Acta Informatica 14, 271-294 (1980)

Acta Informatica

A Survey of Normal Form Covers for Context Free Grammars

Anton Nijholt

Vrije Universiteit, Dept. of Mathematics, PO-Box 7161, Amsterdam, The Netherlands

Summary. An overview is given of cover results for normal forms of context-free grammars. The emphasis in this paper is on the possibility of constructing ε-free grammars, non-left-recursive grammars and grammars in Greibach normal form. Among others it is proved that any ε-free context-free grammar can be right covered with a context-free grammar in Greibach normal form.

All the cover results concerning the ε-free grammars, the non-left-recursive grammars and the grammars in Greibach normal form are listed, with respect to several types of covers, in a cover-table.

1. Introduction

We study the existence and nonexistence of *grammar covers* for some normal forms for context-free grammars. That is, we consider problems in which we ask: Given classes of grammars Γ_1 and Γ_2, can we find for each grammar G in Γ_1 a grammar G' in Γ_2 such that G' covers G?

For Γ_1 we will consider arbitrary context-free grammars. Moreover, by introducing some conditions which should be satisfied we consider also some subclasses of the context-free grammars. For Γ_2 we will concentrate on the *ε-free*, the *non-left-recursive* and the *Greibach normal form* grammars.

A context-free grammar G' is said to cover a context-free grammar G if it is possible to define a homomorphism between the parses of G' and those of G.

We will restrict ourselves to covers which are defined with the help of left and right parses of the grammars in question. It follows that we can define four types of covers, viz. we can define covers in such a way that left parses are mapped on left parses, left parses are mapped on right parses, right parses are mapped on left parses or right parses are mapped on right parses. For each of these covers we will present a *yes* or *no* answer to the question whether several types of context-free grammars can be covered by grammars in a certain normal form.

A variety of results in this research area have been obtained before (cf. Aho

0001-5903/80/0014/0271/$04.80

and Ullman [1], Gray and Harrison [6, 7], Nijholt [21, 22, 23, 24], Soisalon-Soininen [29] and Ukkonen [31-33 and unpublished]. The aim of this paper is to give a complete overview of the relevant cover results for normal forms of context-free grammars. That is, we collect some of the results in the above mentioned papers and we fill in the missing parts.

The concept of grammar cover can be considered as a grammatical similarity relation. Many other relations between grammars have been defined. For example, there is the concept of structural equivalence (Paull and Unger [27]), there is the grammar functor or X-functor approach, initiated by Hotz [10, 11], and there are the grammar forms introduced by Cremers and Ginsburg [3].

One motivation to consider these relations can just be the mathematical interest in comparing and relating different subclasses of the context-free grammars. Especially in the case of normal forms of context-free grammars it is natural to ask whether a grammar belonging to a certain class can be transformed to a grammar in a certain normal form and to determine which relations hold between the two grammars. Dependent on this relation one can then conclude that the transformation preserves certain properties of the original grammar.

For each similarity relation there are some obvious questions concerning decidability and complexity. In Hunt, Rosenkrantz and Szymanski [15, 16] decidability results for context-free grammars with respect to the grammar cover are presented. Among others it is shown that it is undecidable whether a context-free grammar G' covers a context-free grammar G. An overview of complexity results for grammatical similarity relations is given in Hunt and Rosenkrantz [14]. The second motivation to consider grammar covers is their proven usefulness in the theory and practice of parsing and compiler building. Immediately after the presentation of the cover definition we will return to this aspect.

The organization of this paper is as follows. After the presentation of some preliminaries there is a short section in which we discuss the grammar cover concept and how it has appeared, sometimes defined in an informal way, in the literature. In Sect. 2 we list some general theorems on the existence of covering grammars. New theorems and corresponding transformations on context-free grammars to produce grammars in Greibach normal form are also presented in this section.

As the main result of this section we consider that we are able to show that any ε-free context-free grammar can be transformed to a context-free grammar in Greibach normal form in such a way that a right cover (in this case right parses can be mapped on right parses) can be defined.

In Sect. 3 we present an adapted version of a grammar which is due to Ukkonen [33]. Together with the results and observations in Sect. 2 this grammar is sufficient to obtain all negative cover results which are relevant for the classes of grammars which we consider. The example in this section is chosen in such a way that some cover results for strict deterministic, $LL(k)$ and $LR(k)$ grammars become obvious.

Finally, in Sect. 4 a (cover-) table is constructed in which all the results are listed.

For a survey of normal form cover results for regular grammars the reader is referred to Nijholt [26]. A survey which includes results for $LL(k)$, $LR(k)$ and strict deterministic grammars is in preparation. In Mickunas [19], Mickunas, Lancaster and Schneider [20] and in Nijholt [25] other cover results for $LR(k)$ grammars can be found.

Results for the grammar functor approach for normal forms of context-free grammars can be found in Hotz [12] and Benson [2] and for the $LL(k)$ and $LR(k)$ grammars in Hotz and Ross [13] and in Ross, Hotz and Benson [28].

1.1. Preliminaries

We review various commonly known definitions (cf. Aho and Ullman [1]) and give some notations.

A *context-free grammar* (CFG) will be denoted with the usual fourtuple $G=(N, \Sigma, P, S)$, where N is the set of *nonterminal* symbols (generally denoted by the Roman capitals $A, B, C, \ldots$), Σ is the set of *terminal* symbols (denoted by the smalls $a, b, c \ldots$), $P \subseteq N \times (\Sigma \cup N)^*$ is the set of *productions* (we use the notation $A \to \alpha$ if $(A, \alpha) \in P$) and $S \in N$ is the *start symbol*. We define $V = N \cup \Sigma$. Elements of V will generally be denoted by X, Y and Z; elements of V^* by the Greek smalls $\alpha, \beta, \gamma, \ldots$ and elements of Σ^* by the smalls u, v, w, x, y and z. We have the usual notations $\Rightarrow$, $\overset{+}{\Rightarrow}$ and $\overset{*}{\Rightarrow}$ for *derivations* and we use indices L and R to denote *leftmost* and *rightmost* derivations, respectively. The *language* generated by G is the set $L(G) = \{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$.

The sequence of productions which are used in a leftmost derivation from S to a string $w \in \Sigma^*$ is called a *left parse* for w. The reverse of a sequence of productions in such a rightmost derivation is called a *right parse* for w.

If $\alpha \in V^*$ then α^R denotes the *reverse* of α and $|\alpha|$ denotes the *length* of α. The symbol ε is reserved for the *empty string* (the string with length zero). If $|\alpha| < k$ then $k: \alpha$ denotes α, otherwise $k: \alpha$ denotes the prefix of α with length k.

If Q is a set then $|Q|$ denotes the number of elements in Q. For each CFG $G=(N, \Sigma, P, S)$ we define $\Delta_G = \{i \mid 1 \leqq i \leqq |P|\}$, the set of *production numbers* of G. If $A \to \alpha$ is the ith production in P then we sometimes write $i \cdot A \to \alpha$. Moreover, we write $A \overset{\pi}{\Rightarrow} \alpha$, where $\pi \in \Delta_G^*$, if the derivation from A to α is done according to the sequence of productions π. Hence, if $S \underset{L}{\overset{\pi}{\Rightarrow}} w$ then π is a left parse for w and if $S \underset{R}{\overset{\pi}{\Rightarrow}} w$ then π^R is a right parse for w.

The *degree of ambiguity* of a sentence $w \in L(G)$ is the number of different left parses for w. Notation: $\langle w, G\rangle$. If for any $w \in L(G)$ we have $\langle w, G\rangle = 1$ then G is called *unambiguous*.

For any $A \in N$ we define $rhs(A) = \{\alpha \mid A \to \alpha \text{ is in } P\}$.

Definition 1.1. A CFG $G=(N, \Sigma, P, S)$ is said to be

a) *ε-free*, if $P \subseteq N \times V^+$.

b) *cycle-free*, if for any $A \in N$ a derivation $A \overset{+}{\Rightarrow} A$ does not exist.

c) *non-left-recursive* (NLR), if for any $A \in N$ and $\alpha \in V^*$ a derivation $A \stackrel{+}{\Rightarrow} A\alpha$ does not exist.

d) in *Greibach normal form* (GNF), if $P \subseteq N \times \Sigma N^*$.

We will also use the obvious notion of *non-right-recursiveness* (NRR) and we use the notation $\overline{\text{GNF}}$ if $P \subseteq N \times N^* \Sigma$.

Throughout this paper we assume that the (context-free) grammars in question are cycle-free and that the alphabets of the grammars do not contain *useless* symbols (cf. Aho and Ullman [1]).

Definition 1.2. a) Let V_1 and V_2 be alphabets. A *homomorphism* is a mapping $\psi: V_1 \to V_2^*$. The domain of the homomorphism ψ is extended to V_1^* by letting $\psi(\varepsilon) = \varepsilon$ and $\psi(\alpha a) = \psi(\alpha)\psi(a)$ for all $\alpha \in V_1^*$ and $a \in V_1$. We say that ψ is *fine* if $\psi: V_1 \to V_2 \cup \{\varepsilon\}$ and *very fine* if $\psi: V_1 \to V_2$.

b) Let $G = (N, \Sigma, P, S)$ be a CFG. We define $\tau_l(G) = \{(w, \pi) \mid S \underset{L}{\stackrel{\pi}{\Rightarrow}} w\}$ and $\tau_{\bar{r}}(G) = \{(w, \pi^R) \mid S \underset{R}{\stackrel{\pi}{\Rightarrow}} w\}$.

c) Assume that $x, y \in \{l, \bar{r}\}$. A CFG $G = (N, \Sigma, P, S)$ is said to *x-to-y cover* a CFG $G' = (N', \Sigma', P', S')$ if there exists a homomorphism $\psi: \Delta_{G'} \to \Delta_G^*$ such that

(i) if $(w, \pi') \in \tau_x(G')$ then $(w, \psi(\pi')) \in \tau_y(G)$, and

(ii) if $(w, \pi) \in \tau_y(G)$ then there exists π' such that $(w, \pi') \in \tau_x(G')$ and $\psi(\pi') = \pi$.

