COVER RESULTS AND NORMAL FORMS

Anton Nijholt
Vrije Universiteit
Department of Mathematics
P.O.Box 7161, Amsterdam
The Netherlands.

1. INTRODUCTION.

It has been recognized that the idea of *covering* of grammars is very useful. Since the first time definitions of cover appeared in the literature (see Aho & Ullman [1] and Gray & Harrison [4]) several papers have been written in which (variants of) these definitions were used. Consider two context-free grammars. We can talk about a relationship between these grammars. For example, we may conclude that both grammars are LR(k) grammars and therefore their sentences can be parsed with an LR(k) parsing method. Another type of relationship between these two grammars may be that they generate the same language, i.e. they are *equivalent*. A more restrictive type of relationship is that one grammar *covers* the other, which means that not only the grammars are equivalent, but also that their parse trees are close related. Intuitively we say that a context-free grammar (cfg) G' covers a cfg G, when the ability to parse G' allows one to parse G. Hence we can look for grammars which are 'easily parsable' and which cover grammars which are more 'difficult parsable'. In general the cover-relationship between two grammars is expressed by a homomorphism between parses.

In Gray & Harrison [4] results were obtained for the covering of context-free grammars by grammars which are in a certain normal form. The question, which was stated as an open problem in Aho & Ullman [1] and Harrison [7], whether each LR(k) grammar is (right) covered by an LR(1) or LR(0) grammar found its answer in Mickunas [14], Mickunas, Lancaster & Schneider [15] and Nijholt [17]. Some decidability results for covering appeared in Hunt [8] and in Hunt, Rosenkrantz & Szymanski [9], [10]. More results on covering appeared, sometimes informally, in Aho & Ullman [1], Graham [3], Hammer [5], McAfee & Presser [13] and Nijholt [18]. In Nijholt [16] it was proved that every proper cfg can be right covered by a non-left-recursive grammar.

However, a lot of problems have not yet been investigated. In the present paper we try to collect the, in our eyes at this moment, most interesting problems on covering and we give answers to some of them. The most important results in this paper can be obtained in rather simple ways from existing literature. First we show that in spite of some remarks in the literature the possibility to cover context-free grammars by context-free grammars in Greibach normal form is an open question. Another result we present says that each LR(k) grammar is right covered by a non-left-recursive LR(1) grammar or (in case the language is prefix-free) by a strict deterministic grammar (notice that strict deterministic grammars are not left-recursive).

The organization of this paper is as follows. This section concludes with some preliminaries. In the second section we present some results and open problems on the covering of (arbitrary) cfg's by cfg's in GNF (Greibach normal form) and by cfg's in a non-left-recursive form. In the third section we show, and illustrate with examples, some properties of the relationship between *parsability* and *covers*. In the fourth section we have some remarks and results for cover problems for the class of LR(k) grammars and some of its subclasses.

PRELIMINARIES

Definition 1.1 (normal forms)

A *context-free grammar* (cfg) is denoted by the four-tuple $G = (N,T,P,S)$. We make the following conventions. Elements of T are denoted by a,b,c, etc.; u,v,w, etc. denote elements of T^*; A,B,C, etc., denote elements of N; X,Y,Z, etc. denote elements of $V = N \cup T$; and α,β,γ, etc., denote elements of $(N \cup T)^*$. The empty string is denoted by ε. The notation $\alpha \overset{*}{\underset{\ell}{\Longrightarrow}} \beta$ is used for a *leftmost derivation* of β from α; $\alpha \overset{*}{\underset{r}{\Longrightarrow}} \beta$ denotes a *rightmost derivation*.

