ON THE PARSING AND COVERING OF SIMPLE CHAIN GRAMMARS

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

ABSTRACT.

A method is presented for obtaining a simple deterministic pushdown transducer which acts as a parser for simple chain grammars. It is shown that a simple deterministic grammar can be constructed which covers the simple chain grammar. To obtain both the simple deterministic pushdown transducer and the cover result, a new type of parse is introduced which differs from the left and right parses which are common for the usual one pass no back-tracking parsing algorithms. For the simple chain grammars this parse, the so-called left part parse, follows from a simple left part property which is satisfied by the grammatical trees of simple chain grammars.

1. INTRODUCTION

In this paper we show how some negative cover results for a subclass of the context-free grammars can lead to an optimal parsing method for the same subclass of grammars. The class of context-free grammars which we consider is the class of *simple chain grammars*. This class, introduced in [6] is a subclass of the LR(0) grammars. It has been shown that simple chain grammars can be parsed with a very simple *bottom-up parsing method*. Moreover, each simple chain grammar can be transformed to an equivalent *simple* LL(1) (or *simple deterministic* [3]) grammar. Such a transformation leads to some negative cover results. It can be shown that there is no such transformation

which makes it possible to obtain, by a string-homomorphism, the *right parses* of the simple chain grammar from the *left parses* of the simple LL(1) grammar. That is, the simple LL(1) grammar does not *left-to-right-cover* the simple chain grammar. Neither do we have that the left parses of the simple LL(1) grammar can be mapped on the left parses of the simple chain grammar, that is, the simple LL(1) grammar does not *left cover* the simple chain grammar.

How to consider these negative results? One can argue that the simple LL(1) grammars are a too restrictive class of grammars to expect nice cover results. However, the motivation to transform grammars is exactly to obtain such restrictive classes of grammars which can be parsed in a more simple way than the original grammars. Moreover, the transformation which can be given is very simple and straightforward. One can also conclude that a definition of cover by means of a homomorphism is too restrictive. However, as easily can be verified, if we use in the definition of cover instead of a homomorphism, a (deterministic) finite transducer mapping then still we do not obtain positive cover results. Besides, introduction of more complicated mappings than a homomorphism is a rather rude approach of such a simple transformation. An other approach, which we will not follow here, is to let the "homomorphic" transformations be functors which go from a (syntax) category associated with one grammar to the category of the other grammar [2]. Although this way of looking at "structure" preserving transformations has been investigated in several papers (see the references in [2]), this approach has not yet lead to very useful results. A concept as cover gives a relation between (mostly) simple descriptions of parse trees (for example left or right parses) and is therefore simple to work with. Further investigations of the categorical approach may however lead to more satisfactory results than are now available.

In this paper our point of view on this problem is the following. In the theory of parsing we are used to describing the structure of grammatical trees by means of the productions of the context-free grammar with respect to which the parsing is done. To be more specific, we associate left parses with top-down parsing and right parses with bottom-up parsing.

An exception of this rule are the *left corner parses* [7] which become useful when

we consider left corner grammars. Here we introduce another type of parse, which we associate with the simple chain grammars, the *left part parse*. We believe, and we try to illustrate this in the following sections, that this choice of parse is the most natural one, among others since, as for example the left parses for LL(k) grammars, productions are given as output of the parsing process as soon as they are determined. Moreover, this parse reflects the somewhat hybrid character (partly top-down, partly bottom-up) of the simple chain grammars.

We conclude this section with some preliminaries. In the following section we consider the parsing of simple chain grammars with respect to the left part parses. In the third section the cover problem is considered with respect to these parses.

Preliminaries.