Clearly, if G' x-to-y covers G then $L(G) = L(G')$ and $\langle w, G' \rangle \geqq \langle w, G \rangle$. To denote that a production $A \to \alpha$ (or $i \cdot A \to \alpha$) is mapped on a string π of productions by a (cover) homomorphism we will sometimes write $A \to \alpha \langle \pi \rangle$ (or $i \cdot A \to \alpha \langle \pi \rangle$).

For the original cover definition the reader is referred to Gray and Harrison [6, 7][1] (cf. also Aho and Ullman [1]). A more general treatment of covers can be found in Nijholt [24]. The following notation will be useful.

Notation. a) $G'[l/l]\,G$, if G' left-to-left covers G (left cover).

b) $G'[l/\bar{r}]\,G$, if G' left-to-right covers G.

c) $G'[\bar{r}/l]\,G$, if G' right-to-left covers G.

d) $G'[\bar{r}/\bar{r}]\,G$, if G' right-to-right covers G (right cover).

In one of the main transformations of this paper we will use *chains* and *left production chains*.

Definition 1.3. Let $G = (N, \Sigma, P, S)$ be a CFG.

a) Define a relation $CH \subseteq V \times N^* \Sigma$ as follows. If $X_0 \in N$ then $CH(X_0)$, the set of *chains* of X_0 is defined by

$$CH(X_0) = \{X_0 X_1 \ldots X_n \mid X_0 \underset{L}{\Rightarrow} X_1 \psi_1 \underset{L}{\Rightarrow} \ldots \underset{L}{\Rightarrow} X_n \psi_n,\ \psi_i \in V^*,\ 1 \leqq i \leqq n\},$$

and for $c \in \Sigma$,

$$CH(c) = \{c\}.$$

---

[1] It should be observed that our cover definition is slightly different from theirs. Gray and Harrison's definition of complete cover may be compared with our definition of cover if we demand that the homomorphism is fine

b) Define a relation $LP \subseteq N^* \Sigma \times \Delta_G^*$ as follows. Let $\pi = X_0 X_1 \dots X_n \in N^+ \Sigma$, then $LP(\pi)$, the set of *left production chains* of π, is defined by

$$LP(\pi) = \{i_0 i_1 \dots i_{n-1} \mid X_0 \overset{i_0}{\underset{L}{\Rightarrow}} X_1 \psi_1 \overset{i_1}{\underset{L}{\Rightarrow}} \dots \overset{i_{n-1}}{\underset{L}{\Longrightarrow}} X_n \psi_n,\ \psi_i \in V^*,\ 1 \leqq i \leqq n\}.$$

If $\pi \in \Sigma$ then $LP(\pi) = \{\varepsilon\}$.

1.2. Covers and Parsing

Let $G = (N, \Sigma, P, S)$ be a CFG. A parser for G determines whether a string w of symbols is in $L(G)$ and if so it produces a parse tree for w with respect to G. Either left parses or right parses will be used to represent a parse tree. Once the parse tree is known, code generation can take place. Various persing methods have been introduced for the class of context-free grammars and its subclasses. For each parsing method there is a class of grammars which are suitable for this method. One can try to transform a grammar to make it suitable for a chosen parsing method or to improve its parsing properties. If this transformation can be done in such a way that the new grammar G' covers the original grammar G, then we can parse with respect to G' and, by applying the cover homomorphism, obtain the parse with respect to G. This is illustrated in Fig. 1.

It is usual to distinguish between top-down parsing and bottom-up parsing. In top-down parsing the goal is to find a left parse while in bottom-up parsing the goal is a right parse. Both for top-down parsing as for bottom-up parsing there exist conditions which, when satisfied by the grammar, can improve the parsing. A well-known condition for (deterministic) top-down parsing is that the grammar should be non-left-recursive. Grammars in GNF are non-left-recursive.

It has been observed in Griffiths and Petrick [8] that the original GNF transformation distorts the structure of the grammar in such a way that .... "To date, no efficient procedure for relating the structural descriptions of Greibach normal form grammars to the context-free grammars from which they were constructed has been found". Further investigations on this problem can be found in Kuno [17], Kurki-Suoni [18], Foster [4, 5] and Stearns [30]. The latter three authors do in fact use, in an informal way, the notion of a right cover. Gray and Harrison [6, 7] gave a formal definition of right covers. Their definition, which slightly differs from ours, was inspired by cover definitions

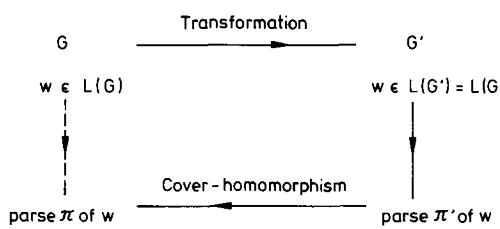

Fig. 1

which appear in unpublished work of J.C. Reynolds and R. Haskell. Soisalon-Soininen [29] has translated the results of Kurki-Suonio in the cover-formalism.

As mentioned in Nijholt [23] there has been some confusion on the possibility to cover grammars with grammars in GNF. In this paper we will give a transformation from arbitrary grammars to grammars in GNF such that a right cover can be defined.

2. Theorems and Transformations

This section contains a rather long list of theorems and transformations which are necessary to construct the cover-table which is presented in Sect. 4. For some algorithms and proofs the reader is referred to other papers. None of the algorithms not given here does have a complicated proof of correctness.

2.1. General Results

Our first results deal with some general observations on covers for context-free grammars. Firstly we will slightly generalize the cover definition in order to be able to present the following lemma. In the remainder of this paper we will not refer to this lemma if it is used. We will admit covers which are defined with the help of reversed left and right parses. We use $\bar{l}$ and r to denote them. Moreover, for any $x \in \{\bar{l}, \bar{r}\}$ we will write $\bar{\bar{x}} = x$.

Lemma 2.1. *If $G'[x/y]G$ then $G'[\bar{x}/\bar{y}]G$.*

Proof. Suppose that $G'[x/y]G$ under a cover-homomorphism ψ. Define $\psi'(i) = (\psi(i))^R$ for any $i \in \Delta_{G'}$. Homomorphism ψ' is the cover-homomorphism under which $G'[\bar{x}/\bar{y}]G$. □

Theorem 2.1. a) *For any CFG G there exists a CFG G' such that $G'[l/\bar{r}]G$.*
b) *For any CFG G there exists a CFG G' such that $G'[\bar{r}/l]G$.*

Proof. (a) Grammar G' is constructed from CFG G by defining

$$P' = \{A \to \alpha H_i \langle \varepsilon \rangle \mid i \cdot A \to \alpha \text{ is in } P\} \cup \{H_i \to \varepsilon \langle i \rangle \mid 1 \leqq i \leqq |P|\}.$$

The symbols H_i, $1 \leqq i \leqq |P|$ are newly introduced nonterminal symbols which are added to N to obtain N'.

(b) Grammar G' is constructed from CFG G by defining

$$P' = \{A \to H_i \alpha \langle \varepsilon \rangle \mid i \cdot A \to \alpha \text{ is in } P\} \cup \{H_i \to \varepsilon \langle i \rangle \mid 1 \leqq i \leqq |P|\}.$$

The symbols H_i, $1 \leqq i \leqq |P|$ are newly introduced nonterminal symbols which are added to N to obtain N'. □

The following observation on 'symmetry' will be very useful if we construct the cover-table in Sect. 4.

Observation 2.1. Let $G = (N, \Sigma, P, S)$ be a CFG. Define $G^R = (N, \Sigma, P^R, S)$ by letting $P^R = \{A \to \alpha^R \mid A \to \alpha \text{ is in } P\}$. Notice that a leftmost derivation of a sentence $w \in L(G)$ coincides with a rightmost derivation of $w^R \in L(G^R)$. In what follows we will frequently make use of this 'symmetry'. For example, if a grammar G can not be left covered by an ε-free grammar then it follows (cf. also Lemma 2.1) that G^R can not be right covered by an ε-free grammar. Another example is the situation in which a grammar G does not have a left-to-right covering grammar in GNF. Then G^R does not have a right-to-left covering grammar in $\overline{\text{GNF}}$. □

2.2 *Non-Left-Recursive Grammars*

Next we turn our attention to results which show the possibility of finding non-left-recursive grammars for 'arbitrary' context-free grammars.

Observation 2.2. If CFG G in Theorem 2.1 is non-left-recursive then (both in a) and b)) G' is non-left-recursive. □

Any ε-free CFG G (cycle-free, no useless symbols) can be transformed to a NLR grammar G' such that $G'[\bar{r}/\bar{r}]G$ and $G'[l/\bar{r}]G$. This result first appeared in Nijholt [22]. Soisalon-Soininen [29] gave a more simple proof of this result. One of the transformations which is used in the latter paper is based on an idea of Kurki-Suonio [18]. This trick can also be used for a transformation presented in Wood [34] and which is due to J.M. Foster.

Corollary 2.1. *Any ε-free CFG G can be transformed to a CFG G' such that G' is NLR and such that $G'[l/\bar{r}]G$ and $G'[\bar{r}/\bar{r}]G$.*

Each of the above mentioned methods to obtain the NLR grammar G' can be adapted in a very simple way in order to obtain an ε-free NLR grammar G'' such that $G''[\bar{r}/\bar{r}]G$. This result can also be obtained from a more general observation of Ukkonen [32, and unpublished] which we give, slightly adapted, below.

Corollary 2.2. *Any NLR grammar G can be transformed to an ε-free NLR grammar G' such that $G'[\bar{r}/\bar{r}]G$.*[2]

In Ukkonen's algorithm for eliminating ε-productions from a grammar $G = (N, \Sigma, P, S)$ it is assumed that if $\varepsilon \in L(G)$ then there do not exist two different rightmost derivations to ε. Since in our definition of ε-free grammar we have $P \subseteq N \times V^+$ we do not bother about introducing a special production $S' \to \varepsilon$ for grammar G'. Hence, in Corollary 2.2 we have $L(G') = L(G)/\{\varepsilon\}$.

