that condition (ii) is true if and only if in every element $[\alpha X\gamma]$ added to N' the length of α and γ is $\leq(\mu-1)(\nu-1)$ where μ is the number of the nonterminals of G and ν is the maximum length of the right-hand sides of the productions of G. If the length of α or γ is $>(\mu-1)(\nu-1)$, the number of elements added to N' is unlimited and the computation does not halt. Hence we have that the method of Definition 1 halts and produces a grammar G' (which is ε-free by the construction) with a uniquely defined homomorphism h if and only if the original grammar G satisfies conditions (i) and (ii).

We now work out an example. Applied to a grammar with productions

1. $S \to BAL$	4. $B \to LB$
2. $L \to \varepsilon$	5. $B \to \varepsilon$
3. $A \to aL$	6. $B \to b$

the transformation method of Definition 1 yields a grammar with productions

$[S] = [B][AL]\langle 1\rangle$	$[aLL] \to [aL]\langle 2\rangle$
$[S] \to [BAL]\langle 1\rangle$	$[aL] \to a\langle 2\rangle$
$[B] \to [LB]\langle 4\rangle$	$[BAL] \to [AL]\langle 5\rangle$
$[B] \to b\langle 6\rangle$	$[BAL] \to [LBAL]\langle 4\rangle$
$[LB] \to [B]\langle 2\rangle$	$[LBAL] \to [BAL]\langle 2\rangle$
$[AL] \to [aLL]\langle 3\rangle$.	

To analyze the time complexity of the method in Definition 1 consider a grammar G with productions

$$S \to A_1 A_2 \cdots A_k,$$

$$A_i \to a_i \mid \varepsilon \quad \text{for all } i,\ 1 \leq i \leq k.$$

The size of G, that is, the sum of the lengths of all strings $A\alpha$ such that $A \to \alpha$ is a production, is linear in k. It is easily seen that applied to G, our method gives a grammar with size exponential in k. Thus the running time of the method may be exponential in the size of the input.

Theorem 2. *The method of Definition 1 produces an ε-free grammar G' which faithfully left covers the given grammar G for every G such that*

(i) *ε is unambiguous in G, and*

(ii) *G has no nonterminal A such that $A \Rightarrow_G^* \beta A \alpha$ where α is nonempty and $\alpha \Rightarrow_G^* \varepsilon$.*

Proof. Noting the discussion after Definition 1, it remains for us to prove that G' faithfully left covers G under the homomorphism h.

To see that every left derivation in G' covers some left derivation in G, let $[\beta X\gamma]$ be a nonterminal of G', and $[\beta X\gamma] \Rightarrow_{G',\mathrm{L}}^{\pi} w$ for some nonempty $w \in \Sigma^*$. By a straightforward induction on the length of π it follows that $\beta X\gamma \Rightarrow_{G,\mathrm{L}}^{h(\pi)} w$ (and here w is derived from X). Consequently, if $S' \Rightarrow_{G',\mathrm{L}} [S] \Rightarrow_{G',\mathrm{L}}^{\pi} w$, then $S \Rightarrow_{G,\mathrm{L}}^{h(\pi)} w$. Moreover, if $\varepsilon \in L(G')$, G' has a production $i\ .\ S' \to \varepsilon$ and $S \Rightarrow_{G,\mathrm{L}}^{h(i)} \varepsilon$ by the construction of G'. Thus G' satisfies the first part of the definition of the left cover.

Next we verify the second part and show that the cover is faithful. Let again $[\beta X\gamma]$ be a nonterminal of G' and $\beta X\gamma \Rightarrow_{G,\mathrm{L}}^{\pi} w$ where X derives w. We show by induction on $\lg(\pi)$ that there is a unique covering derivation $[\beta X\gamma] \Rightarrow_{G',\mathrm{L}}^{\pi'} w$ such that $h(\pi') = \pi$. Then it follows for $[\beta X\gamma] = [S]$ that every derivation $S \Rightarrow_{G,\mathrm{L}}^{\pi} w$, $w \neq \varepsilon$, has a unique covering derivation starting from $[S]$ and hence, from S'. For $w = \varepsilon$ the assertion is true immediately by the construction.