A cfg G is said to be *unambiguous* if each sentence has exactly one leftmost derivation. Otherwise G is said to be ambiguous. G is said to be *cycle-free* if there is no derivation $A \overset{+}{\Longrightarrow} A$, for any $A \in N$. Cfg G is said to be ε*-free* if there are no productions, except for $S \rightarrow \varepsilon$, of the form $A \rightarrow \varepsilon$ in P. A nonterminal A is said to be *left-recursive* if $A \overset{+}{\Longrightarrow} A\alpha$ for some $\alpha \in V^*$. A cfg G is said to be left-recursive if there is at least one left-recursive nonterminal. Cfg G is in pseudo- *Greibach normal form* (pseudo-GNF) if every production is of the form $A \rightarrow a\alpha$, where $a \in T$ and $\alpha \in V^*$. If $\alpha \in N^*$ then G is said to be in GNF.

Definition 1.2. (homomorphism)

Let T_1 and T_2 be two alphabets. Let f be a function, $f: T_1 \rightarrow T_2^*$;f is extended to a homomorphism $f': T_1^+ \rightarrow T_2^*$ by letting $f'(a_1a_2\ldots a_n) = f(a_1)f(a_2)\ldots f(a_n)$, for $a_1a_2\ldots a_n \in T_1^+$; f' is said to be *fine* if for each $a \in T_1$, $f'(a) \in T_2 \cup \{\varepsilon\}$.

Definition 1.3. (cover)

Let G' and G be cfg's, G' = (N',T,P',S') and G = (N,T,P,S). We say that G' *right covers* G under cover-homomorphism h: $P'^{+} \to P^{*}$, if for all w in T^{*}

(i) if $S' \xRightarrow[r]{\pi'} w$, then $S \xRightarrow[r]{h(\pi')} w$, and

(ii) if $S \xRightarrow[r]{\pi} w$, then there exist π' such that $S' \xRightarrow[r]{\pi'} w$ and h(π') = π.

Analogously the notion of *left cover* is defined. Moreover, if left parses with respect to G' are mapped on right parses (i.e. the concatenation of productions used in a rightmost derivation but in a reversed order) with respect to G, then we say G' *left-to-right* covers G. Analogously the notion of *right-to-left* cover is defined.

Other definitions and notations will be given on the places where they are needed or the reader is referred to literature. All cfg's in this paper are assumed to be *reduced*.

2. TO COVER OR NOT TO COVER.

Before we give in this and in coming sections our, in general rather negative, results on the covering of context-free grammars, we want to start with a more positive result.

Consider the following cfg G_0 with only productions

$$0./1./2./3./\ S \to S0|S1|0|1$$

In Aho & Ullman [1,p.280] it is stated in a problem that G_0 can not be right covered by a cfg in GNF under an arbitrary cover-homomorphism, moreover, according to the following problem given there, even if we replace the homomorphism in the definition of cover by a finite transducer mapping there is no such a cover. Intuitively we agreed with this, but in our paper Nijholt [16] we asked for a proof. There is no such proof. The following cfg G is in GNF and right covers G_0. Below we list the productions of G; the start symbol is S', each production is followed by its image under the cover-homomorphism.

0. $S' \to 0$	*(2)*	10. $S \to 0$	*(0)*	20. $C \to 1$	*(11)*
1. $S' \to 1$	*(3)*	11. $S \to 1$	*(1)*	21. $F \to 1$	*(10)*
2. $S' \to 0D'$	*(ε)*	12. $S \to 0D''$	*(ε)*	22. $C'' \to 0$	*(01)*
3. $S' \to 0F'$	*(ε)*	13. $S \to 1C''$	*(ε)*	23. $C'' \to 1$	*(11)*
4. $S' \to 1E'$	*(ε)*	14. $S \to 0DS$	*(ε)*	24. $D'' \to 0$	*(00)*
5. $S' \to 1C'$	*(ε)*	15. $S \to 0FS$	*(ε)*	25. $D'' \to 1$	*(10)*
6. $S' \to 1E'S$	*(ε)*	16. $S \to 1CS$	*(ε)*	26. $E' \to 0$	*(03)*
7. $S' \to 1C'S$	*(ε)*	17. $S \to 1ES$	*(ε)*	27. $D' \to 0$	*(02)*
8. $S' \to 0D'S$	*(ε)*	18. $E \to 0$	*(01)*	28. $C' \to 1$	*(13)*
9. $S' \to 0F'S$	*(ε)*	19. $D \to 0$	*(00)*	29. $F' \to 1$	*(12)*

Table I. Productions for G'.