The following corollary follows from the transitivity of the cover relation.

Corollary 2.3. *Any ε-free CFG G can be transformed to an ε-free NLR grammar G' such that $G'[\bar{r}/\bar{r}]G$.*

With this corollary we conclude our observations on finding non-left-recursive grammars.

---

[2] It is assumed that if $\varepsilon \in L(G)$ then there do not exist two different rightmost derivations to ε

2.3. Elimination of Single Productions

Before we turn our attention to the problem of finding grammars in GNF we have a few remarks on some special conditions. Consider a CFG G with productions $S \to A$, $S \to B$, $A \to a$ and $B \to a$. Suppose we want to find an equivalent ε-free grammar without *single productions* (i.e. productions of the form $X \to Y$ with both X and Y in N). There is only one grammar which has this property, grammar G' with the one production $S' \to a$. It follows that in genral elimination of single productions can not be done in such a way that a left or right cover can be defined since condition (ii) of the cover definition can not always be satisfied.

In some cases we find it convenient to talk about grammars without single productions. Although it is not always necessary (in some cases we could use more refined conditions) we assume for a few algorithms in the remainder of this paper that they have an input grammar without single productions. We use a rather rude approach to solve the problem of eliminating single productions. The method which is in the proof of the following theorem was first shown in [21] and we include it here. It should be observed that a more simple method can be used if we allow, as is possible in the grammar functor approach, that one production can have different labels. However, from the point of view of parsing we recognize productions rather than labels. Therefore we use the following method.

Theorem 2.2. *Let $G=(N, \Sigma, P, S)$ be an ε-free CFG. Grammar $G_0=(N \cup \{S_0\}, \Sigma \cup \{\perp\}, P \cup \{S_0 \to S\perp\}, S_0)$ can be transformed to a CFG G' without single productions in such a way that $G'[\bar{r}/\bar{r}]G_0$ and $G'[l/l]G_0$.*

Proof. We show how the elimination can be done. We use auxiliary sets P_0 and P_1. The set P_0 is the set of all the single productions in P. Initially $P_1 = \{A \to \alpha\langle i\rangle \mid i \cdot A \to \alpha$ is in $P - P_0\}$, $N' = N$ and $P' = \emptyset$.

(i) Let $A \in N$. If $A \overset{\delta}{\Rightarrow} \beta \overset{i}{\Rightarrow} \gamma$ is a derivation in G such that $\delta \neq \varepsilon$ and either $|\gamma| \geqq 2$ or $\gamma \in \Sigma$ then add $[A\delta i] \to \gamma\langle\pi\rangle$ to P_1 and $[A\delta i]$ to N'. To obtain a left cover define $\pi = \delta i$. To obtain a right cover define $\pi = i\delta^R$. Notice that since G is cycle-free there are finitely many derivations to consider.

(ii) Define a homomorphism $h: N' \cup \Sigma \to N \cup \Sigma$ by defining $h(X) = X$ for each $X \in N \cup \Sigma$ and $h([A\pi]) = A$ for each $[A\pi] \in N' - N$. For each production $A' \to \gamma\langle\pi\rangle$ in P_1 (hence, $A' \in N'$ and $\gamma \in VV^+$) add the productions in the set $\{A' \to \gamma'\langle\pi\rangle \mid A' \to \gamma\langle\pi\rangle$ in P_1 and $h(\gamma') = \gamma\}$ to P'.

(iii) Remove the useless symbols.

Clearly, grammar $G' = (N', \Sigma, P', S_0)$ which is obtained does not have single productions. Grammar G' left covers grammar G. This follows from the following observations. They can be formally proved by induction on the lengths of the derivations. Similar observations hold for the right cover.

a) If $A \underset{L}{\overset{\pi'}{\Rightarrow}} w$ in G' then $A \underset{L}{\overset{\pi}{\Rightarrow}} w$ in G, with $\psi(\pi') = \pi$.

b) If $[A\delta] \underset{L}{\overset{\pi'}{\Rightarrow}} w$ in G' then $A \underset{L}{\overset{\pi}{\Rightarrow}} w$ in G, with $\psi(\pi') = \pi$.

c) If $A \underset{L}{\overset{\pi}{\Rightarrow}} w$ in G then there exists π' such that either $A \underset{L}{\overset{\pi'}{\Rightarrow}} w$ in G' or $[A\delta] \underset{L}{\overset{\pi'}{\Rightarrow}} w$ in G', for some $\delta \in \Delta_G^*$, and with $\psi(\pi') = \pi$.

In observation c) we have for G' the grammar which is obtained from step (i) and (ii). The implicitly defined cover homomorphism is denoted by ψ. This concludes the proof of the theorem. □

We emphasize that it is not always necessary to introduce the special production $S_0 \to S\bot$. For example, if G is unambiguous. In this case the method mentioned in the proof can be simplified. In fact, only in the case that there exist, for some $a \in \Sigma$, different derivations from S to a it is necessary to introduce this production.

In what follows we do not bother about this special production. The result mentioned in the following observation is an immediate consequence of the method which is used in the proof of Theorem 2.2.

Observation 2.3. If CFG G in Theorem 2.2 is non-left-recursive then CFG G' without single productions is also non-left-recursive. □

2.4. Grammars in Greibach Normal Form

Now we are sufficiently prepared to consider grammars in GNF. This normal form can be obtained in such a way, from ε-free and non-left-recursive grammars, that a left cover can be defined. This was shown in Nijholt [21]. Moreover, this result is a special case of a more general theorem in Nijholt [24]. In the latter paper a transformation (the '*left part transformation*') is used which we will recall here. This transformation will be used later, in an adapted form, to obtain right cover results.

We use a special alphabet which is defined below.

Definition 2.1. Let $G=(N, \Sigma, P, S)$ be a CFG. Define the set

$$[N]=\{[Ai\alpha] \mid i \cdot A \to \alpha\beta \text{ is in } P, \beta \in V^*\}$$

and a homomorphism $\xi: [N] \to [N]$ by letting $\xi([Ai\alpha])$ is

(i) ε if $i \cdot A \to \alpha$ is in P.

(ii) $[Ai\alpha]$ if $i \cdot A \to \alpha\beta$ is in P, with $\beta \neq \varepsilon$.

We present the algorithm in such a way that each newly obtained production is followed by its image under a cover-homomorphism ψ for a left cover.

Algorithm 2.1. *Input. An ε-free NLR grammar $G=(N, \Sigma, P, S)$ without single productions. Qutput. A CFG $G'=(N', \Sigma, P', [S])$ in GNF such that $G'[l/l]G$. Method. The set P' consists of all the productions which are introduced below. Set N' will contain $[S]$ and all symbols of $[N]$ which appear in the productions. Initially set $P'=\emptyset$.*

(i) *For each pair (π, ρ), $\pi=SX_1X_2\ldots X_n \in CH(S)$ and $\rho=i_0i_1\ldots i_{n-1} \in LP(\pi)$, add $[S] \to X_n\xi([X_{n-1}i_{n-1}X_n]\ldots[Si_0X_1])\langle\rho\rangle$ to P'.*

(ii) *Let $i \cdot A \to \alpha X_0 \varphi$ be in P, $\alpha \neq \varepsilon$. For each pair (π, ρ), $\pi=X_0X_1\ldots X_n \in CH(X_0)$ and $\rho=i_0i_1\ldots i_{n-1} \in LP(\pi)$, add $[Ai\alpha] \to X_n\xi([X_{n-1}i_{n-1}X_n]\ldots[X_0i_0X_1][Ai\alpha X_0])\langle\rho\rangle$ to P'.* □

Notice that for this algorithm the condition that the input grammar G does not have single productions is not a necessary condition. It would be sufficient to demand that, for any $A \in N$ and $X \in V$, if $A \overset{\pi}{\Rightarrow} X$ and $A \overset{\pi'}{\Rightarrow} X$ then $\pi = \pi'$. However, as we have shown the single productions can be eliminated in a simple way and we can avoid the introduction of new conditions.

Theorem 2.3. *Each ε-free NLR grammar G can be transformed to a CFG G' in GNF such that $G'[l/l]\,G$.*

Proof. We assume that the single productions have been eliminated. We use Algorithm 2.1 to transform G to a grammar G'. Clearly, G' is in GNF. The cover homomorphism which is implicitly defined in the algorithm is denoted by ψ. We use two claims.

Claim 1. If $[Ai\alpha] \underset{L}{\overset{\pi'}{\Rightarrow}} w$ in G' then there exists $i \cdot A \to \alpha\varphi$ in P, $\varphi \neq \varepsilon$, such that $\varphi \underset{L}{\overset{\pi}{\Rightarrow}} w$ in G, with $\pi = \psi(\pi')$.

Proof of Claim 1. Induction on $|\pi'|$.

Basis. Assume $|\pi'| = 1$. In this case we have a production $\pi' \cdot [Ai\alpha] \to w$ in P' with $w \in \Sigma$. This production is obtained from either a production $i \cdot A \to \alpha w$ in P such that $\pi = \varepsilon$ and $\psi(\pi') = \varepsilon$ or from productions $i \cdot A \to \alpha X_0$ and $j \cdot X_0 \to w$, with $\psi(\pi') = j$.

Induction. Assume $|\pi'| > 1$. We can write

$$[Ai\alpha] \underset{L}{\overset{i'}{\Longrightarrow}} a\xi([X_{n-1}i_{n-1}X_n] \dots [X_0 i_0 X_1][Ai\alpha X_0]) \underset{L}{\overset{\pi''}{\Longrightarrow}} av,$$

where $X_n = a$, $i'\pi'' = \pi'$ and $av = w$.

We can factorize π'' such that

(a) $\pi'' = \pi'_n \dots \pi'_0$, $v = w_n w_{n-1} \dots w_0$.

