A GENERAL SCHEME FOR SOME DETERMINISTICALLY PARSABLE GRAMMARS AND THEIR STRONG EQUIVALENTS
(Extended Abstract)

Anton Nijholt
Twente University of Technology
Dept. of Computer Science
PO Box 217, 7500 AE Enschede
The Netherlands

Jan Pittl
Research Inst. Math. Machines
Loretanske Nam. 3
11855 Prague 1
Czechoslovakia

1. INTRODUCTION

In the past years there have been many attempts to fill in the gap between the classes of LL(k) and LR(k) grammars with new classes of deterministically parsable grammars. Almost always the introduction of a new class was accompanied by a parsing method and/or a grammatical transformation fitting the following scheme. If parsers were at the centre of the investigation the new method used to be designed to possess certain advantages with respect to already existing ones. As far as transformations were concerned the intention was to produce methods of transforming grammars into "more easily" parsable ones.

The problem of finding classes of context-free grammars which can be transformed to LL(k) grammars has received much attention. Parsing strategies and associated classes of grammars generating LL(k) languages have been extensively studied (among others cf. e.g. [6,14,18]). An equally interesting class of grammars is the class of strict deterministic grammars [8,9], a subclass of the LR(0) grammars with elegant theoretical properties. Generalizations of this concept have been introduced by Friede[3] and Pittl[15]. The purpose of this paper is to show how the above mentioned classes of grammars can be dealt with within a general framework originated by Nijholt[12]. Roughly speaking, we study the phenomena corresponding e.g. to the relationship between strong LL(k) and LL(k) grammars. A general scheme using adjectives "strong" and "weak" is shown to be applicable for the description of the grammar families under consideration.

PRELIMINARIES

In the remainder of this section we review several concepts of formal language theory. The reader is referred to Aho and Ullman[1] or Harrison[7] for further details.

A context-free grammar (abbreviated a CFG) is denoted by $G = (N,T,P,S)$. Define $V = N \cup T$. Troughout the paper we assume all the grammars under consideration to be reduced.

Let $\alpha \in V^*$. The length of the word α is denoted by $|\alpha|$; the symbol Λ is reserved for the empty string. For any nonnegative integer k the expression $k : \alpha$ denotes α if $|\alpha| < k$, otherwise the prefix of α of length k. Furthermore we define

$$T^{*k} = \{u \in T^* \mid |u| \leq k\}.$$

The following operations relate to derivations in G. For any $\alpha \in V^*$ and $A \in N$ we define

$$FIRST_k(\alpha) = \{u \in T^{*k} \mid \alpha \overset{*}{\Longrightarrow} w \text{ and } k : w = u \text{ for some } w \in T^*\}$$

$$FOLLOW_k(A) = \{u \in T^{*k} \mid S \overset{*}{\Longrightarrow} \beta A \gamma \text{ and } u \in FIRST_k(\gamma) \text{ for some } \beta, \gamma \in V^*\}$$

The $FIRST_k$ operator can be extended to handle subsets X of V^*:

$$FIRST_k(X) = \{u \in T^{*k} \mid u \in FIRST_k(\alpha) \text{ for some } \alpha \in X\}$$

Finally, we recall the definitions of four wellknown classes of grammars. The first two concepts we present describe grammars introduced by Rosenkrantz and Stearns[17].

DEFINITION 1.1. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. The grammar G is called an LL(k) grammar iff for all $A \in N$, $w \in T^*$ and $\alpha, \beta, \gamma \in V^*$, if

$$S \overset{*}{\underset{L}{\Longrightarrow}} wA\alpha \underset{L}{\Longrightarrow} w\beta\alpha$$

$$S \overset{*}{\underset{L}{\Longrightarrow}} wA\alpha \underset{L}{\Longrightarrow} w\gamma\alpha$$

$$FIRST_k(\beta\alpha) \cap FIRST_k(\gamma\alpha) \neq \emptyset$$

then $\beta = \gamma$.

DEFINITION 1.2. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. G is called a strong LL(k) grammar iff for any $A \in N$ and $\beta, \gamma \in V^*$, if $A \to \beta$ and $A \to \gamma$ are in P then $FIRST_k(\beta FOLLOW_k(A)) \cap FIRST_k(\gamma FOLLOW_k(A)) \neq \emptyset$ implies $\beta = \gamma$.