The proof that G right covers cfg G_0 is straightforward and is therefore omitted. Although the long list of productions suggests the contrary G can be derived from G_0 in a rather intuitive way. In Gray & Harrison [4] there is a theorem which states that cfg G_0 can not be right covered by a cfg in GNF (pseudo-GNF) under a fine cover-homomorphism. Their proof is not correct since their claim 3 is incorrect. However, to show that there exist cfg's which cannot be right (or left) covered by a cfg in GNF we can look at more simple grammars. For example, the unambiguous cfg G_1 with only productions 0. $S \to A$ and 1.$A \to a$ can not be right (left) covered by a cfg G' = (N',T,P',S') under a fine cover-homomorphism h, since such a cover-homomorphism should map the only production $S' \to a$ on 01, hence h can not be fine.

Corollary 2.1

Not every cfg can be right (left) covered by a cfg in GNF under a fine cover-homomorphism.

Arbitrary cover-homomorphisms (i.e. not necessarily fine) lead to more interesting problems. First we list a few properties of covers.

Notation: Let $\alpha \in V^*$ then α^R is the string α written in reversed order.

Lemma 2.1.

If G' right (left) covers G then the degree of ambiguity of G' (see Aho & Ullman [1]) is greater then or equal to the degree of ambiguity of G.

Proof. Follows directly from the definition of cover.□

Observation 2.1.

Clearly there is a close relation between right covers (which are defined for rightmost derivations) and mappings of right parses. If G' right covers G under h then right parses of G' can be mapped on right parses of G. Define h' as: for each

$i \in P'$, if $h(i) = \rho$ then $h'(i) = \rho^R$. If G' right covers G under h then we have that

(i) if $S' \xrightarrow[r]{\pi'} w$ then $S \xrightarrow[r]{h(\pi')} w$, which means, if we let $\pi = h(\pi')$, that $h'(\pi'^R) = \pi^R$ and

(ii) if $S \xrightarrow[r]{\pi} w$, then there exists π' such that $S' \xrightarrow[r]{\pi'} w$, where $h(\pi') = \pi$, i.e. $h'(\pi'^R) = \pi^R$.□

The proof of the following lemma is straightforward and therefore omitted.

Lemma 2.2.
A cfg G is right (left) covered by a cfg in pseudo-GNF iff G is right (left) covered by a cfg in GNF.

Can each cfg be right (left) covered (under an arbitrary cover-homomorphism) by a cfg in GNF? Consider the following cfg G_2 with only productions $S \to A|a$ and $A \to S$. Clearly a cfg in GNF which rightcovers G_2 does not exist. The same result holds for the (also ambiguous) cfg G_3 with only productions.

$$S \to A|B,\ A \to a \text{ and } B \to a.$$

Corollary 2.2.
Not every cfg can be right (left) covered by a cfg in GNF.

There remain the following questions. Can each unambiguous cfg be right (left) covered by a cfg in GNF? Can each ε-free unambiguous cfg be right (left) covered by a cfg in GNF? If there are at least two ways to derive ε in a cfg then clearly this cfg can not be right (left) covered by an ε-free grammar.

Corollary 2.3.
Not every cfg can be right (left) covered by an ε-free cfg.

There remains the question: Can each unambifuous cfg be right (left) covered by an ε-free cfg?
Instead of GNF we can consider the less restricted class of cfg's which are not left-recursive. Then we have from Nijholt [16] the following result.

Corollary 2.4.
Each cfg which is ε-free and cycle-free is right covered by a non-left-recursive cfg.

The following lemma can easily be obtained from the usual tranformation of a non-left-recursive cfg to a cfg in GNF (see for example Aho & Ullman [1]), therefore the proof is omitted.