We assume that the reader is familiar with [1]. For notational reasons we review some concepts. A *context-free grammar* (CFG for short) is denoted by the 4-tuple $G = (N,T,P,S)$, where N consists of the nonterminals (denoted by the Roman capitals A, B, C,D.E.S), T consists of the terminals (denoted by the Roman smalls a,b,c,d,e), $V = N \cup T$ (elements of V will be denoted by X,Y,Z; elements of V^* by $\alpha,\beta,\gamma,\phi,\psi$), P is the set of productions, $P \subseteq N \times V^+$ (notation $A \to \alpha$ for $(A,\alpha) \in P$) and S is the startsymbol. Leftmost and rightmost derivations are denoted as usual. We assume that the CFG's in this paper are reduced. Notice that P does not have ε-rules, i.e. productions of the form $A \to \varepsilon$, where ε is the empty string. A *left parse* of the sentence $w \in L(G)$ is the sequence of productions used in a leftmost derivation of w. A *right parse* is the reverse of the sequence of productions used in a rightmost derivation. A *simple syntax-directed translation schema* (simple SDTS for short) is a 5-tuple $Q = (N,T,\Delta,P,S)$, where N,T and S are as in the case of CFG's, Δ is a finite output alphabet and P is a finite set of rules of the form $A \to \alpha,\beta$, where $\alpha \in (N \cup T)^+$, $\beta \in (N \cup \Delta)^*$ and the nonterminals in α are the same as the nonterminals in β and they appear in the same order.

Derivations of, and translations defined by such schemes are as usual [1]. Let $\alpha \in V^+$, then $FIRST(\alpha) = \{a \in T \mid \alpha \overset{*}{\Rightarrow} a\gamma \text{ for some } \gamma \in V^*\}$. A CFG is said to be a *simple* LL(1) or a simple deterministic grammar [3] if $P \subseteq N \times TV^*$ and for each pair

$A \to a\varphi$ and $A \to b\psi$ in P we have that $a \neq b$ or $a\varphi = b\psi$. Simple deterministic grammars generate simple deterministic languages. Simple deterministic languages can be accepted by simple deterministic pushdown automata (acceptors). Here we define immediately the notion of a *simple deterministic pushdown transducer* (simple DPDT).

DEFINITION 1.1. A simple DPDT is a 5-tuple $R = (T,\Delta,\Gamma,\delta,S)$, where T is the input alphabet, Δ is the output alphabet, Γ is the alphabet of pushdown list symbols, δ is a mapping from $T \times \Gamma$ to $\Gamma^* \times \Delta^*$, and $S \in \Gamma$ is the initial pushdown list symbol. A *configuration* of R is a triple (w,α,y) in $T^* \times \Gamma^* \times \Delta^*$, where w will stand for the unused portion of the input, α represents the contents of the pushdown list and y is the output string emitted sofar. If $\delta(a,Z) = (\alpha,z)$ then we write $(ax,\gamma Z,y) \vdash (x,\gamma\alpha,yz)$† for all $x \in T^*$, $y \in \Delta^*$ and $\gamma \in \Gamma^*$. The transitive and reflexive transitive completion of $\vdash$ is defined as usual. The *translation* defined by a simple DPDT R, denoted by $\tau(R)$, is $\{(x,y) \mid (x,S,\varepsilon) \overset{*}{\vdash} (\varepsilon,\varepsilon,y)\}$.
A slight adaption of the definition of a simple chain grammar, as presented in [6], leads to the following definition.

DEFINITION 1.2. A CFG $G = (N,T,P,S)$ is said to be a simple chain grammar if

(i) $FIRST(X) \cap FIRST(Y) = \emptyset$ for all productions $A \to \alpha X\varphi$ and $A \to \alpha Y\psi$ in P with $X \neq Y$, and

(ii) $A \to \alpha$ and $A \to \alpha\beta$ in P implies $\beta = \varepsilon$.

The following theorem, which we give without proof, gives an indication how to introduce look-ahead for simple chain grammars. This can be done (not here) in such a way that the LL(k) grammars are properly included.