Among various (and different) definitions of LR(k) grammars we have chosen the one due to Geller and Harrison[5].

DEFINITION 1.3. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. The grammar G is said to be LR(k) iff $S \overset{+}{\Longrightarrow} S$ impossible and for all $A, A' \in N$; $\alpha, \alpha', \beta, \beta', \gamma \in V^*$ and $w, w', x \in T^*$, if

$$S \overset{*}{\underset{R}{\Longrightarrow}} \alpha A w \underset{R}{\Longrightarrow} \alpha\beta w = \gamma w$$

$$S \overset{*}{\underset{R}{\Longrightarrow}} \alpha' A' x \underset{R}{\Longrightarrow} \alpha'\beta' x = \gamma w'$$

and $k : w = k : w'$ then $(A \to \beta, |\alpha\beta|) = (A' \to \beta', |\alpha'\beta'|)$.

Before we can give the definition of the fourth class to be dealt with we need a few preliminaries. Let Q be a set. A weak partition of Q is a set π of nonempty subsets of Q such that for each $q \in Q$ there is some $B \in \pi$ such that $q \in B$. The elements of π are called blocks of π. For $p, q \in Q$ we write $p \equiv q \pmod{\pi}$ iff $p \in B$ and $q \in B$ for some block B of π. A weak partition π of Q is called a partition of Q iff its blocks are pairwise disjoint. The following grammar family was introduced by Harrison and Havel[8].

DEFINITION 1.4. Let $G = (N,T,P,S)$ be a CFG, let π be a partition of $V = N \cup T$. Such a partition is called strict iff T forms a block of π and for all $A, A' \in N$ and $\alpha, \beta, \beta' \in V^*$, if $A \to \alpha\beta$, $A' \to \alpha\beta'$ are in P and $A \equiv A' \pmod{\pi}$ then either

(i) both $\beta, \beta' \neq \Lambda$ and $^{(1)}\beta \equiv {}^{(1)}\beta' \pmod{\pi}$, or

(ii) $\beta = \beta' = \Lambda$ and $A = A'$.

DEFINITION 1.5. A CFG $G = (N,T,P,S)$ is called strict deterministic iff there exists a strict partition π of V.

2. GRAMMARS TRANSFORMABLE TO LL(k) GRAMMARS

In this section we shortly review some definitions of classes of grammars which have the property that they can be transformed to LL(k) grammars. We do not go into proofs or historical details. These can be found in Nijholt[12] and in Soisalon-Soininen and Ukkonen[18].

In order to intuitively characterize the different classes of grammars to be defined we give an intuitive idea of their parsing strategies. In Figure 1 we have displayed the following situation. There exist terminal strings w, x, y and z, a nonterminal A and symbols $X_1, X_2, \ldots, X_p$ in V, such that $A \to X_1 \ldots X_p$ is a production and there exist derivations

$$S \overset{*}{\Longrightarrow} wAz, \quad X_1 \overset{*}{\Longrightarrow} x, \text{ and } \quad X_2 \ldots X_p \overset{*}{\Longrightarrow} y.$$

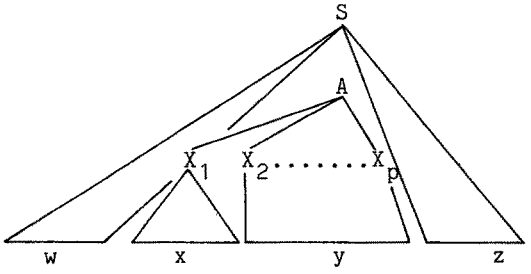

Figure 1. Parsing strategies.

In the following table we have collected six parsing strategies which are illustrated with the help of Figure 1. The following abbreviations are used:

LL : reading from the left using left parses [17]
PLC: predictive left corner grammars [12]
LP : left part grammars [12,14]
LC : left corner grammars [18]
PLR: predictive LR-grammars [18]
LR : reading from the left using right parses [5]

GRAMMAR	READ	RECOGNITION of A	READ	RECOGNITION of $A \to X_1 \ldots X_p$
LL	w	k : xyz	w	k : xyz
PLC	w	k : xyz	wx	k : yz
LP	w	k : xyz	wxy	k : z
LC	wx	k : yz	wx	k : yz
PLR	wx	k : yz	wxy	k : z
LR	wxy	k : z	wxy	k : z

Table I. Parsing strategies.