Lemma 2.3.
Each unambiguous and ε-free non-left-recursive grammar is left covered by a cfg in GNF.

Notice that the condition of unambiguity is necessary, see for example the non-left-recursive cfg G_1 mentioned above which cannot be left covered by a cfg in GNF.
In section 4 we return to some of the questions here but then for more restricted classes of cfg's.

3. PARSABILITY AND COVERS.

For the formal definitions of some notions in this section we refer the reader to Aho & Ullman [1] and Nijholt [17]. A *deterministic pushdown transducer* P (dpdt) is said to be a *valid* dpdt for cfg G if P acts as a *parser* for G. P may for example act as a left parser (producing left parses) or as a right parser (producing right parses). If for a cfg G there exists a valid dpdt then G is said to be a *parsable* grammar (*left parsable, right parsable*).

Examples. The cfg G_4 with only productions

$S \to BAb|CAc \qquad B \to a$
$A \to BA|a \qquad C \to a$

is a left parsable grammar. The cfg G_5 with only productions

$S \to Ab|Ac \qquad B \to a$
$A \to AB\ a$

is a right parsable grammar. It is not difficult to prove that G_4 is not right parsable and G_5 is not left parsable.

We can use the idea of parsable grammars to show the impossibility of certain covers.

Lemma 3.1.
(i) Suppose cfg G is not left parsable. Then G cannot be left covered by a left parsable grammar.
(ii) Suppose cfg G is not right parsable. Then G cannot be right covered by a right parsable grammar.

Proof. (sketch) Part(i). Suppose there exists G', G' left covers G under cover-homomorphism h and G' is left parsable. Hence there exists a valid dpdt P' for G' which acts as a left parser. By applying h to the output of P' we obtain a new dpdt P which is, since G' left covers G, a valid dpdt for G, hence G is left parsable. This contradicts the assumption that G is not left parsable.

Therefore we must conclude that G cannot be left covered by a left parsable grammar. Part (ii) goes analogously.□

With two examples we show the use of this lemma.

Example 3.1. In Nijholt [19] the definition of a *simple chain grammar* was introduced. Let $G = (N,T,P,S)$be an ε-free grammar. Let $X_0 \in V$, then $CH(X_0) = \{<X_0X_1\ldots.X_n> | X_0X_1\ldots.X_n \in N^*T \ \& \ X_0 \underset{\ell}{\Longrightarrow} X_1\psi_1 \underset{\ell}{\Longrightarrow} \ldots \underset{\ell}{\Longrightarrow} X_n\psi_n$, where $\psi_i \in V^*$, $1 \leq i \leq n\}$. If $\pi = <X_0X_1\ldots X_n> \in CH(X_0)$ then $l(\pi) = X_n$. V is said to be chain-independent if for all X in V, if π_1,π_2 in CH(X) and $\pi_1 \neq \pi_2$ then $l(\pi_1) \neq l(\pi_2)$. Let $X,Y \in V$, $X \neq Y$. We write $X \ddagger Y$ if for each pair $\pi_1 \in CH(X)$ and $\pi_2 \in CH(Y)$ we have that $l(\pi_1) \neq l(\pi_2)$. We use this notation also if V is chain-independent, then if π_1,π_2 in CH(X) we have $l(\pi_1) \neq l(\pi_2)$, hence $X \ddagger X$. A set of productions P is prefix-free if $A \rightarrow \alpha$ and $A \rightarrow \alpha\beta$ in P implies $\beta = \varepsilon$. A cfg $G = (N,T,P,S)$ is said to be a *simple chain grammar* if V is chain-independent, P is prefix-free and for each pair productions $A \rightarrow \alpha X\phi$ and $A \rightarrow \alpha Y\psi$, where $X \neq Y$, we have $X \ddagger Y$.

In Nijholt [20] it is shown that each simple chain grammar can be transformed to a simple LL(1) grammar (i.e. a cfg which satisfies (i) each production is of the form $A \rightarrow a\phi$ and (ii) if $A \rightarrow a\phi$ and $A \rightarrow b\psi$ then $a \neq b$ or $a\phi = b\psi$).