(b) If $\xi([Ai\alpha X_0]) \neq \varepsilon$ then

$$[Ai\alpha X_0] \underset{L}{\overset{\pi'_0}{\Longrightarrow}} w_0$$

and it follows from the induction hypothesis that there exists $i \cdot A \to \alpha X_0 \varphi_0$ such that $\varphi_0 \underset{L}{\overset{\pi_0}{\Rightarrow}} w_0$, with $\psi(\pi'_0) = \pi_0$. If $\xi([Ai\alpha X_0]) = \varepsilon$ then $\varphi_0 = \pi_0 = w_0 = \varepsilon$.

(c) If $\xi([X_{k-1}i_{k-1}X_k]) \neq \varepsilon$, $0 < k \leqq n$, then

$$[X_{k-1}i_{k-1}X_k] \underset{L}{\overset{\pi'_k}{\Longrightarrow}} w_k$$

and it follows that there exists $i_{k-1} \cdot X_{k-1} \to X_k \varphi_k$ such that $\varphi_k \underset{L}{\overset{\pi_k}{\Longrightarrow}} w_k$, with $\psi(\pi'_k) = \pi_k$. If $\xi([X_{k-1}i_{k-1}X_k]) = \varepsilon$ then $\varphi_k = \pi_k = w_k = \varepsilon$. Since G has no single productions, this case can only occur if $k = n$.

Hence, since $\psi(i') = i_0 i_1 \dots i_{n-1} = \rho$ we obtain a derivation

$$\varphi = X_0\varphi_0 \underset{L}{\overset{\rho}{\Longrightarrow}} X_n \varphi_n \varphi_{n-1} \dots \varphi_1 \varphi_0 \underset{L}{\overset{\delta}{\Longrightarrow}} X_n w_n w_{n-1} \dots w_1 w_0 = av,$$

with $\delta = \pi_n \dots \pi_1 \pi_0$ and $\psi(\pi') = \rho\delta = \pi$. This concludes the proof of Claim 1. □

If $[S] \overset{\pi'}{\underset{L}{\Rightarrow}} w$ in G' then we can, as we did in the induction part of the proof of Claim 1, distinguish the first production which is used and then factorize π' in a similar way to obtain the conclusion that $S \overset{\pi}{\underset{L}{\Rightarrow}} w$ in G, with $\psi(\pi')=\pi$.

Claim 2. Assume $i \cdot A \to \alpha X_0 \varphi_0 \in P$, $\alpha \neq \varepsilon$. If $X_0 \varphi_0 \overset{\pi}{\underset{L}{\Rightarrow}} w$ in G then there exists $\pi' \in \Delta_G^*$, such that $[A i \alpha] \overset{\pi'}{\underset{L}{\Rightarrow}} w$ in G' and $\psi(\pi')=\pi$.

Proof of Claim 2. The proof is by induction on $|\pi|$.

Basis. If $|\pi|=0$ then $X_0 \varphi_0 \in \Sigma^+$ and from the construction it follows that there exists a derivation $[A i \alpha] \overset{\pi'}{\underset{L}{\Rightarrow}} w$ in G' such that $\psi(\pi')=\pi=\varepsilon$. If $|\pi|=1$, then $X_0 \varphi_0$ can be written as $v_1 B v_3$ with $B \in N$, $v_1 v_3 \in \Sigma^*$ and $\pi \cdot B \to v_2$ in P with $v_2 \in \Sigma^+$. In this case there exists a derivation

$$[A i \alpha] \overset{\pi_1}{\underset{L}{\Longrightarrow}} v_1 [A i \alpha v_1] \overset{\pi_2}{\underset{L}{\Longrightarrow}} v_1 v_2 \xi([A i \alpha v_1 B]) \overset{\pi_3}{\underset{L}{\Longrightarrow}} v_1 v_2 v_3 = w$$

in G' such that $\psi(\pi_1)=\psi(\pi_3)=\varepsilon$ and $\psi(\pi_2)=\pi$.

Induction. Assume $|\pi|>1$. Then there exists $\rho=i_0 i_1 \dots i_{n-1} \in \Delta_G^*$ such that $i_k \cdot X_k \to X_{k+1} \varphi_{k+1}$, $0 \leqq k \leqq n-1$, with $X_n \in \Sigma$ and such that $\varphi_k \overset{\pi_k}{\underset{L}{\Longrightarrow}} w_k$. Hence, we can write

$$X_0 \varphi_0 \overset{\rho}{\underset{L}{\Rightarrow}} X_n \varphi_n \dots \varphi_1 \varphi_0 \overset{\delta}{\underset{L}{\Rightarrow}} w,$$

with $\delta=\pi_n \dots \pi_1 \pi_0$ and $w=X_n w_n \dots w_1 w_0$.

From the construction and the induction hypothesis it follows that

(a) $i' \cdot [A i \alpha] \to X_n \xi([X_{n-1} i_{n-1} X_n] \dots [A i \alpha X_0])\langle\rho\rangle$ is in P'.

(b) If $\varphi_k \neq \varepsilon$, $0<k\leqq n$, then $[X_{k-1} i_{k-1} X_k] \overset{\pi'_k}{\underset{L}{\Longrightarrow}} w_k$, with $\psi(\pi'_k)=\pi_k$. If $\varphi_k=\varepsilon$ then $\pi_k=\pi'_k=w_k=\varepsilon$. Since G has no single productions this latter case can only occur if $k=n$.

(c) If $\varphi_0 \neq \varepsilon$ then $[A i \alpha X_0] \overset{\pi'_0}{\underset{L}{\Longrightarrow}} w_0$, with $\psi(\pi'_0)=\pi_0$. If $\varphi_0=\varepsilon$ then $\pi_0=\pi'_0=w_0 =\varepsilon$.

Hence, there exists a derivation

$$[A i \alpha] \overset{\pi'}{\underset{L}{\Longrightarrow}} w, \quad \text{with } \pi'=i' \pi'_n \dots \pi'_0 \quad \text{and} \quad \psi(\pi')=\rho\delta=\pi.$$

This concludes the proof of Claim 2. □

It remains to show that if $S \overset{\pi}{\underset{L}{\Rightarrow}} w$ in G, then $[S] \overset{\pi'}{\underset{L}{\Rightarrow}} w$ in G', with $\psi(\pi')=\pi$. However, also in this case this can be shown by proceeding in a way similar to the induction part of the proof of Claim 2. It follows that $G'[l/l]G$. □

Next we consider the possibility to obtain a CFG in GNF which right covers the ε-free NLR grammar. We use two transformations. Firstly, we transform ε-free NLR grammars to grammars which are *almost* GNF. For convenience of description we assume that the input grammar is such that terminal symbols in the righthand sides of the productions can only appear at the leftmost positions of the righthand sides. This can be done without loss of generality. For example, if a grammar has a production $i \cdot A \to \alpha a \beta$, with $\alpha \neq \varepsilon$, then we can replace this production by $A \to \alpha H_a \beta \langle i \rangle$ and $H_a \to a \langle \varepsilon \rangle$ and the new grammar right covers the original grammar. The second transformation will produce GNF grammars from almost GNF grammars.

Definition 2.2. A CFG $G=(N, \Sigma, P, S)$ is said to be an *almost* GNF grammar if for any production $A \to \alpha$ in P either

(i) $\alpha \in \Sigma$, or

(ii) $\alpha \in NN^+$ and $rhs(1:\alpha) \subseteq \Sigma$.

Algorithm 2.2. *Input. A NLR grammar $G=(N, \Sigma, P, S)$ such that $P \subseteq N \times (\Sigma N^* \cup NN^+)$. Output. An almost GNF grammar $G'=(N', \Sigma, P', [S])$, $G'[\bar{r}/\bar{r}]G$. Method. The set P' will contain all productions introduced below. The set N' will contain $[S]$, all symbols of $[N]$ which appear in the productions and some special indexed symbols H. Initially set $P'=\emptyset$.*

(i) For each production of the form $i \cdot S \to a$ in P with $a \in \Sigma$, add $[S] \to a\langle i \rangle$ to P'.

(ii) For each pair (π, ρ), $\pi = SX_1 \dots X_n \in CH(S)$ and $\rho = i_0 i_1 \dots i_{n-1} \in LP(\pi)$, $n > 1$, add

$$[S] \to H_{i_{n-1}} \xi([X_{n-1} i_{n-1} X_n] \dots [S i_0 X_1])\langle \varepsilon \rangle$$

and

$$H_{i_{n-1}} \to X_n \langle p \rangle$$

to P'. Here, $p = i_{n-1}$ if $i_{n-1} \cdot X_{n-1} \to X_n \in P$ and $p = \varepsilon$ otherwise.

(iii) Let $i \cdot A \to \alpha X_0 \varphi$ be in P, $\alpha \neq \varepsilon$. For each pair (π, ρ), $\pi = X_0 X_1 \dots X_n \in CH(X_0)$ and $\rho = i_0 i_1 \dots i_{n-1} \in LP(\pi)$, the following two cases are distinguished:

(1) $n=1$, $\varphi=\varepsilon$, and $\xi([X_0 i_0 X_1])=\varepsilon$; add $[A i \alpha] \to X_n \langle i_0 i \rangle$ to P'.

(2) otherwise, add

$$[A i \alpha] \to H_{i_{n-1}} \xi([X_{n-1} i_{n-1} X_n] \dots [X_0 i_0 X_1][A i \alpha X_0])\langle p \rangle$$

and

$$H_{i_{n-1}} \to X_n \langle q \rangle$$

to P', where $p = i$ if $i \cdot A \to \alpha X_0$ is in P and $p = \varepsilon$ otherwise, and $q = i_{n-1}$ if $i_{n-1} \cdot X_{n-1} \to X_n \in P$ and $q = \varepsilon$ otherwise. □

Lemma 2.2. *Any ε-free NLR grammar G can be transformed to an almost GNF grammar G' such that $G'[\bar{r}/\bar{r}]G$.*

Proof. Without loss of generality we may assume that G does not have single productions. We use Algorithm 2.2 to transform G to a grammar G'. By construction G' is almost GNF.

Claim 1. Cover homomorphism ψ, implicitly defined in the algorithm, is well-defined.