With the help of Figure 1 the table should be read as follows. Consider the terminal string wxyz. The production $A \to X_1X_2 \ldots X_p$ depicted in this parse tree of wxyz can be recognized with certainty after scanning

(i) w and k : xyz if the grammar is LL(k)
(ii) wx and k : yz if the grammar is PLC(k) or LC(k)
(iii) wxy and k : z if the grammar is LP(k), PLR(k) or LR(k)

However, if the grammar is PLC(k) or LP(k), then the lefthand side A of the production $A \to X_1X_2 \ldots X_p$ is already recognized after scanning w and k : xyz. If the grammar is PLR(k), then A is recognized after scanning wx and k : yz.

It is necessary to formalize the above intuitive ideas in order that the specific properties of grammar classes may be picked up. It is instructive to consider this formalization for LL(k) grammars.

LEMMA 2.1. Let G = (N,T,P,S) be an LL(k) grammar, $k \geq 0$, $w \in T^*$, β, $\gamma \in V^*$ and $n \geq 0$. If $S \overset{n}{\underset{L}{\Rightarrow}} w\beta$, $S \overset{n}{\underset{L}{\Rightarrow}} w\gamma$ and $FIRST_k(\beta) \cap FIRST_k(\gamma) \neq \emptyset$ then $\beta = \gamma$.

PROOF. Cf. [1], Lemma 8.1. □

Here we will not pay attention to a formal definition of PLC(k) grammars (cf.[12]) but instead we immediately define LP(k) grammars. In [14] these grammars were originally called Ch(k) grammars. We give here a slight restatement of the original definition.

DEFINITION 2.1. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. G is said to be an LP(k) grammar iff for any $A \in N$; α, β, β', γ, $\gamma' \in V^*$ and $w \in T^*$, if

$$S \overset{*}{\underset{L}{\Rightarrow}} wA\gamma \underset{L}{\Rightarrow} w\alpha\beta\gamma \quad \text{and} \quad S \overset{*}{\underset{L}{\Rightarrow}} wA\gamma \underset{L}{\Rightarrow} w\alpha\beta'\gamma'$$

and $FIRST_k(\beta\gamma) \cap FIRST_k(\beta'\gamma') \neq \emptyset$ then ${}^{(1)}\beta = {}^{(1)}\beta'$.

The "strong" variant of this class (strong LP(k) grammars) is defined analogously to the LL(k) case by demanding that ${}^{(1)}\beta = {}^{(1)}\beta'$ for any two productions of the form $A \to \alpha\beta$ and $A \to \alpha\beta'$ such that

$$FIRST_k(\beta FOLLOW_k(A)) \cap FIRST_k(\beta' FOLLOW_k(A)) \neq \emptyset.$$

We refer the reader to [12,14] for more detailed treatments on the class of LP(k) grammars. For the purposes of comparison we only give a result related to Lemma 2.1.

LEMMA 2.2. Let $G = (N,T,P,S)$ be an LP(k) grammar, $k \geq 0$, $w \in T^*$, β, $\gamma \in V^*$ and $n \geq 0$. If $S \overset{n}{\underset{L}{\Rightarrow}} w\beta$, $S \overset{n}{\underset{L}{\Rightarrow}} w\gamma$ and $FIRST_k(\beta) \cap FIRST_k(\gamma) \neq \emptyset$ then ${}^{(1)}\beta = {}^{(1)}\gamma$.

PROOF. Cf. [12], Lemmas 12.3 and 12.4. □

A probably better known class of grammars generating LL(k) languages is represented by LC(k) grammars (cf. [1]). We consider here the characterization of this class given by Soisalon-Soininen and Ukkonen[18] in terms of rightmost derivations. Recall that a production $A \to \beta$ is said to satisfy the LR(k) condition iff the body of Definition 1.3 is satisfied for it. The underlining in the following two definitions denotes that the underlined substrings are not rewritten in the rightmost derivation.