To begin the induction, let $\lg(\pi) = 1$. This implies that $[\beta X\gamma]$ is $[Aa]$, $[aA]$ or $[A]$ for some $A \in N$, $a \in \Sigma$. If $[\beta X\gamma] = [Aa]$ or $[aA]$ then $w = a$ and the production π of G must be $A \to \varepsilon$. Then by the construction, $\pi'\ .\ [Aa] \to a\langle\pi\rangle$ or respectively, $\pi'\ .\ [aA] \to a\langle\pi\rangle$ is the only production of G' with the left-hand side $[Aa]$ or $[aA]$ such that $h(\pi') = \pi$. In the remaining case $[\beta X\gamma] = [A]$. Then the production π must be $A \to w$. Now $\pi'\ .\ [A] \to w\langle\pi\rangle$ is the only production of G' with the left-hand side $[A]$ such that $h(\pi') = \pi$.

Let then $\lg(\pi) > 1$. Write $\pi = \pi_1\omega$ where π_1 is the first element of π. We again have three cases depending on whether the leftmost nonterminal of G in the string $\beta X\gamma$ appears in β, X, or γ. If the first nonterminal is in β, that is, $[\beta X\gamma] = [B\beta' X\gamma]$ for some $B \in N$, $\beta' \in N^*$, then the production π_1 of G must be of the form $B \to \beta''$ for some β''. Then by the construction, $\pi_1'\ .\ [B\beta' X\gamma] \to [\beta''\beta' X\gamma]\langle\pi_1\rangle$ is the only production of G with left-hand side $[B\beta' X\gamma]$ such that $h(\pi_1') = \pi_1$. Because

$\beta''\beta'X\gamma \Rightarrow^{\omega}_{G,L} w$ and $\lg(\omega) < \lg(\pi)$, there is by induction hypothesis a unique covering derivation $[\beta''\beta'X\gamma] \Rightarrow^{\omega'}_{G',L} w$ where $h(\omega') = \omega$. Thus $h(\pi_1'\omega') = \pi$ and $\pi_1'\omega'$ is uniquely determined by π and $[\beta X\gamma]$, as required. The case in which the leftmost nonterminal of $\beta X\gamma$ is in γ can be considered analogously.

It remains the case $[\beta X\gamma] = [A\gamma]$ for some $A \in N$. Then the production π_1 of G must be of the form $A \to \alpha$ for some nonempty $\alpha \in (N \cup \Sigma)^*$ and therefore, $A\gamma \Rightarrow^{\pi_1}_{G,L} \alpha\gamma \Rightarrow^{\omega}_{G,L} w$. Now it is possible to represent α as $\alpha = \alpha_1 X_1 \alpha_2 X_2 \cdots \alpha_n X_n \alpha_{n+1}$ where strings α_i and letters X_i are chosen so that the subderivations of ω applied to each α_i derive ε, and the subderivations applied to each X_i derive a nonempty string w_i (and hence $w = w_1 w_2 \cdots w_n$). Then G' must have a production π_1'. $[A\gamma] \to Y_1 Y_2 \cdots Y_n \langle \pi_1 \rangle$ constructed from this representation of α in step (2) of Def. 1. By induction hypothesis each derivation $\alpha_1 X_1 \Rightarrow^{\omega_1}_{G,L} w_1, \ldots, \alpha_{n-1} X_{n-1} \Rightarrow^{\omega_{n-1}}_{G,L} w_{n-1}$, $\alpha_n X_n \alpha_{n+1}\gamma \Rightarrow^{\omega_n}_{G,L} w_n$ where $\omega_1 \cdots \omega_n = \omega$, has a unique covering derivation starting from $Y_1, \ldots, Y_n$, respectively. Combined with π_1' these derivations give a leftmost derivation π' of w from $[A\gamma]$ such that $h(\pi') = \pi$.

To see that π' is uniquely determined by π we must still show that π_1' is unique. Suppose we may use π_1''. $[A\gamma] \to Y_1' \cdots Y_{n'}' \langle \pi_1 \rangle$ instead of π_1', $\pi_1'' \neq \pi_1'$. This implies that π_1'' is also constructed from α but the representation used differs from that of π_1'. Let now $Y_1' \cdots Y_{n'}' \Rightarrow^{\omega''}_{G',L} w$. Then noting the first part of this proof, $\alpha_1 X_1 \cdots \alpha_n X_n \alpha_{n+1}\gamma \Rightarrow^{h(\omega'')}_{G,L} w$. The difference of representations for α in π_1' and π_1'' then implies that in derivation $h(\omega'')$ some α_i does not derive ε or some X_i does not derive a nonempty string. Hence $h(\omega'') \neq \omega$. Therefore $\pi_1'' = \pi_1'$. This completes the proof of the inductive step and the theorem.