Now consider the cfg G with only productions

$$S \rightarrow aEc \mid aEd \quad \text{and} \quad E \rightarrow aEb \mid ab.$$

One can easily verify that G satisfies the conditions of a simple chain grammar and moreover that G is not a left parsable grammar. Therefore, with lemma 3.1. we can immediately conclude that there is no transformation from simple chain grammars to simple LL(1) grammars which yields a left cover.

Example 3.2. In Hammer [5] the class of *k-transformable* grammars is introduced, a subclass of the LR(k) grammars. Moreover, a transformation is presented from k-transformable grammars to (strong) LL(k) grammars. Consider the following k-transformable grammar G from that paper, with only productions

$$S \rightarrow bAc \qquad A \rightarrow ABx \mid ABy \mid a \qquad B \rightarrow Bd \mid d$$

Again, one can easily verify that G is not a left parsable grammar. Therefore, with lemma 3.1, we can conclude immediately that there is no transformation from k-transformable to LL(k) grammars which yields a left cover.

The result of example 3.1. is rather surprising. An extremely simple transformation can yield a simple LL(1) grammar. For example, replace

$$S \rightarrow aEc \mid aEd \quad \text{and} \quad E \rightarrow aEb \mid ab$$

by

$$S \rightarrow aED, \quad E \rightarrow aEb \mid ab \quad \text{and} \quad D \rightarrow c \mid d.$$

This new grammar does not left cover the original grammar. Moreover, with the same type of argument, there is no LL(k) grammar which left covers the original grammar.

4. COVERS AND DETERMINISTIC GRAMMARS.

In this last section we give some remarks on problems and results for the covering of LR(k) grammars and of grammars belonging to subclasses of the class of LR(k) grammars. In the preceeding section we already saw two (negative) results. From example 3.1. it follows that not every LR(k) grammar which generates an LL(k) language has an left covering LL(k) grammar.
In Lomet [12] and in Geller, Harrison & Havel [2] it is shown that each LR(k) language may be given an LR(1) grammar in GNF. Moreover each SD-grammar (*Strict Deterministic* grammar, see Harrison & Havel [6]) can be transformed to a SD-grammar in GNF. Therefore we ask the same questions as we did in section 2, i.e. can each SD-grammar be right covered by a SD-grammar in GNF?; can each LR(k) grammar be right covered by an LR(1) grammar in GNF? Questions for which we have no answers yet. However, trivially we obtain again (see for example cfg G_1 of section 2) that for a fine cover-homomorphism the answers are no. Consider also the following properties. First recall that SD-grammars are not left-recursive. In Nijholt [17] it is shown that each LR(k) grammar G can be transformed to an LR(1) grammar (or in case L(G) is prefix-free to a SD-grammar) which right covers G. Moreover, although not mentioned there, it can easily be verified that the LR(1) grammar which is obtained is non-left-recursive.

Corollary 4.1.
Each LR(k) grammar G is right covered by a non-left-recursive LR(1) grammar, or in case L(G) is prefix-free by a SD-grammar.

The following result can also be obtained from Nijholt [17]; here we prefer to use some other results. Let G = (N,T,P,S) be an LL(k) grammar. Let p be the total number of productions in P. Then construct a new cfg G' = (N',T,P',S) where
$N' = N \cup \{H_i \mid 1 \leq i \leq p\}$ (the H_i's are newly introduced nonterminals);
$P' = P \cup \{H_i \rightarrow \varepsilon \mid 1 \leq i \leq p\}$. In Hunt III & Szymanski [11] it is proved that G' is LL(k) if and only if G is LL(k). One can easily prove that G' right-to-left covers G. Since G' is LL(k) and hence LR(k) we have

Corollary 4.2.
Each LL(k) grammar is right-to-left covered by an LR(k) grammar.
In this corollary we can replace, with the aid of corollary 4.1. LR(k) by LR(1) (or SD). The last result in this section is obtained from Geller, Harrison & Havel [6]. The transformation given there to obtain a SD-grammar in GNF from a SD-grammar yields a left cover.