DEFINITION 2.2. A CFG $G = (N,T,P,S)$ is said to be an LC(k) grammar if $S \overset{+}{\underset{R}{\Rightarrow}} S$ is not possible, each A-production satisfies the LR(k) condition and if for each w, w', y, $y' \in T^*$; α, α', α'', β, $\gamma \in V^*$; $X \in V$; A, $A' \in N$ and production $A \to X\beta$ in P, the conditions

(i) $S \overset{*}{\underset{R}{\Rightarrow}} \alpha Aw \underset{R}{\Rightarrow} \alpha X\beta w \overset{*}{\underset{R}{\Rightarrow}} \alpha Xyw$

(ii) $S \overset{*}{\underset{R}{\Rightarrow}} \alpha'A'w' \underset{R}{\Rightarrow} \alpha'\alpha''X\gamma w' \overset{*}{\underset{R}{\Rightarrow}} \alpha'\alpha''Xy'w'$

(iii) $\alpha'\alpha'' = \alpha$ and $k : yw = k : y'w'$,
always imply that $\alpha A = \alpha'A'$ and $\beta = \gamma$.

We have included the condition that $S \overset{*}{\underset{R}{\Rightarrow}} S$ is not possible for an LC(k) grammar. Otherwise the following ambiguous grammar with productions $S \to S \mid a$ is to be called LC(0) (cf.[5] where similar problems are treated for LR(k) definitions). Finally the following class of grammars has been shown [18] to generate LL(k) languages.

DEFINITION 2.3. A CFG $G = (N,T,P,S)$ is said to be a PLR(k) grammar if G is LR(k) and if for each $w, w', y, y' \in T^*$; $\alpha, \alpha', \alpha'', \beta, \gamma \in V^*$; $X \in V$; $A, A' \in N$ and production $A \to X\beta$ in P, the conditions

(i) $S \overset{*}{\underset{R}{\Rightarrow}} \alpha Aw \underset{R}{\Rightarrow} \alpha X\beta w \overset{*}{\underset{R}{\Rightarrow}} \alpha Xyw$

(ii) $S \overset{*}{\underset{R}{\Rightarrow}} \alpha'A'w' \underset{R}{\Rightarrow} \alpha'\alpha''X\gamma w' \overset{*}{\underset{R}{\Rightarrow}} \alpha'\alpha''Xy'w'$

(iii) $\alpha'\alpha'' = \alpha$ and $k : yw = k : y'w'$
always imply that $\alpha A = \alpha'A'$.

In Figure 2 we present the relationships between the classes of grammars which have been mentioned in this section. All arrows denote proper inclusions.

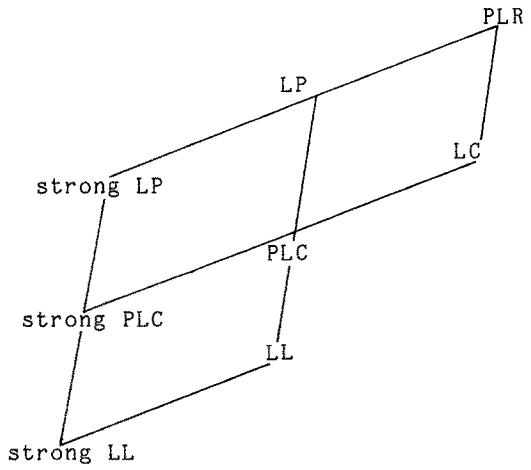

Figure 2. First inclusion diagram.

This paper is not meant to bring a discussion on the transformations converting grammars into LL(k) ones. However, for the sake of completeness, Hammer's "k-transformable" grammars [6] should be mentioned to provide such a transformation.

3. STRICT DETERMINISTIC GRAMMARS WITH LOOKAHEAD

Harrison and Havel[8] mentioned the possibility of generalizing their results by a suitable incorporation of lookahead. One of the approaches leading to the goal has appeared in Friede[4]. For notational purposes we prefer to call these grammars strong SD(k) instead of the original denotation having sounded as partitioned LL(k) grammars.

DEFINITION 3.1. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. G is said to be a strong SD(k) grammar iff there exists a partition π of V such that T forms a block of π and for all A, $A' \in N$, α, β, $\beta' \in V^*$, if $A \to \alpha\beta$, $A' \to \alpha\beta'$ are in P, $A \equiv A' \pmod{\pi}$ and

$$FIRST_k(\beta FOLLOW_k(A)) \cap FIRST_k(\beta' FOLLOW_k(A')) \neq \emptyset$$

then either

(i) both β, $\beta' \neq \Lambda$ and ${}^{(1)}\beta \equiv {}^{(1)}\beta' \pmod{\pi}$, or

(ii) $\beta = \beta' = \Lambda$ and $A = A'$.