The transformation of Definition 1 has also the following property.

Lemma 3. *If the transformed grammar $G' = (N', \Sigma, P', S')$ produced by the method of Definition 1 is left recursive, then also the original grammar $G = (N, \Sigma, P, S)$ is left recursive.*

Proof. For a left recursive G' we have $[\beta X\gamma] \Rightarrow^{\pi}_{G',L} [\beta X\gamma]\delta$ for some $[\beta X\gamma] \in N'$, $\delta \in (N' \cup \Sigma)^*$ and nonempty π. Then

$$\beta X\gamma \Rightarrow^{h(\pi)}_{G,L} \beta X\gamma\delta' \tag{1}$$

where $\beta X\gamma\delta' \in (N \cup \Sigma)^+$. Let A be the leftmost (and only!) nonterminal of G in $\beta X\gamma$ such that the subderivation of (1) starting from A does not derive ε. Hence we may represent $[\beta X\gamma]$ as $[\alpha A\beta' X\gamma]$ or $[a\alpha A\gamma']$ or $[\alpha A\gamma]$ and derivation (1) as $\alpha A\beta' X\gamma \Rightarrow^{\sigma}_{G,L} A\beta' X\gamma \Rightarrow^{\tau}_{G,L} \alpha A\beta' X\gamma$ or $a\alpha A\gamma' \Rightarrow^{\sigma}_{G,L} aA\gamma' \Rightarrow^{\tau}_{G,L} a\alpha A\gamma'$ or $\alpha A\gamma \Rightarrow^{\sigma}_{G,L} A\gamma \Rightarrow^{\tau}_{G,L} \alpha A\gamma\gamma'\gamma$, respectively, where $\sigma\tau = h(\pi)$ and $\alpha \Rightarrow^{\sigma}_{G,L} \varepsilon$. The first two derivations then imply that $A \Rightarrow^{\tau}_{G,L} \alpha A \Rightarrow^{\sigma}_{G,L} A$, and the last that $A \Rightarrow^{\tau}_{G,L} \alpha A\gamma\gamma' \Rightarrow^{\sigma}_{G,L} A\gamma\gamma'$. Hence G is left recursive because A is a left recursive nonterminal.

We now turn our attention to those grammars which do not satisfy condition (i) or (ii) of Theorem 2 and for which an ε-free left cover cannot therefore be produced by the method of Definition 1. By the definition of an ε-free grammar, a grammar not satisfying condition (i) clearly has no ε-free (left) cover. Condition (ii) is more interesting because there exists grammars which do not satisfy it but which nevertheless have an ε-free left cover. For example, a grammar G_1 with productions

1. $S \to 1SL$
2. $S \to 2$
3. $L \to \varepsilon$

does not satisfy (ii) but has, for instance, an ε-free left cover G_1' with productions

$S' \to 2\langle 2\rangle$

$S' \to S''2$

$S'' \to S''L'\langle 1\rangle$

$S'' \to 1\langle 2\rangle$

$L' \to 1\langle 3\rangle$

On the other hand, there are grammars not satisfying (ii) and not having an ε-free cover. It is shown in [9] that the following is such a grammar

$S \to 0SL \mid 0RL$

$R \to 1RL \mid 1$

$L \to \varepsilon$

This means that the condition that precisely characterizes the ε-free left coverability of a context-free grammar must be slightly weaker than the condition (ii) of Theorem 2. Laufkötter [3] succeeds in giving such a characterization but only at the price of using rather unnatural covers. Therefore we will introduce in the next section an additional restriction on covers, and this condition is hoped to isolate a more natural subclass of covers. For such covers, the method of Definition 1 is as general as possible.

4. Compatible covers

In a grammar G, let

$$S \Rightarrow^{\pi}_{G,L} c\gamma\alpha \Rightarrow^{*}_{G,L} xy$$

where γ is produced by the last production of π. Then the terminal string x is called the *consumed input* for the left parse π. Similarly, if

$$S \Rightarrow^{*}_{G,R} \alpha y \Rightarrow^{\pi}_{G,R} xy$$

where α either ends with a nonterminal or is ε, then the terminal string x is called the *consumed input* for the right parse $\tilde{\pi}$. Note that in terms of parsing the consumed input is approximately the same as the input string read by usual parsing algorithms up to the time of announcing π.