Proof of Claim 1. To verify that for any pair p and p' of productions in P' it follows that if $\psi(p)=\pi$ and $\psi(p')=\pi'$, with $\pi \neq \pi'$ then $p \neq p'$. This is straightforward to verify and therefore it is omitted. □

In the following claims $\varphi: \Delta_{G'} \to \Delta_G^*$ is defined by letting, for any $p \in \Delta_{G'}$, $\varphi(p) = \pi^R$ iff $\psi(p)=\pi$.

Claim 2. If $[Ai\alpha] \xRightarrow[R]{\pi'} w$ then $A \xRightarrow[R]{\varphi(\pi')} \alpha w$.

Proof of Claim 2. The proof is by induction on $|\pi'|$.

Basis. If $|\pi'|=1$ then $\pi'=[Ai\alpha] \to a\langle ji \rangle$. In this case there is a derivation

$$A \xRightarrow[R]{i} \alpha X_0 \xRightarrow[R]{j} \alpha a, \text{ for } X_0 \in N.$$

Induction. Assume $|\pi'|=m$, $m>1$ and assume the property holds for all rightmost derivations with lengths less than m. Let

$$p' \cdot [Ai\alpha] \to H_{i_{n-1}} \xi([X_{n-1} i_{n-1} X_n] \dots [X_0 i_0 X_1][Ai\alpha X_0])$$

be the first production which is used in the derivation $[Ai\alpha] \xRightarrow[R]{\pi'} w$. Hence, we may write $w = X_n x$ and $\pi' = p'\gamma q'$, where $q' = H_{i_{n-1}} \to X_n$. Then we have

$$[Ai\alpha] \xRightarrow[R]{p'} H_{i_{n-1}} \xi([X_{n-1} i_{n-1} X_n] \dots [X_0 i_0 X_1][Ai\alpha X_0]) \xRightarrow[R]{\gamma} \dots$$
$$\dots \xRightarrow[R]{\gamma} H_{i_{n-1}} x_n x_{n-1} \dots x_1 x_0 \xRightarrow[R]{q'} X_n x_n x_{n-1} \dots x_1 x_0 = w,$$

such that

(a) if $\xi([Ai\alpha X_0]) \neq \varepsilon$ then $[Ai\alpha X_0] \xRightarrow[R]{\pi_0} x_0$, otherwise $x_0 = \pi_0 = \varepsilon$,

(b) $[X_{k-1} i_{k-1} X_k] \xRightarrow[R]{\pi_k} x_k$, $1 \leqq k \leqq n-1$,

(c) if $\xi([X_{n-1} i_{n-1} X_n]) \neq \varepsilon$ then $[X_{n-1} i_{n-1} X_n] \xRightarrow[R]{\pi_n} x_n$, otherwise $\pi_n = x_n = \varepsilon$,

and

(d) $q' \cdot H_{i_{n-1}} \to X_n$

with $p' \pi_0 \pi_1 \dots \pi_n q' = p' \gamma q' = \pi'$.

It follows from the induction hypothesis that

(a)′ $A \xRightarrow[R]{\varphi(p'\pi_0)} \alpha X_0 x_0$, with either $\varphi(p')=\varepsilon$ or $\pi_0 = x_0 = \varepsilon$,

(b)′ $X_{k-1} \xRightarrow[R]{\varphi(\pi_k)} X_k x_k$, $1 \leqq k \leqq n-1$, and

(c)′ $X_{n-1} \xRightarrow[R]{\varphi(\pi_n q')} X_n x_n$, with either $\varphi(q')=\varepsilon$ or $\pi_n = x_n = \varepsilon$. Thus, $A \xRightarrow[R]{\varphi(\pi')} \alpha w$. □

Claim 3. Assume that $i \cdot A \to \alpha X_0 \varphi$ is in P and $A \xRightarrow[R]{i\pi} \alpha w$. Then there exists $\pi' \in \Delta_{G'}^*$ such that $[Ai\alpha] \xRightarrow[R]{\pi'} w$ and $\varphi(\pi') = i\pi$.

Proof of Claim 3. The proof is by induction on $|\pi|$.

Basis. If $|\pi|=1$ then, with $\pi \cdot X_0 \to w$ in P, $w \in \Sigma$, we have

$$A \underset{R}{\overset{i}{\Rightarrow}} \alpha X_0 \underset{R}{\overset{\pi}{\Rightarrow}} \alpha w$$

in G, and by construction of G'

$$[A i \alpha] \underset{R}{\overset{i'}{\Rightarrow}} w,$$

with $\varphi(i')=i\pi$.

Induction. Assume $|\pi|>1$. We factorize

$$A \underset{R}{\overset{i}{\Rightarrow}} \alpha X_0 \varphi \underset{R}{\overset{\pi}{\Rightarrow}} \alpha w$$

into

$$A \underset{R}{\overset{i}{\Rightarrow}} \alpha X_0 \varphi,$$

$$\varphi \underset{R}{\overset{\rho_1}{\Longrightarrow}} v_1, \quad \text{and} \quad X_0 \underset{R}{\overset{\rho_0}{\Longrightarrow}} a v_0,$$

where $a v_0 v_1 = w$. Since $X_0 \in N$ we have $|\rho_1|<|\pi|$ and from the induction hypothesis we obtain, if $\varphi \neq \varepsilon$,

$$[A i \alpha X_0] \underset{R}{\overset{\rho'_1}{\Longrightarrow}} v_1, \quad \text{with} \quad \varphi(\rho'_1)=i\rho_1.$$

Moreover, there exist productions $i_k \cdot X_k \to X_{k+1}\varphi_k$, $0 \leqq k \leqq n-1$ and $X_n=a$, such that

(i) $X_k \underset{R}{\overset{i_k}{\Longrightarrow}} X_{k+1}\varphi_k \underset{R}{\overset{\pi_k}{\Longrightarrow}} X_{k+1} w_k$, with $0 \leqq k \leqq n-1$ and such that $|\pi_k|<|\pi|$,

hence

$$[X_k i_k X_{k+1}] \underset{R}{\overset{\pi'_k}{\Longrightarrow}} w_k \quad \text{and} \quad \varphi(\pi'_k)=i_k\pi_k$$

(ii) $X_{n-1} \underset{R}{\overset{i_{n-1}}{\Longrightarrow}} a\varphi_n \underset{R}{\overset{\pi_{n-1}}{\Longrightarrow}} a w_{n-1}$, such that $|\pi_{n-1}|<|\pi|$, hence, if $\varphi_n \neq \varepsilon$,

$$[X_{n-1} i_{n-1} a] \underset{R}{\overset{\pi'_{n-1}}{\Longrightarrow}} w_{n-1}, \quad \text{and} \quad \varphi(\pi'_{n-1})=i_{n-1}\pi_{n-1}$$

(iii) $w_{n-1} \dots w_1 w_0 = v_0$ and $i_0 \pi_0 i_1 \pi_1 \dots i_{n-1} \pi_{n-1} = \rho_0$.

It follows that in P' there exists a production

$$p' \cdot [A i \alpha] \to H_{i_{n-1}} \xi([X_{n-1} i_{n-1} a] \dots [X_0 i_0 X_0][A i \alpha X_0])$$

and a derivation $[A i \alpha] \underset{R}{\overset{\pi'}{\Rightarrow}} w$, such that

(a) $w = a v_0 v_1$, $\pi' = p' \rho'_1 \pi'_0 \dots \pi'_{n-1} q'$, with q' is $H_{i_{n-1}} \to a$.

(b) $\varphi(p' \rho'_1) = i\rho_1$,

$$\varphi(\pi'_0 \dots \pi'_{n-1}) = i_0 \pi_0 i_1 \dots i_{n-2} \pi_{n-2},$$

$$\varphi(\pi'_{n-1} q') = i_{n-1} \pi_{n-1}, \quad \text{and} \quad i_0 \pi_0 i_1 \dots i_{n-2} \pi_{n-2} i_{n-1} \pi_{n-1} = \rho_0.$$

Hence, $\varphi(\pi') = i \rho_1 \rho_0 = i\pi$. □

Now it is not difficult to verify that $G'[\bar{r}/\bar{r}]G$. We leave the details to the reader and we only mention that if $[S] \overset{\pi'}{\underset{R}{\Rightarrow}} w$ then one should distinguish the first production from the remainder of the derivation. A similar argument can be used for the reverse direction (Condition (ii) of the cover definition). This concludes the proof of Lemma 2.2. □

Next we show that any almost GNF grammar can be transformed to a GNF grammar. This is done in the following algorithm. The newly obtained grammar will right cover the original grammar.

Algorithm 2.3. *Input. An almost GNF grammar $G=(N, \Sigma, P, S)$. Output. A GNF grammar $G'=(N', \Sigma, P', S)$ such that $G'[\bar{r}/\bar{r}]G$. Method. We use two auxiliary sets, N_0 and P_0. Initially set $N'=N$, $N_0=\emptyset$ and*

$$P_0=\{A\to\alpha\langle i\rangle \mid i\cdot A\to\alpha \text{ is in } P \text{ and } \alpha\in\Sigma\}.$$

Step 1: For each production $i\cdot A\to BC\alpha$ in P (with $B, C\in N$ and $\alpha\in N^*$) the following is done.

(i) If $j\cdot C\to D\beta E$ is in P (with $D, E\in N$ and $\beta\in N^*$) then, for any pair of productions $k\cdot B\to a$ and $l\cdot D\to b$ in P add

$$A\to aH_{kl}\beta[Ej]\alpha\langle i\rangle$$

and

$$H_{kl}\to b\langle kl\rangle$$

to P_0. Add $[Ej]$ to N_0 and $[Ej]$ and H_{kl} to N'.

(ii) If $j\cdot C\to b$ is in P, then, for any production $k\cdot E\to a$ add

$$A\to aH_{kj}\alpha\langle i\rangle$$

and

$$H_{kj}\to b\langle kj\rangle$$

to P_0. Add H_{kj} to N'.

Step 2: Set $P'=P_0$. For each $[Ej]$ in N_0 add $[Ej]\to\alpha\langle ij\rangle$ to P' for each production $E\to\alpha\langle i\rangle$ in P_0.