To justify our terminology we refer the reader to compare the above definition with the one describing strong LP(k) grammars (Section 2).

THEOREM 3.1. Let $G = (N,T,P,S)$ be a strong LP(k) grammar, $k \geq 0$. Then G is a strong SD(k) grammar.

PROOF. It follows immediately from the definitions that the partition $\pi = \{\{A\} \mid A \in N\} \cup \{T\}$ satisfies the desired properties. □

This inclusion is proper since LP(k) grammars generate merely LL(k) languages whereas strong SD(k) ones were shown to generate all deterministic context-free languages (cf.[3]). A more intriguing generalization of strict deterministic grammars has been given by Pittl[15]. This latter generalization was obtained as a characterization of an existing class of grammars, namely, the LLP(k) grammars (cf. Lomet[10]).

DEFINITION 3.2. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. We define

$$M_k(G) = \{(A,u) \mid A \in N \text{ and } u \in FOLLOW_k(A)\}.$$

Let π be a weak partition of $M_k(G)$. Such a weak partition is called admissible iff for any (A,u), $(A',u') \in M_k(G)$, α, β, $\beta' \in V^*$, if $A \to \alpha\beta$, $A' \to \alpha\beta'$ are in P, $(A,u) \equiv (A',u') \pmod{\pi}$ and

$\text{FIRST}_k(\beta u) \cap \text{FIRST}_k(\beta' u') \neq \emptyset$ then either

(i) both β, β' are in TV^*, or

(ii) $\beta = C\gamma$, $\beta' = C'\gamma'$ for some $C, C' \in N$, $\gamma, \gamma' \in V^*$ and $(C,z) \equiv (C,z') \pmod{\pi}$ for all $z \in \text{FIRST}_k(\gamma u)$ and $z' \in \text{FIRST}_k(\gamma' u')$, or

(iii) $\beta = \beta' = \Lambda$ and $A = A'$.

In [15] it is shown that LLP(k) grammars are exactly those possessing an admissible weak partition. For this reason in [12] they were renamed as weak SD(k) grammars.

DEFINITION 3.3. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. G is called a weak SD(k) grammar iff there exists an admissible weak partition of $M_k(G)$.

Again, we wish to relate the new concept to previously defined ones. An admissible weak partition with disjoint blocks will be called an admissible partition.

LEMMA 3.1. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. The grammar G is strong SD(k) iff there exists an admissible partition π of $M_k(G)$ such that for any block B of π and $(A,u) \in M_k(G)$, $(A,u) \in B$ implies $(A,v) \in B$ for all $v \in \text{FOLLOW}_k(A)$.

PROOF. Let π be an admissible partition of $M_k(G)$ which satisfies the above condition. Define $\pi' = \{\{A \mid (A,u) \in B\} \mid B \in \pi\} \cup \{T\}$. Then π' is a partition of V possessing the desired properties. On the other hand let π' be a partition of V mentioned in Definition 3.1. Then the partition $\pi = \{\{(A,u) \mid A \in B \text{ and } u \in \text{FOLLOW}_k(A)\} \mid B \in \pi' - \{T\}\}$ yields clearly the result. □

THEOREM 3.2. Let $G = (N,T,P,S)$ be a strong SD(k) grammar, $k \geq 0$. Then G is weak SD(k).

PROOF. An immediate consequence of Lemma 3.1. □

It can be shown that the above inclusion is proper. Weak SD(k) grammars can be characterized in an interesting way by means of leftmost derivations.

THEOREM 3.3. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. Then G is a weak SD(k) grammar iff for any $n \geq 0$, $A, A' \in N$, $\alpha, \beta, \beta', \gamma, \gamma' \in V^*$ and $w \in T^*$, if

$$S \overset{n}{\underset{L}{\Rightarrow}} wA\gamma \underset{L}{\Rightarrow} w\alpha\beta\gamma$$

$$S \overset{n}{\underset{L}{\Rightarrow}} wA'\gamma' \underset{L}{\Rightarrow} w\alpha\beta'\gamma'$$

$$FIRST_k(\beta\gamma) \cap FIRST_k(\beta'\gamma') \neq \emptyset$$

then either (i) both β, β' are in TV^*, or
(ii) both β, β' are in NV^*, or
(iii) $\beta = \beta' = \Lambda$ and $A = A'$.