In the following definition we restrict ourselves to left covers; the other cases are similar. Assume that G' left covers G under homomorphism h. Grammar G' is said to be a *compatible* left cover of G if there is an integer $k \geq 0$ such that whenever

$$S' \Rightarrow^{\pi}_{G',L} x'\alpha' \quad \text{and} \quad S \Rightarrow^{h(\pi)}_{G,L} x\alpha$$

where x' is the consumed input for π and x for $h(\pi)$, then

$$|\lg(x) - \lg(x')| \leq k.$$

This definition can be motivated by the following remark. Suppose that we want to process strings in language $L(G)$ in some syntax-directed way using, say, left parses in G. For some reason, however, we must do the actual parsing in grammar G'. Since G' left covers G, the parser for G' is easily augmented to produce left parses in G. If in this situation the cover is not compatible, it is practically impossible to organize the syntax-directed processing in a natural fashion because incompatibility means that the parses in G and G' consume the input in very different ways. As an extreme example consider a grammar with productions

1. $E \to E + i$
2. $E \to i$

and its incompatible left cover with productions

$$E' \to i + E'\langle 1 \rangle$$

$$E' \to i\langle 2 \rangle$$

A typical left parser (say, an $LL(2)$ parser) for the covering grammar recognizes production $E' \to i + E'$ for the first time when reading the first $i+$ of the input. If the parser has been augmented to produce parses in the covered grammar, it should at this moment announce the production 1. $E \to E + i$ of the covered grammar. This in turn activates the syntax-directed processing associated with production 1. But now a difficulty arises if the processing refers to the i in production 1 because this i denotes the last i in the input string while the parser has read no more than the first i of the input.

It is an easy inductive exercise to prove:

Lemma 4. *The left cover produced by the method of Definition 1 is compatible.*

Note that the left cover G_1' for our example grammar G_1 at the end of Section 3 is not compatible.

Theorem 5. *A grammar G has an ε-free left cover that is compatible if and only if G satisfies conditions* (i) *and* (ii) *of Theorem* 2. *Thus for producing a compatible ε-free left cover the method of Definition* 1 *is as general as possible.*

Proof. Theorem 2 and Lemma 4 establish the sufficiency of conditions (i) and (ii).

Conversely, suppose that (i) or (ii) is not true. If (i) is not true, then an ε-free cover, compatible or not, trivially cannot exist. So it remains for us to be proved that a compatible ε-free left cover of G cannot exist if (ii) is not true. Suppose that G has a nonterminal A such that $A \Rightarrow_G^* \beta A\alpha$ where $\alpha \neq \varepsilon$ and $\alpha \Rightarrow_G^* \varepsilon$. Then we may assume that, for some π,

$$A \Rightarrow_{G,\mathrm{L}}^{\pi} xA\alpha$$

where x is a terminal string.

To derive a contradiction suppose that G' is a compatible ε-free left cover of G with compatibility constant k. In G we have derivations

$$S \Rightarrow_{G,\mathrm{L}}^{\mu} uA\gamma \Rightarrow_{G,\mathrm{L}}^{\pi^i} ux^iA\alpha^i\gamma \Rightarrow_{G,\mathrm{L}}^{\nu_i} ux^iw$$

for every $i>0$ and for some u, w in Σ^*, γ in $(\Sigma \cup N)^*$. Derivation ν_i is here of the form $\nu_i = \eta\psi^i\kappa$ where each ψ is applied on the corresponding α to produce ε, and η is applied on A and κ on γ to produce w.

Then G' must have covering derivations of the form

$$S' \Rightarrow_{G',\mathrm{L}}^{\sigma_i} y_i\delta_i \Rightarrow_{G',\mathrm{L}}^{\tau_i} ux^iw$$

such that $h(\sigma_i) = \mu\pi^i$, $h(\tau_i) = \nu_i$ and y_i is the consumed input for σ_i. We then have $|\mathrm{lg}(ux^i) - \mathrm{lg}(y_i)| \leqslant k$ because of compatibility. This implies

$$\mathrm{lg}(\delta_i) = \mathrm{lg}(y_i\delta_i) - \mathrm{lg}(y_i) \leqslant \mathrm{lg}(ux^iw) - \mathrm{lg}(y_i) \leqslant \mathrm{lg}(w) + k$$

where the first inequality follows from the ε-freeness of G'. But such bounded strings δ_i must be identical for infinitely many indices i. Hence we may choose i_1 and i_2 such that $\delta_{i_1} = \delta_{i_2}$ and $i_2 > \mathrm{lg}(\eta) + i_1 \cdot \mathrm{lg}(\psi) + \mathrm{lg}(\kappa)$.