THEOREM 1.1. CFG $G = (N,T,P,S)$ is a simple chain grammar iff

(i) for all $w,w',w'' \in T^*$; $X,Y \in V$ and $\varphi,\psi \in V^*$, if

$$S \overset{n}{\underset{l}{\Longrightarrow}} wX\varphi \overset{*}{\underset{l}{\Longrightarrow}} ww', \text{ and}$$

$$S \overset{n}{\underset{l}{\Longrightarrow}} wY\psi \overset{*}{\underset{l}{\Longrightarrow}} ww'',$$

† Notice that the top of the pushdown list is assumed to be on the right.

and $^{(1)}w' = {}^{(1)}w''$, then $X \neq Y$. †

(ii) $A \to \alpha$ and $A \to \alpha\beta$ in P implies $\beta = \varepsilon$.

Whenever we use the words "parse of a sentence w" then we refer to a description of a grammatical tree for w by means of the productions which are used in the derivation of w.

To discuss cover results for simple chain grammars we need the following definition.

DEFINITION 1.3. A CFG G' x-to-y covers a CFG G if there exists a homomorphism $h : P' \to P^*$ such that

(i) if π' is an x-parse of w with respect to G', then $h(\pi')$ is an y-parse of w with respect to G, and

(ii) for each π such that π is an y-parse of w with respect to G there exists an x-parse π' for w with respect to G' such that $h(\pi') = \pi$.

In this definition x and y can be replaced by any type of parse, for example "left", "right", "left-corner" etc. Left-to-left and right-to-right covers will be referred to as *left covers* and *right covers*, respectively. If G' x-to-y covers G then we use the notation G'[x/y]G. We use l to abbreviate "left" and $\overline{r}$ for "right".

An example of a simple chain grammar is the CFG G with only productions $S \to aEc$, $S \to aEd$, $E \to aEb$ and $E \to ab$. It can be shown that there does not exist a simple LL(1) grammar G' such that G'[l/l]G. Another example is the simple LL(1) grammar G (and hence it is a simple chain grammar) with only productions $S \to aB$, $B \to aB$, $B \to b$, and $B \to c$, for which it can be shown that there does not exist a simple LL(1) grammar G' such that G'[l/$\overline{r}$]G. The way these results can be obtained is to try to construct a simple DPDT for these grammars which acts as a right parser and a left parser, respectively. Since this turns out to be impossible the negative cover results follow.

† For any $\alpha \in V^+$, $^{(1)}\alpha$ denotes the first element of α.

2. ON THE PARSING OF SIMPLE CHAIN GRAMMARS

As mentioned in the preceeding section there exist simple chain grammars which can not be parsed with a simple DPDT yielding a left parse or a right parse. As we show here, it is, however, possible to construct directly from the simple chain grammar a simple DPDT which acts as a parser for the grammar. In this case the parses are however not left or right parses but, as we will call them, *left part parses*. First we recall the definition of the set of *chains* of an element in the alphabet V and the notion of *chain-independency*.

DEFINITION 2.1. Let $G = (N,T,P,S)$ be a CFG. Let $X_0 \in V = N \cup T$, then $CH(X_0)$, the set of chains of X_0 is defined by

$$CH(X_0) = \{X_0X_1\ldots X_n \in N^*T \mid X_0 \underset{1}{\Rightarrow} X_1\psi_1 \underset{1}{\Rightarrow} \ldots \underset{1}{\Rightarrow} X_n\psi_n,\ \psi_i \in V^*,\ 1 \leq i \leq n\}.$$

$X_0 \in V$ is said to be chain-independent if for each pair $\pi_1 = X_0X_1\ldots X_n$ and $\pi_2 = X_0X'_1\ldots X'_m$ in $CH(X_0)$ such that $\pi_1 \neq \pi_2$, we have that $X_n \neq X'_m$.

Easily can be verified that simple chain grammars are chain-independent, that is, each element in V is chain-independent.

Informally the left part parse is now introduced with the aid of the following Figure 1. Here two grammatical trees are displayed for the simple chain grammar G with only productions 1. S → aBC, 2. S → aBD, 3. B → aB, 4. B → d, 5. B → e, 6. C → c and 7. D → d.