PROOF. Cf. [15], Theorem 3.2.(c). □

This characterization allows us to compare the classes of LP(k) and weak SD(k) grammars.

THEOREM 3.4. Let $G = (N,T,P,S)$ be an LP(k) grammar, $k \geq 0$. Then G is a weak SD(k) grammar.
PROOF. Use Lemma 2.2. and Theorem 3.3. □

As mentioned in [9], a lot of erroneous results has appeared in the literature connected with the conversion of rightmost derivations to leftmost ones. These technical difficulties can be overcome using the approach presented in Pittl[15]. The crucial result of that paper we recall is that proving any weak SD(k) grammar to be LR(k). We next improve it by showing these grammars to be included in an interesting subclass of LR(k) grammars introduced by Ukkonen[19,20].

DEFINITION 3.4. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. G is called to be a weak PLR(k) grammar iff it is LR(k) and for all A, $A' \in N$, α, α', α'', β, $\beta' \in V^*$, w, $w' \in T^*$ and $X \in V$, if

$$S \underset{R}{\overset{*}{\Longrightarrow}} \alpha Aw \underset{R}{\Longrightarrow} \alpha X\beta w \quad \text{and} \quad S \underset{R}{\overset{*}{\Longrightarrow}} \alpha'A'w' \underset{R}{\Longrightarrow} \alpha'\alpha''X\beta'w' = \alpha X\beta'w'$$

and $FIRST_k(\beta w) \cap FIRST_k(\beta'w') \neq \emptyset$
then $\alpha = \alpha'$ (i.e. $\alpha'' = \Lambda$).

Clearly, any PLR(k) grammar is weak PLR(k). The inclusion is proper due to the different classes of languages generated.

THEOREM 3.5. Let $G = (N,T,P,S)$ be a weak SD(k) grammar, $k \geq 0$. Then G is weak PLR(k).
PROOF. A slight modification of the proof of Theorem 5.2.[15] which proves G to be LR(k) can be shown to yield the required argument. □

The inclusion mentioned in the theorem is proper since there are left recursive PLR(k) grammars. By [15] no such grammar can be weak SD(k). Figure 3 summarizes the results concerning relationships

between the families of grammars. An arrow means a proper inclusion.

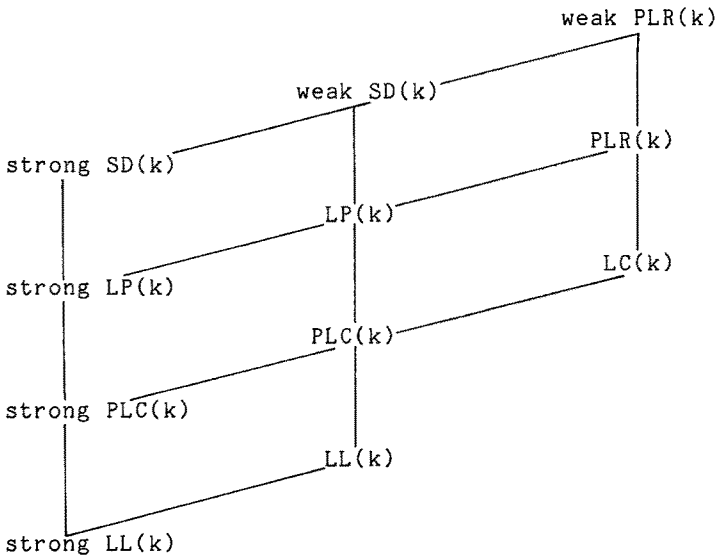

Figure 3. Second inclusion diagram.

4. TRANSFORMATIONS TO "STRONG" GRAMMARS

Most of the grammatical concepts treated in this paper originated from the attempts to facilitate parser construction for deterministic languages. From this point of view "strong" versions of grammars appeared very attractive. Indeed, the utilization of $FOLLOW_k$ sets instead of local follow sets for each sentential form yields considerable improvements in parser size. As typical examples strong LL(k) grammars and simple LR(k) grammars [2] deserve to be mentioned. This fact has lead to the investigation of transformations converting grammars into their "strong" counterparts (cf. [2,17]). We next show all these classes of grammars to possess a certain "common denominator". We present a general method providing "strong" grammars for all the types of grammar families known to the authors.