Step 3: Remove the useless symbols. □

The general idea of the transformation is displayed in Fig. 2.

Lemma 2.3. *Any almost GNF grammar G can be transformed to a GNF grammar G' such that $G'[\bar{r}/\bar{r}]G$.*

Proof. Let $\psi: \Delta_{G'}\to\Delta_G^*$ be the cover homomorphism which is defined in the algorithm. As we did in the proof of Lemma 2.2 we will use homomorphism φ instead of ψ. Two claims are used in the proof of Lemma 2.3. For any triple of strings α, β and γ with $\alpha=\beta\gamma$ we have that $\alpha/3$ denotes γ.

Claim 1. Assume $A\in N$.

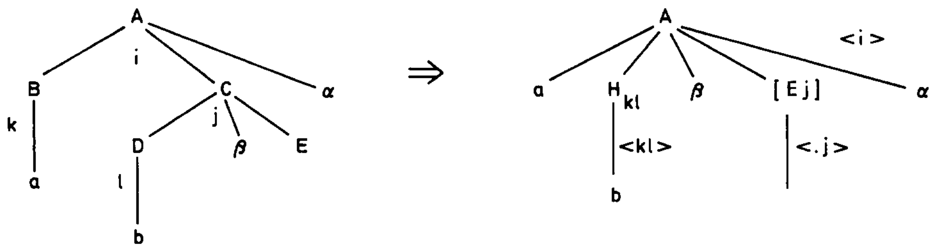

Fig. 2. Step 1 of Algorithm 2.3

(i) If $A \overset{\pi'}{\underset{R}{\Rightarrow}} w$ in G' then $A \overset{\varphi(\pi')}{\underset{R}{\Longrightarrow}} w$ in G.

(ii) If $[Ak] \overset{\pi'}{\underset{R}{\Rightarrow}} w$ in G' then $A \overset{\delta}{\underset{R}{\Rightarrow}} w$ in G, with $\delta = \varphi(\pi')/k$.

Proof of Claim 1. The proof is by induction on $|\pi'|$.

Basis. If $|\pi'| = 1$ then we have

(i) Production $A \to w$ is both in P and P', hence the claim is trivially satisfied.

(ii) Production $\pi' \cdot [Ak] \to w$ is in P'. From step 2 of the algorithm it follows that $\varphi(\pi') = ki$, where $i \cdot A \to w$ is in P. Therefore $A \overset{\delta}{\underset{R}{\Rightarrow}} w$ in G, with $\delta = \varphi(\pi')/k$.

Induction. Consider case (i). Assume $A \overset{\pi'}{\underset{R}{\Rightarrow}} w$ in G', with $|\pi'| > 1$. The first production which is used in this derivation is either of the form $i' \cdot A \to aH_{kl}\beta[Ej]\alpha\langle i\rangle$ or $i' \cdot A \to aH_{kj}\alpha\langle i\rangle$. Notice that in both cases we can completely determine from which two productions of P such a production has been constructed. We continue with the former case. The case in which $A \to aH_{kj}\alpha$ is the first production can be treated similarly and is therefore omitted. Now we can factorize the derivation in the following way:

(a) $i' \cdot A \to aH_{kl}\beta[Ej]\alpha$, with $\varphi(i') = i \cdot A \to BC\alpha$, where B is the lefthand side of production k in P and C is the lefthand side of production j in P.

(b) $\alpha \overset{\pi_0'}{\underset{R}{\Longrightarrow}} w_0$, and from the induction hypothesis it follows that $\alpha \overset{\pi_0}{\underset{R}{\Longrightarrow}} w_0$ in G, where $\pi_0 = \varphi(\pi_0')$.

(c) $[Ej] \overset{\pi_1'}{\underset{R}{\Longrightarrow}} w_1$, and from the induction hypothesis it follows that $E \overset{\pi_1}{\underset{R}{\Longrightarrow}} w_1$ in G, where $\pi_1 = \varphi(\pi_1')/j$.

(d) $\beta \overset{\pi_2'}{\underset{R}{\Longrightarrow}} w_2$, and from the induction hypothesis it follows that $\beta \overset{\pi_2'}{\underset{R}{\Longrightarrow}} w_2$ in G, where $\pi_2 = \varphi(\pi_2')$.

(e) $q' \cdot H_{kl} \to b$, where we assume that $b \in \Sigma$ is the righthand side of production l in P. Moreover, $\varphi(q') = lk$.

It follows that $i'\pi_0'\pi_1'\pi_2'q' = \pi'$, $abw_2w_1w_0 = w$ and $\varphi(\pi') = i\pi_0 j\pi_1\pi_2 lk$, such that (if we assume that D is the lefthand side of production l)

$$A \overset{i}{\underset{R}{\Longrightarrow}} BC\alpha \overset{\pi_0}{\underset{R}{\Longrightarrow}} BCw_0 \overset{j}{\underset{R}{\Longrightarrow}} BD\beta Ew_0 \overset{\pi_1}{\underset{R}{\Longrightarrow}} BD\beta w_1 w_0 \cdots$$
$$\cdots \overset{\pi_2}{\underset{R}{\Longrightarrow}} BDw_2w_1w_0 \overset{l}{\underset{R}{\Longrightarrow}} Bbw_2w_1w_0 \overset{k}{\underset{R}{\Longrightarrow}} abw_2w_1w_0 = w.$$

This concludes the verification of case (i). Case (ii) can be verified along similar lines and therefore this case is omitted. This concludes the induction part of the proof and therefore the claim is proved. □

Claim 2. Consider CFG G' before step 3 of the algorithm is executed. If $A \overset{\pi}{\underset{R}{\Rightarrow}} w$ in G then there exists $\pi' \in \Delta_{G'}^*$ such that $A \overset{\pi'}{\underset{R}{\Rightarrow}} w$ in G' and $\varphi(\pi') = \pi$.

Proof of Claim 2. In the proof which may proceed by induction on $|\pi|$ one should distinguish that $A \overset{\pi}{\underset{R}{\Rightarrow}} w$ in G can also imply $[Ak] \overset{\pi'}{\underset{R}{\Rightarrow}} w$, for some $k \in \Delta_G$ and with $\varphi(\pi')/k = \pi$. We omit the proof since it proceeds along the same lines as the proof of Claim 1. □

From these two claims it is now clear that $G'[\bar{r}/\bar{r}]G$. □

The next theorem follows from the previous results.

Theorem 2.4. *Any ε-free CFG G can be transformed to a CFG G′ in GNF such that* $G'[\bar{r}/\bar{r}]G$.

Proof. For any ε-free CFG G we can find an ε-free NLR grammar G_0 (Corollary 2.3) such that $G_0[\bar{r}/\bar{r}]G$. The single productions of G_0 can be eliminated in such a way that the right cover is preserved (Theorem 2.2) and the new grammar, which is also non-left-recursive (Observation 2.3) can be transformed with Algorithm 2.2 followed by Algorithm 2.3 to a grammar G' which is in GNF and which has the property $G'[\bar{r}/\bar{r}]G$. □

Now that we have seen this positive cover result one can ask for analogous results for left covers and left-to-right covers. Unfortunately these results can not be given in such a general way as the right cover result. We will extensively return to this problem in the forthcoming sections. A few positive results on $l/\bar{r}$- and $\bar{r}/l$-covers will be presented here.

Lemma 2.4. *Any CFG G in* $\overline{\text{GNF}}$ *can be transformed to a CFG G′ in GNF such that* $G'[l/\bar{r}]G$.

Proof. Assume that $G = (N, \Sigma, P, S)$ is in $\overline{\text{GNF}}$. Define $G_R = (N, \Delta_G, P_R, S)$ with

$$P_R = \{A \to \alpha i \mid i \cdot A \to \alpha a \in P,\ a \in \Sigma\}.$$

Define a homomorphism $\varphi: \Delta_G \to \Sigma$ by letting $\varphi(i) = a$ if $i \cdot A \to \alpha a$ is in P. Notice that G_R is unambiguous. Find for G_R an equivalent grammar $G_L = (N', \Delta_G, P_L, S')$ in GNF (for example, apply to G_R a transformation to GNF). Grammar G' and the associated cover homomorphism ψ is obtained from G_L by defining

$$P' = \{i' \cdot A' \to a\alpha' \langle j \rangle \mid i' \cdot A' \to j\alpha' \in P_L \text{ and } \varphi(j) = a\}.$$

We may conclude that $G'[l/\bar{r}]G$ if we have verified that ψ is well-defined, that is, if $i' \cdot A' \to i\alpha'$ and $j' \cdot A' \to j\alpha'$ are in P_L, then $i \neq j$ (hence, $i' \neq j'$) implies $\varphi(i) \neq \varphi(j)$.

But this property is trivially satisfied since otherwise we are able to generate sentences of the form $\pi_1 i \pi_2$ and $\pi_1 j \pi_2$ in $L(G_L)$ and since $\varphi(\pi_1 i \pi_2) = \varphi(\pi_1 j \pi_2) = w \in L(G)$ we have two different right parses for the same sentence w. Since these two right parses are only different in one production this is impossible. □

The usefulness of this lemma will become clear with the following observation. We know (Theorem 2.3 and 'symmetry') that any ε-free NRR grammar G can be transformed to a CFG G_0 in $\overline{\text{GNF}}$ such that $G_0[\bar{r}/\bar{r}]\,G$. From Lemma 2.4 it follows that we can transform G_0 to a grammar G' in GNF such that $G'[l/\bar{r}]\,G_0$ and from transitivity we obtain $G'[l/\bar{r}]\,G$.

Corollary 2.4. *Any ε-free NRR grammar G can be transformed to a CFG G' in GNF such that $G'[l/\bar{r}]\,G$.*

A similar result has been obtained in Ukkonen [31 and unpublished].

Once we have a grammar in GNF there is still one more useful transformation which can be applied. The following algorithm is a slight generalization of a method which was first used in [23].