Then we have in G' a derivation

$$S' \Rightarrow_{G',\mathrm{L}}^{\sigma_{i_2}} y_{i_2}\delta_{i_2} = y_{i_2}\delta_{i_1} \Rightarrow_{G',\mathrm{L}}^{\tau_{i_1}} z$$

where the result z is clearly a terminal string. Because G' covers G and $h(\sigma_{i_2}) = \mu\pi^{i_2}$ and $h(\tau_{i_1}) = \eta\psi^{i_1}\kappa = \nu_{i_1}$, the corresponding covered derivation of z in G should be of the form

$$S \Rightarrow_{G,\mathrm{L}}^{\mu\pi^{i_2}} ux^{i_2}A\alpha^{i_2}\gamma \Rightarrow_{G,\mathrm{L}}^{\nu_{i_1}} \beta$$

where $\beta = z$. But β cannot be a terminal string because by the choice of i_1 and i_2, $\mathrm{lg}(\nu_{i_1}) = \mathrm{lg}(\eta) + i_1 \cdot \mathrm{lg}(\psi) + \mathrm{lg}(\kappa) < i_2$ and there are more than i_2 nonterminals in $A\alpha^{i_2}\gamma$. Therefore $\beta \neq z$. This contradiction completes our proof.

Next we consider some simple modifications of the method of Definition 1 which can be used in obtaining compatible ε-free right, left-to-right or right-to-left covers.

First, it is clear that a left-to-right symmetric method to the method of Definition 1 produces ε-free right covers that are faithful and compatible. The method is again valid for all grammars G that have such a cover, that is, for grammars in which ε is unambiguous and there is no nonterminal A such that $A \Rightarrow_G^* \alpha A \beta$ where α is nonempty and $\alpha \Rightarrow_G^* \varepsilon$.

To produce a left-to-right cover we first apply a well-known marking technique. We construct for a given grammar $G = (N, \Sigma, P, S)$ a new grammar such that at the end of the right-hand side of each production a new nonterminal producing only the empty string is added as an identification of the production. Formally, the new grammar is $G'' = (N \cup P, \Sigma, P'', S)$ where $P'' = \{A \to \alpha(A, \alpha) \mid A \to \alpha$ is in $P\} \cup \{(A, \alpha) \to \varepsilon \mid A \to \alpha$ is in $P\}$. Define a homomorphism $h'': P''^* \to P^*$ by $h''(A \to \alpha(A, \alpha)) = \varepsilon$ and $h''((A, \alpha) \to \varepsilon) = i$ for all i . $A \to \alpha$ in P. Then it is obvious that G'' faithfully and compatibly right covers G under h''. But because only the ε-productions of G'' have a nonempty image under h'' and the ε-productions appear in the same relative order in both left and right parses, G'' left-to-right covers G under h'', too. If the method of Definition 1 is then applied to G'', an ε-free grammar G' is obtained which faithfully and compatibly left-to-right covers G. The conditions under which this method is valid are easily derived from Theorem 2 and the construction of G''. We have

Theorem 6. *There is a method for producing a faithful, compatible and ε-free left-to-right cover for every grammar G such that*
(i) *ε is unambiguous in G, and*
(ii) *G is not right recursive.*

The following theorem, which can be proved with similar techniques as Theorem 5, shows that for cycle-free grammars the method of Theorem 6 is again as general as possible.

Theorem 7. *Let G be a cycle-free grammar. Then G has an ε-free left-to-right cover that is compatible if and only if G satisfies conditions* (i) *and* (ii) *of Theorem* 6.

By symmetry, the counterparts of Theorems 6 and 7 for right-to-left covering are clearly true, too.