TRANSFORMATION. Input: A CFG $G = (N,T,P,S)$, $k \geq 0$. Output: A CFG $\tau(G) = (N',T,P',S')$. Method: Let $Y \subseteq M_k(G)$, $\alpha \in V^*$. Then we define

$$SUCC(Y,\alpha) = \{(B,v) \mid (A,u) \in Y,\ A \to \alpha B\beta \in P,\ v \in FIRST_k(\beta u) \text{ for some } B \in N \text{ and } \beta \in V^*\}$$

Let $l_m(G) = \max\{|\alpha| \mid A \to \alpha \text{ is in } P\}$. Next a set π_c of subsets of $M_k(G)$ is to be created in three phases.

Step 1. Initially let $\pi_c = \emptyset$. Then place the set $\{(S,A)\}$ into π_c as an unmarked element.

Step 2. If a set $Y \in \pi_c$ is unmarked then for all $\alpha \in V^*$ such that $|\alpha| \leq l_m(G)$ compute the set $SUCC(Y,\alpha)$. If this set is nonempty then place it into π_c unmarked. Then mark Y.

Step 3. Repeat Step 2. until all sets in π_c are marked.

Clearly this algorithm is guaranteed to halt since $M_k(G)$ is a finite set. Now the grammar $\tau(G) = (N',T,P',S')$ is constructed as follows. Define $S' = (\{(S,A)\},S)$ and

$$N' = \{(Y,A) \mid Y \in \pi_c \text{ and } (A,u) \in Y \text{ for some } u \in T^{*k}\}$$

The set P' contains only the productions described below. For any production $A \to X_1 \ldots X_n$ in P, where $A \in N$, $n \geq 0$, $X_i \in V$, $1 \leq i \leq n$, P' involves all the productions $B \to Z_1 \ldots Z_n$ such that $B = (Y,A)$, $Y \in \pi_c$, $(A,u) \in Y$ for some $u \in T^{*k}$ and $Z_i = X_i$ if $X_i \in T$, $Z_i = (SUCC(Y,X_1 \ldots X_{i-1}),X_i)$ if $X_i \in N$, $1 \leq i \leq n$.

This transformation represents a generalization of a similar one used in Pittl[15], Theorem 4.2. To facilitate the investigation of its properties we introduce a homomorphism $\phi : V'^* \to V^*$ by defining $\phi(a) = a$ for all $a \in T$ and $\phi((Y,A)) = A$ for all $(Y,A) \in N'$. Induction arguments on the length of the derivations prove the following two assertions.

LEMMA 4.1. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$, $Z \in V'$, $\gamma \in V'^*$ and $Z \overset{*}{\Longrightarrow}_{\tau(G)} \gamma$. Then $\phi(Z) \overset{*}{\Longrightarrow}_G \phi(\gamma)$.

LEMMA 4.2. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$, $X \in V$, $\alpha \in V^*$ and $X \overset{*}{\Longrightarrow}_G \alpha$. Then there are $X' \in V'$ and $\alpha \in V'^*$ such that $\phi(X') = X$, $\phi(\alpha') = \alpha$ and $X' \overset{*}{\Longrightarrow}_{\tau(G)} \alpha'$.

We conclude that $L(G) = L(\tau(G))$. It is easy to see that the pairs of derivations corresponding to each other are structurally equivalent with respect to the homomorphism ϕ. One can easily recuperate any parse of a word w in L(G) from a parse of w according to $\tau(G)$. Similar results appear in Moura[11]. A comparison of his results with ours has not yet been done. The next lemmas are almost direct consequences of the definitions.

LEMMA 4.3. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$, $(Y,A) \in N'$. Then $FOLLOW_k^{\tau(G)}((Y,A)) = \{u \in T^{*k} \mid (A,u) \in Y\}$.

LEMMA 4.4. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$, $n \geq 0$, $(Y,A) \in N'$, $\alpha, \beta \in V'^*$ and $(B,v) \in Y$. If $S' \overset{R}{\Longrightarrow}_{\tau(G)} \alpha(Y,A)\beta$ then there is $\gamma \in V'^*$

such that $S' \overset{n}{=\!\!>}_{\tau(G)} \alpha(Y,B)\gamma$ and $v \in \text{FIRST}_k^{\tau(G)}(\gamma)$.

It remains to verify that τ produces the desired output.

THEOREM 4.1. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. If G is an LL(k) (PLC(k), LP(k)) grammar then $\tau(G)$ is a strong LL(k) (strong PLC(k), strong LP(k)) grammar respectively.