Algorithm 2.4. *Input. A CFG $G=(N,\Sigma,P,S)$ in GNF. Output. A CFG $G'=(N',\Sigma,P',S)$ in GNF such that $G'[\bar{r}/l]\,G$. Method. Initially set $P'=\{A\to a\langle i\rangle \mid i\cdot A\to a\in P,\ a\in\Sigma\}$ and $N'=N$. The indexed symbols H which are introduced below are added to N'.*

(i) For each production of the form $i\cdot A\to a\alpha$ in P, $\alpha\neq\varepsilon$, the following is done. Assume $\alpha=B\gamma$, $\gamma\in N^*$. For any $j_k\cdot B\to b_k\gamma_k$ in P, $1\leq k\leq |rhs(B)|$ add

$$A\to aH_{ij_k}\gamma_k\gamma\langle\varepsilon\rangle$$

and

$$H_{ij_k}\to b_k\langle ij_k\rangle$$

to P'.

(ii) Remove all useless symbols. □

Theorem 2.5. *Any CFG G in GNF can be transformed to a CFG G' in GNF such that $G'[\bar{r}/l]\,G$.*

Proof. We use two claims to prove the theorem. Homomorphism φ is defined as in the proof of Lemma 2.2.

Claim 1. If $A\overset{\pi'}{\underset{R}{\Longrightarrow}}w$ in G' then $A\overset{\varphi(\pi')}{\underset{L}{\Longrightarrow}}w$ in G.

Proof. Notice that $A\in N$. The proof is by induction on $|\pi'|$.

Basis. If $|\pi'|=1$ then $\varphi(\pi')=\pi'$ and the result is clear.

Induction. Assume $|\pi'|=m$, $m>1$. For $A\overset{\pi'}{\underset{R}{\Longrightarrow}}w$ we may write

$$A\overset{i'}{\underset{R}{\Longrightarrow}}aH_{ij_k}\gamma_k\gamma\overset{\rho'}{\underset{R}{\Longrightarrow}}aH_{ij_k}w'\overset{j'}{\underset{R}{\Longrightarrow}}abw',$$

where $i'\rho'j'=\pi'$ and $abw'=w$.

Since $|\rho'|<m$ and $\gamma_k\gamma\in N^*$ it is easily verified with the help of the induction hypothesis that $\gamma_k\gamma\overset{\varphi(\rho')}{\underset{L}{\Longrightarrow}}w'$ in G. Moreover, $\varphi(i')=\varepsilon$ and $\varphi(j')=j_k i$, with $j_k\cdot B\to b\gamma_k$ and $i\cdot A\to aB\gamma$. Hence, $A\overset{\varphi(\pi')}{\underset{L}{\Longrightarrow}}w$ in G. □

In the following claim we consider grammar G' before step (ii) of the algorithm is executed.

Claim 2. If $A \overset{\pi}{\underset{L}{\Longrightarrow}} w$ in G then there exists $\pi' \in \Delta_{G'}^*$ such that $A \overset{\pi'}{\underset{R}{\Longrightarrow}} w$ in G' and $\varphi(\pi') = \pi$.

Proof of Claim 2. The argument is similar to that of Claim 1. Notice that if $|\pi| > 1$ then we can write

$$A \overset{i}{\underset{L}{\Longrightarrow}} aB\gamma \overset{j}{\underset{L}{\Longrightarrow}} ab\gamma_k\gamma \overset{\rho'}{\underset{L}{\Longrightarrow}} abw' = w.$$

The details are left to the reader. □

In both claims we may take $A = S$ and we can conclude that $G'[\bar{r}/l]\, G$. □

Theorem 2.5 will be used in the construction of the cover-table.

3. Counter Example Grammar

In Ukkonen [33] it is shown that grammar G with productions

$$S \to 0SL \mid 0RL$$
$$R \to 1RL \mid 1$$
$$L \to \varepsilon$$

can not be left covered with an ε-free CFG. Now consider CFG G_0 with productions

1. $S \to 0SL$
2. $S \to 1RL$
3. $R \to 1RL$
4. $R \to 2$
5. $L \to \varepsilon$.

Clearly, if G does not have an ε-free CFG which left covers G then also G_0 does not have such a grammar. Grammar G_0 will turn out to be useful if we construct the cover table.[3]

In Table 1 we list the productions of a CFG G_N which is such that $G_N[\bar{r}/l]\, G_0$. Grammar G_N is in GNF and since $G_N[\bar{r}/l]\, G_0$ we may immediately conclude that G_N does not have an ε-free CFG G' such that $G'[l/\bar{r}]\, G_N$.

We have a few notes on special properties of the grammars G_0 and G_N. Grammar G_0 is both $LL(1)$ and *strict deterministic* (of *degree* 1) (cf. Harrison and Havel [9]). Therefore the following result is obvious.

Corollary 3.1. a) *Not every LL(k) grammar can be left covered with an ε-free grammar.*

b) *Not every strict deterministic grammar can be left covered with an ε-free grammar.*

---

3 In Ukkonen [33] not only grammar G but also other counter examples for possible cover results are given

Table 1. Grammar G_N

$S \to 0H_{00}S$	$\langle 55 \rangle$	$H_{00} \to 0$	$\langle 11 \rangle$
$S \to 0H_{01}R$	$\langle 55 \rangle$	$H_{01} \to 1$	$\langle 12 \rangle$
$S \to 1H_{11}R$	$\langle 55 \rangle$	$H_{11} \to 1$	$\langle 23 \rangle$
$S \to 1H_{12}$	$\langle 5 \rangle$	$H_{12} \to 2$	$\langle 24 \rangle$
$R \to 1Q_{11}R$	$\langle 55 \rangle$	$Q_{11} \to 1$	$\langle 33 \rangle$
$R \to 1Q_{12}$	$\langle 5 \rangle$	$Q_{12} \to 2$	$\langle 34 \rangle$
$R \to 2$	$\langle 4 \rangle$		

A consequence is that left covering GNF grammars can not be obtained. Notice that for $LL(k)$ grammars this result is in contradiction with excercise 8.1.20 in Aho and Ullman [1].

Both $LL(k)$ and strict deterministic grammars are $LR(k)$ grammars. Therefore the negative results hold for $LR(k)$ grammars as well.

Now we consider grammar G_N. This grammar is defined in such a way that it is both $LL(2)$ and strict deterministic.

Corollary 3.2. a) *Not every ε-free LL(k) grammar can be left-to-right covered with an ε-free grammar.*

b) *Not every ε-free strict deterministic grammar can be left-to-right covered with an ε-free grammar.*

Also in this case the results hold for $LR(k)$ grammars in GNF as well.

4. The Cover-Table

Once more we mention that the context-free grammars which we consider are cycle-free, they do not have useless symbols and if the empty word is in the language then there is exactly one leftmost derivation for this word. Moreover, we will not refer to the special production $S_0 \to S\bot$ which may be introduced in the case of elimination of single productions. The *cover-table* has five rows (ARB, ε-FREE, NLR, ε-FREE NLR, GNF) and seven columns (ARB, ε-FREE, NLR, ε-FREE NLR, GNF, NRR and ε-FREE NRR). Each row has four subrows, one for each type of cover which we consider (l/l, $l/\bar{r}$, $\bar{r}/l$ and $\bar{r}/\bar{r}$). We use a simple reference system to the entries of the table. Except for the ARB-row all places are labeled with either letters (*a.*, ..., *p.*) or numbers (1., ..., 96.).

The details of the construction of the table can be found in the Appendix. Example. Entry 25. is *no*. This means that not every ε-free grammar can be left covered with a NLR grammar.

5. Conclusions

In the present paper we have given an overview of cover results for some normal forms for context-free grammars. Similar cover results as obtained in this paper

Table 2. Cover-table

G' \ G	COVER	ARB	ε-FREE	NLR	ε-FREE NLR	GNF	NRR	ε-FREE NRR
ARB	l/l	yes	yes	yes	yes	yes	yes	yes
	$l/\bar{r}$	yes	yes	yes	yes	yes	yes	yes
	$\bar{r}/l$	yes	yes	yes	yes	yes	yes	yes
	$\bar{r}/\bar{r}$	yes	yes	yes	yes	yes	yes	yes
ε-FREE	l/l	a. no	1. yes	5. no	9. yes	13. yes	17. yes	21. yes
	$l/\bar{r}$	b. no	2. no	6. no	10. no	14. no	18. yes	22. yes
	$\bar{r}/l$	c. no	3. no	7. yes	11. yes	15. yes	19. no	23. no
	$\bar{r}/\bar{r}$	d. no	4. yes	8. yes	12. yes	16. yes	20. no	24. yes
NLR	l/l	e. no	25. no	29. yes	33. yes	37. yes	41. no	45. no
	$l/\bar{r}$	f. no	26. yes	30. yes	34. yes	38. yes	42. no	46. yes
	$\bar{r}/l$	g. no	27. no	31. yes	35. yes	39. yes	43. no	47. no
	$\bar{r}/\bar{r}$	h. no	28. yes	32. yes	36. yes	40. yes	44. no	48. yes
ε-FREE NLR	l/l	i. no	49. no	53. no	57. yes	61. yes	65. no	69. no
	$l/\bar{r}$	j. no	50. no	54. no	58. no	62. no	66. no	70. yes
	$\bar{r}/l$	k. no	51. no	55. yes	59. yes	63. yes	67. no	71. no
	$\bar{r}/\bar{r}$	l. no	52. yes	56. yes	60. yes	64. yes	68. no	72. yes
GNF	l/l	m. no	73. no	77. no	81. yes	85. yes	89. no	93. no
	$l/\bar{r}$	n. no	74. no	78. no	82. no	86. no	90. no	94. yes
	$\bar{r}/l$	o. no	75. no	79. yes	83. yes	87. yes	91. no	95. no
	$\bar{r}/\bar{r}$	p. no	76. yes	80. yes	84. yes	88. yes	92. no	96. yes

will be given in forthcoming papers for regular and deterministically parsable grammars.