Corollary 4.3.

Each ε-free SD-grammar is left covered by a SD-grammar in GNF.

5. CONCLUSIONS.

The purpose of this paper was to sketch an area of problems for the concept of cover. We showed that in spite of some remarks in the literature the problem of covering (unambiguous and ε-free) cfg's with cfg's in GNF is open. Moreover we gave some properties of covers and we showed a relation between covers and parsability.

References.

1. Aho A.V. and Ullman J.D., The Theory of Parsing, Translation and Compiling, Vols. I and II, Prentice Hall, Englewood Cliffs, New Jersey, 1972 and 1973.

2. Geller M.M., Harrison M.A. and Havel I.M., Normal forms of deterministic languages, Discrete Mathematics, Vol. 16, pp. 313-322, 1976.

3. Graham S.L., On bounded right context languages and grammars, SIAM Journal on Computing, Vol. 3, pp. 224-254, 1974.

4. Gray J.N. and Harrison M.A., On the covering and reduction problems for context-free grammars, Journal of the Association for Computing Machinery, Vol. 19, pp. 675-698, 1972.

5. Hammer M, A new grammatical transformation into LL(k) form, Conference Record of the 6th annual ACM Symposium on Theory of Computing, pp. 266-275, 1974.

6. Harrison M.A. and Havel I.M., Strict deterministic grammars, Journal of Computer and System Sciences, Vol. 7, pp. 237-277, 1973.

7. Harrison M.A., On covers and precedence analysis, Lecture Notes in Computer Science 1, G.I. 3. Jahrestagung, pp. 2-17, 1973.

8. Hunt III H.B., A complexity theory of grammar problems, Conference Record of the 3rd ACM Symposium on Principles of Programming Languages, pp. 12-18, 1976.

9. Hunt III H.B., Rosenkrantz D.J. and Szymanski T.G., The covering problem for linear context-free grammars, Theoretical Computer Science, Vol. 2, pp. 361-382, 1976.

10.Hunt III H.B., Rosenkrantz D.J. and Szymanski T.G., On the equivalence, containment, and covering problems for the regular and context-free languages, Journal of Computer and System Sciences, Vol. 12, pp. 222-268, 1976.

11.Hunt III H.B., Szymanski T.G., Lower bounds and reductions between grammar problems, Technical Report 216, Princeton University, 1976.

12.Lomet D.B., A formalization of transition diagram systems, Journal of the Association for Computing Machinery, Vol. 20, pp. 235-257, 1973.

13.McAfee J. and Presser L., An algorithm for the design of simple precedence grammars, Journal of the Association for Computing Machinery, Vol. 19, pp. 385-395, 1972.

14.Mickunas M.D., On the complete covering problem for LR(k) grammars, Journal of the Association for Computing Machinery, Vol. 23, pp. 17-30, 1976.

15.Mickunas M.D., Lancaster R.L. and Schneider V.B., Transforming LR(k) grammars to LR(1), SLR(1) and (1,1) Bounded Right Context grammars, Journal of the Association for Computing Machinery, Vol. 23, pp. 511-533, 1976.

16.Nijholt A., On the covering of left-recursive grammars, Conference Record of the 4th ACM Symposium on Principles of Programming Languages, pp. 86-96, 1977.

17.Nijholt A., On the covering of parsable grammars, to appear in Journal of Computer and System Sciences.

18.Nijholt A., On the parsing of LL-Regular grammars, Lecture Notes in Computer Science 45, Proc. 5th Int. Symposium on Mathematical Foundations of Computer Science, pp. 446-452, 1976.

19.Nijholt A., Simple Chain Grammars, Proc. 4th Int. Conference on Automata, Languages and Programming, 1977 (to appear).

20.Nijholt A., Simple Chain Languages, manuscript, march 1977.