THEOREM 4.2. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. If G is a weak SD(k) grammar then $\tau(G)$ is a strong SD(k) grammar.
PROOF (hint). The partition π of V' ensuring $\tau(G)$ to be strong SD(k) is constructed as follows. Let $Y \in \pi_c$. Define

$$B_Y = \{(Y,A) \mid (A,u) \in Y \text{ for some } u \in T^{*k}\}$$

and

$$\pi = \{B_Y \mid Y \in \pi_c\} \cup \{T\}. \square$$

THEOREM 4.3. Let $G = (N,T,P,S)$ be a CFG, $k \geq 0$. If G is an LR(k) grammar then $\tau(G)$ is a simple LR(k) grammar.

Due to its generality, our transformation is far from being optimal for many classes of grammars.

ACKNOWLEDGEMENTS. Part of these results first appeared in an internal report of McMaster University (Report No. 80-CS-25) in 1980. Independently, Friede[4] has obtained similar results.

REFERENCES

1. A.V. Aho and J.D. Ullman. The Theory of Parsing, Translation, and Compiling. Vols. 1 and 2, Prentice Hall, N.J., 1972 and 1973.

2. F.L. DeRemer. Simple LR(k) grammars. Comm. ACM 14 (1971), 453-460.

3. D. Friede. Partitioned LL(k) grammars. In: Automata, Languages and Programming. H.A. Maurer (ed.), Lect. Notes in Comp. Sci. 71, Springer, Berlin, 1979, 245-255.

4. D. Friede. Partitioned context-free grammars. TUM-I8115, December 1981, Institut fuer Informatik, Technische Universitaet Muenchen.

5. M.M. Geller and M.A. Harrison. On LR(k) grammars and languages. Theoret. Comput. Sci. 4 (1977), 245-276.

6. M. Hammer. A new grammatical transformation into deterministic top-down form. MAC TR-119, Mass. Inst. of Technology, 1974.

7. M.A. Harrison. Introduction to Formal Language Theory. Addison-Wesley, Reading, Mass., 1978.

8. M.A. Harrison and I.M. Havel. Strict deterministic grammars. J. Comput. System Sci. 7 (1973), 237-277.

9. M.A. Harrison and I.M. Havel. On the parsing of deterministic languages. J. Assoc. Comput. Mach. 21 (1974), 525-548.

10. D.B. Lomet. The construction of efficient deterministic language processors. Ph.D. Thesis, Univ. of Pennsylvania, 1969.

11. A. Moura. Syntactic equivalence of grammar classes. Ph.D. Thesis, Univ. of California at Berkeley, September 1980.

12. A. Nijholt. Context-Free Grammars: Covers, Normal Forms, and Parsing. Lect. Notes in Comp. Sci. 93, Springer, Berlin, 1980.

13. A. Nijholt. Parsing strategies: A concise survey. In: Mathematical Foundations of Computer Science. J. Gruska and M. Chytil (eds.), Lect. Notes in Comp. Sci. 118, Springer, Berlin, 1981.

14. A. Nijholt and E. Soisalon-Soininen. Ch(k) grammars - A characterization of LL(k) languages. In: Mathematical Foundations of Computer Science. J. Becvar (ed.), Lect. Notes in Comp. Sci. 74, Springer, Berlin, 1979, 390-397.

15. J. Pittl. On LLP(k) grammars and languages. Theoret. Comput. Sci. 16 (1981), 149-175.

16. J. Pittl. On LLP(k) parsers. J. Comput. System Sci. 24 (1982), 36-68.

17. D.J. Rosenkrantz and R.E. Stearns. Properties of deterministic top-down grammars. Information and Control 17 (1970), 226-256.

18. E. Soisalon-Soininen and E. Ukkonen. A method for transforming grammars into LL(k) form. Acta Informatica 12 (1979), 339-369.

19. E. Ukkonen. Transformations to produce certain covering grammars. In: Mathematical Foundations of Computer Science. J. Winkowski (ed.), Lect. Notes in Comput. Sci. 64, Springer, Berlin, 1978, 516-525.

20. E. Ukkonen. A modification of the LR(k) method for constructing compact bottom-up parsers. In: Automata, Languages and Programming. H.A. Maurer (ed.), Lect. Notes in Comput. Sci. 71, Springer, Berlin, 1979, 646-658.