The main problems which had to be solved in order to obtain the covertable of Sect. 4 were the elimination of left recursion, the elimination of ε-productions and the problem of finding a right covering grammar in Greibach normal form from an ε-free non-left-recursive grammar. It would be interesting to have a thorough comparison between results for grammar covers and for grammar functors. Unfortunately the elimination of left-recursion does not admit a grammar functor between the original and the non-left-recursive grammar (see e.g. [2]). This does not imply, as has become clear in Hotz [12, 13] that the grammar functor approach does not have useful applications when considering normal form transformations.

Acknowledgements. I would like to thank one of the referees for pointing out a few errors in the manuscript and for suggestions concerning the presentation of the results.

References

1. Aho AV, Ullman JD (1972/1973) The theory of parsing, translation and compiling, Vol. I and II, Prentice Hall, Englewood Cliffs, NJ
2. Benson DB (1977) Some preservation properties of normal form grammars. SIAM J Comput 6:381–402

3. Cremers AB, Ginsburg S (1975) Context-free grammar forms. J. Comput. System Sci. 11:86–117
4. Foster JM (1968) A syntax improving program. Comput J. 11:31–34
5. Foster JM (1970) Automatic syntactic analysis. MacDonald, London
6. Gray JN, Harrison MA (1969) Single pass precedence analysis, IEEE Conf. Record of the 10th Annual Symposium on Switching and Automata Theory, 106–117
7. Gray JN, Harrison MA (1972) On the covering and reduction problems for context-free grammars. J. Assoc. Comput. Mach. 19:385–395
8. Griffiths TV, Petrick SR (1965) On the relative efficiencies of context-free grammar recognizers. Comm. ACM 8, 289–300
9. Harrison MA, Havel IM (1973) Strict deterministic grammars. J. Comput. System Sci. 7, 237–277
10. Hotz G (1965) Eine Algebraisierung des Syntheseproblems von Schaltkreisen, I und II. Elektron Informationsverarbeit Kybernetik 1:185–231
11. Hotz G (1966) Eindeutigkeit und Mehrdeutigkeit formaler Sprachen. Elektron Informationsverarbeit Kybernetik 2:235–246
12. Hotz G (1978) Normal-form transformations of context-free grammars. Acta Cybernet. 4:65–84
13. Hotz G, Ross RJ (1979) $LL(k)$- und $LR(k)$-Invarianz von kontextfreien Grammatiken unter einer Transformation auf Greibach Normalform. Elektron Informationsverarbeit Kybernetik 15:73–86
14. Hunt HB, Rosenkrantz DJ (1977) Complexity of grammatical similarity relations. Proc. of the Conf. on Theoretical Computer Science, Waterloo 139–145
15. Hunt HB, Rosenkrantz DJ, Szymanski TG (1976) The covering problem for linear context-free grammars. Theor. Comput. Sci. 2:361–382
16. Hunt HB, Rosenkrantz DJ, Szymanski TG (1976) On the equivalence, containment and covering problems for regular and context-free languages. J. Comput. System Sci. 12:222–268
17. Kuno S (1966) The augmented predictive analyzer for context-free languages – Its relative efficiency. Comm. ACM 9:810–823
18. Kurki-Suonio R (1966) On top-to-bottom recognition and left recursion. Comm. ACM 9:527–528
19. Mickunas MD (1976) On the complete covering problem for $LR(k)$ grammars. J. Assoc. Comput. Mach. 23:17–30
20. Mickunas MD, Lancaster RL, Schneider VB (1976) Transforming $LR(k)$ grammars to $LR(1)$, $SLR(1)$ and $(1,1)$ bounded right context grammars. J. Assoc. Comput. Mach. 23:511–533
21. Nijholt A (1979) Grammar functors and covers: From non-left-recursive to Greibach normal form grammars. BIT 19:73–78
22. Nijholt A (1977) On the covering of left-recursive grammars. Conf. Record of the 4th ACM Symposium on Principles of Programming Languages 86–96
23. Nijholt A (1979) From left regular to Greibach normal form grammars. Information Processing Lett. 9:51–55
24. Nijholt A (1979) Structure preserving transformations on non-left-recursive grammars. Proc. 6th Int. Coll. on Automata, Languages and Programming. Maurer HA (ed) Springer, Berlin Heidelberg New York (Lecture Notes in Computer Science Vol. 71 p 446)
25. Nijholt A (1977) On the covering of parsable grammars. J. Comput. System Sci. 15:99–110
26. Nijholt A (1979) A survey of normal form covers for regular grammars. Informatica report 52, Vrije Universiteit, Amsterdam
27. Paull MC, Unger SH (1968) Structural equivalence of context-free grammars. J. Comput. System Sci. 2:427–463
28. Ross RJ, Hotz G, Benson DB (1978) A general Greibach normal form transformation. CS-78-048, Washington State University, Pullman
29. Soisalon-Soininen E (1979) On the covering problem for left-recursive grammars. Theor. Comput. Sci. 8:1–12
30. Stearns RE (1971) Deterministic top-down parsing. Proc. 5th Princeton Conf. on Information Sciences and Systems 182–188
31. Ukkonen E (1978) Transformations to produce certain covering grammars. Proc. 7th Int. Sympos. on Mathematical Foundations of Computer Science. Winkowski J (ed) Springer, Berlin Heidelberg New York (Lecture Notes in Computer Science Vol. 84 p 516)
32. Ukkonen E (1979) Remarks on the non-existence of some covering grammars. Proc. 4th GI Conference on Theoretical Computer Science. Weihrauch K (ed) Springer, Berlin Heidelberg New York (Lecture Notes in Computer Science Vol. 67 p 298)

33. Ukkonen E (1978) The non-existence of some covering context-free grammars. Information Processing Lett. 8:187–192
34. Wood D (1969) The normal form theorem – another proof. Comput. J. 12:139–147

Received April 19, 1979; revised February 11, 1980

Appendix

In this Appendix we give the details of the construction of the cover-table (Table 2) of Sect. 4.

A) All the l/l and $\bar{r}/\bar{r}$ entries of the ARB-row are trivially *yes*. The $l/\bar{r}$ and $\bar{r}/l$ entries are *yes* because of Theorem 2.1.

B) Trivially *yes* are also the entries 1., 4., 9., 12., 13., 16., 21., 24., 29., 32., 33., 36., 37. and 40. Because of Theorem 2.1 and Observation 2.2 the entries 30., 31., 34., 35., 38. and 39. are *yes*. Trivially yes are also the entries 57., 60., 61., 64., 85. and 88.

C) Due to grammar G_0 we have that entry a. is *no* and from 'symmetry' it follows that entry d. is *no*. Therefore, also i., l., m. and p. are *no*. Since G_0 is NLR it follows that entry 5. is *no* and again from 'symmetry' entry 20. is *no*. Thus, entries 68. and 92. are *no*.

D) Next we consider grammar G_N. This grammar has the property that $G_N[\bar{r}/l]\,G_0$. Since G_0 has no ε-free grammar which left covers G_0 it follows that G_N does not have an ε-free grammar which left-to-right covers G_N. Moreover, G_N is in GNF, hence, the entries 14., 10., 6., 2. and b. are all *no*. Because of 'symmetry' it follows that the entries c., 3., 19. and 23. are *no*.

We have the following immediate consequences.

(i) Since entries b. and c. are *no* it follows that entries j., k., n. and o. are *no*.

(ii) Since entries 2. and 3. are *no* it follows that entries 50., 51., 74. and 75. are *no*.

(iii) Since entries 5. and 6. are *no* it follows that entries 53., 54., 77. and 78. are *no*.

(iv) Since entries 10. and 14. are *no* it follows that entries 58., 82., 62. and 86. are *no*.

(v) Since entries 19. and 23. are *no* it follows that entries 67., 91., 71. and 95. are *no*.

E) Due to the Corollaries 2.1 and 2.3 the entries 26., 28. and 52. are *yes*. From Theorem 2.3 it follows that entry 81. is *yes*. From Theorem 2.4 it follows that entries 76., 84. and 96. are *yes*. Since entry 96. is *yes* it follows that entries 72. and 48. are *yes*. From Corollary 2.4 it follows that entry 94. is *yes* and, consequently, entries 70., 46. and 22. are *yes*. Since the entries 81. and 85. are *yes* Theorem 2.5 tells us that entries 83. and 87. are *yes* and, consequently, entries 59., 11., 63. and 15. are *yes*.

With some simple observations, in which Theorem 2.5 can be used to obtain contradictions, it follows that the entries 73., 90., 93. and 89. are *no*.

Since the entries 73., 89., 93. and 90. are *no* we have that entries 49., 69., 65. and 66. are *no*. Otherwise a contradiction with Theorem 2.3 can be obtained.

F) Because of Corollary 2.2 we have that entry 8. is *yes* and from 'symmetry' it follows that entry 17. is *yes*. The assumption that entries h. and g. are *yes* leads, with the help of Corollary 2.2, to a contradiction with entries d. and c. are *no*, respectively. Similarly, with Corollary 2.2 and since entry 31. is *yes*, we must conclude that entries f. and e. are *no* in order to avoid contradictions with h. and g., respectively.

Since both entry 19. and entry 20. are *no* we obtain with the same type of argument that entries 41., 42., 43. and 44. are *no*. Entry 56. is *yes* because of entries 8. and 52. are *yes*.

The entries 25. and 27. are both *no* since otherwise a contradiction can be obtained (via entry 31. and 56. in the case of entry 25. and via entry 56. in the case of entry 27.) with entry 3. is *no*. For any NLR grammar G there exists a NLR grammar G' such that $G'[\bar{r}/l]\,G$. Grammar G' has an ε-free NLR grammar G'' such that $G''[\bar{r}/l]\,G$. Hence, entry 55. is *yes* and therefore also entry 7. is *yes* and ('symmetry') entry 18. is *yes*.

Since entries 55. and 56. are *yes* it follows (with entry 84. is *yes*) that entries 79. and 80. are *yes*.

Both entries 45. and 47. are *no* because otherwise, with the help of 55. and 56., a contradiction with entry 71. is *no* is obtained. This concludes the construction of the cover-table.