5. Other transformations

In [4; 5, Algorithm 5.1] an algorithm is given for producing a faithful left cover in Greibach normal form for every non-left-recursive and ε-free grammar G that satisfies the additional condition that if v is in $L(G)$ and $\lg(v) \leqslant 1$, then v is unambiguous in G. The cover produced by the algorithm is compatible. When this algorithm is applied on the result of the method of Definition 1 we get a combined transformation which, noting Lemma 3, has the following properties:

Proposition 8. *There is a method for producing a faithful and compatible left cover in Greibach normal form for every grammar G such that*

(i) *G is not left recursive,*

(ii) *every v in L(G) such that* $\lg(v) \leq 1$ *is unambiguous in G, and*

(iii) *G has no nonterminal A such that* $A \Rightarrow_G^* \beta A \alpha$ *where* α *is non-empty and* $\alpha \Rightarrow_G^* \varepsilon$.

It follows from the definition of Greibach normal form that for any transformation, the condition (ii) cannot be eliminated from Proposition 8. If we restrict ourselves to compatible covers then the elimination of conditions (i) and (iii) is impossible, too. For condition (iii) this follows from Theorem 5 and for condition (i) from the following lemma whose easy proof is left to the reader.

Lemma 9. *If grammar G is left recursive, then G has no compatible left cover in Greibach normal form.*

Consequently the method of Proposition 8 is as general as possible for producing a compatible left cover in Greibach normal form. If we want to construct a compatible cover in Greibach normal form for a left recursive grammar, it is therefore necessary to consider other cover types than the left covering. In this case the left-to-right covering is a natural cover type to be used.

To obtain a compatible left-to-right cover in Greibach normal form we may use the following strategy: eliminate first the left recursion using a method that produces a compatible left-to-right cover and then apply the method of Proposition 8. Several techniques are known for the elimination of the left recursion. Here we briefly consider the method originally proposed in [7] and further analyzed in [6, 8, 10]. This method produces for every context-free grammar G a left-to-right cover $T(G)$ that is faithful and compatible. Moreover, cover $T(G)$ is non-left-recursive if and only if G is cycle-free and there is in G no nonterminals A such that $A \Rightarrow_G^* \alpha A \beta$ where $\alpha \neq \varepsilon$ and $\alpha \Rightarrow_G^* \varepsilon$ [10]. By applying the method of Proposition 8 to $T(G)$ we get a combined transformation having properties symmetric with Proposition 8:

Proposition 10 ([10]). *There is a method for producing a faithful and compatible left-to-right cover in Greibach normal form for every grammar G such that*

(i) *G is not right recursive,*

(ii) *every v in L(G) such that* $\lg(v) \leq 1$ *is unambiguous in G, and*

(iii) *G has no nonterminals A such that* $A \Rightarrow_G^* \alpha A \beta$ *where* α *is non-empty and* $\alpha \Rightarrow_G^* \varepsilon$.

The proof that for producing a compatible left-to-right cover in Greibach normal form the method of Proposition 10 is again as general as possible, is left to the reader.

References

[1] A.V. Aho and J.D. Ullman, *The Theory of Parsing, Translation, and Compiling, Vol. I: Parsing* (Prentice-Hall, Englewood Cliffs, NJ, 1972).

[2] J.N. Gray and M.A. Harrison, On the covering and reduction problems for context-free grammars, *J. ACM* **19** (1972) 675–698.

[3] F. Laufkötter, Überdeckung Kontextfreier Grammatiken, Insbesondere ε-freie Überdeckungen Diplomarbeit, Rhein.-Westf. Techn. Hochschule Aachen (Dezember 1979).

[4] A. Nijholt, Grammar functors and covers: From non-left-recursive to Greibach normal form grammars, *BIT* **19** (1979) 73–78.

[5] A. Nijholt, *Context-free Grammars: Covers, Normal Forms, and Parsing*, Lecture Notes in Computer Science **93** (Springer, Berlin, 1980).

[6] E. Soisalon-Soininen, On the covering problem for left-recursive grammars, *Theoret. Comput. Sci.* **8** (1979) 1–12.

[7] E. Soisalon-Soininen and E. Ukkonen, A characterization of LL(k) languages, In: S. Michaelson and R. Milner, Eds, *Automata, Languages and Programming* (Edinburgh University Press, Edinburgh, 1976) 20–30.

[8] E. Soisalon-Soininen and E. Ukkonen, A method for transforming grammars into LL(k) form, *Acta Informat.* **12** (1979) 339–369.

[9] E. Ukkonen, The nonexistence of some covering context-free grammars, *Information Processing Lett.* **8** (1979) 187–192.

[10] E. Ukkonen, Notes on grammatical covering by context-free grammars in normal forms, Report C-1979-37, Department of Computer Science, University of Helsinki (1979).