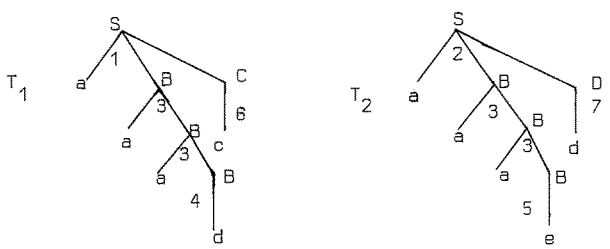

Figure 1. Two grammatical trees for simple chain grammar G.

Consider tree T_1. In Figure 2 it is displayed how tree T_1 can be built up by partial subtrees by considering the next terminal symbol, reading from left to right.

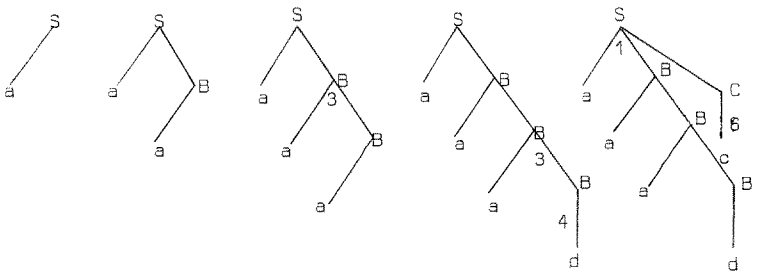

Figure 2. Partial subtrees of tree T_1.

After reading the third a the first production 3 is complete in the partial subtree, after the d the second production 3 and production 4 is complete and after reading the c the productions 1 and 6 are complete. Such a sequence, in this case 33416, will be called a left part parse. The left part parse for tree T_2 is 33527.
That, for instance, in addition after reading the third a the first production 3 is *uniquely* determined is caused by the properties of a simple chain grammar. In fact it follows immediately from the left part property of the grammatical trees of a simple chain grammar [5]. Here we do not consider this property in detail.
The left property is illustrated with the aid of Figure 3.

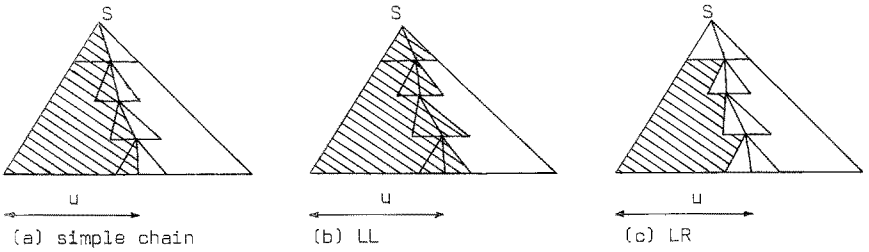

Figure 3. Structure of grammatical trees.

Informally, the left part property says that for each $A \in N$ and for each prefix u of $w = uv \in L(A)$ u uniquely determines the "left part" of the grammatical tree which corresponds to the derivation of w from A.
The left part of the tree in Figure 3(a) is the shaded part, i.e. the part determined by the prefix u and the path from the root of the tree to the last symbol of u.

It can be proved [5] that the trees of simple chain grammars satisfy this property. For LL-grammars the prefix u determines, eventually by look-ahead, all the (complete) productions which are necessary to derive u from S (see Figure 3(b)). For LR-grammars only the part of the tree determined by all possible reductions (eventually by look-ahead) from u (see Figure 3(c)) is uniquely determined.
Formally the left part parses are introduced in the following way.

DEFINITION 2.2. Let $G = (N,T,P,S)$ be a CFG. From G we obtain a simple SDTS in the following way. For each production $i.A \to \alpha X$ in P (where $\alpha \in V^*$ and $X \in V$) let $A \to \alpha X$, $\alpha' i X'$ be a rule of the simple SDTS, where $\alpha' X'$ is equal to αX with the terminal symbols deleted. The translation of $w \in L(G)$ by this simple SDTS is said to be a left part parse of w.

EXAMPLE. Let G be the simple chain grammar with only productions 0. $S \to aA$, 1. $A \to Sa$ and 2. $S \to c$. Then we have simple SDTS Q with rules $S \to aA,oA$; $A \to Sa,S1$; and $S \to c,2$; and the translation defined by Q is the set $\tau(Q) = \{(a^n c a^n, o^n 2 1^n) \mid n \geq 0\}$.

Now we are sufficiently prepared for the main results of this paper. We show that the simple syntax directed translation of Definition 2.2 on a simple chain grammar can be implemented on a simple DPDT. Therefore we give first the construction of a simple DPDT from a simple chain grammar.

CONSTRUCTION 2.1.

Input. A simple chain grammar $G = (N,T,P,S)$.

Output. A simple DPDT which acts as a left part parser for G.

Method. Let $R = (T,\Delta,\Gamma,\delta,\overline{S})$ be the simple DPDT which is constructed, where Δ consists of the numbers of productions (from 1 to $|P|$), $\Gamma = \{\overline{S}\} \cup \{\overline{A\alpha} \mid A \to \alpha\beta$ in P, $\alpha \neq \varepsilon$ and $\beta \neq \varepsilon\}$, and δ will be specified below. To do this we need again some preliminaries.

Let $i.C \to \alpha X_0 \varphi$ be in P and let $X_0 X_1 \ldots X_n \in CH(X_0)$, $n \geq 0$. Now consider the sequence $\pi = \overline{C\alpha k_0 X_0}\ \overline{X_0 k_1 X_1} \ldots \overline{X_{n-1} k_n X_n}$, where

(a) $k_0 = i$ if $\varphi = \varepsilon$, and $k_0 = \varepsilon$ otherwise, and

(b) for $1 \leq i \leq n$, $k_i = j$ if $j.X_{i-1} \to X_i$ is in P, and $k_i = \varepsilon$ otherwise.

The sequence which is obtained from π by deleting all elements from π which are not in Γ (hence all elements which represent complete productions) is denoted by $\hat{\pi}$, thus $\hat{\pi} \in \Gamma^*$.

Now the transition function δ is defined as follows:

(i) for each $SX_1 \ldots X_n \in CH(S)$ let

$$\delta(X_n, \overline{S}) = (\overline{\overline{Sk_1X_1}\ \overline{X_1k_2X_2} \ldots \overline{X_{n-1}k_nX_n}}\hat{}, k_1 \ldots k_n).$$

(ii) for each $A \to \alpha X_0 \varphi$ in P, $\alpha \neq \varepsilon$ and $X_0X_1 \ldots X_n \in CH(X_0)$, let

$$\delta(X_n, \overline{A\alpha}) = (\overline{\overline{A\alpha k_0X_0}\ \overline{X_0k_1X_1} \ldots \overline{X_{n-1}k_nX_n}}\hat{}, k_0 \ldots k_n).$$

This concludes the construction. □

First we give an example of this construction, then we prove that the simple DPDT which is obtained in this way indeed acts as a "left part parser" for a simple chain grammar.

EXAMPLE. Consider again the simple chain grammar with only productions 1. $S \to aBC$, 2. $S \to aBD$, 3. $B \to aB$, 4. $B \to d$, 5. $B \to e$, 6. $C \to c$ and 7. $D \to d$. We display the transition function δ for the tuples in $T \times \Gamma$ for which δ is defined.

$\delta(a, \overline{S}) = (\overline{Sa}, \varepsilon)$ $\qquad$ $\delta(d, \overline{SaB}) = (\varepsilon, 27)$

$\delta(a, \overline{Sa}) = (\overline{SaB}\ \overline{Ba}, \varepsilon)$ $\qquad$ $\delta(a, \overline{Ba}) = (\overline{Ba}, 3)$

$\delta(d, \overline{Sa}) = (\overline{SaB}, 4)$ $\qquad$ $\delta(d, \overline{Ba}) = (\varepsilon, 34)$

$\delta(e, \overline{Sa}) = (\overline{SaB}, 5)$ $\qquad$ $\delta(e, \overline{Ba}) = (\varepsilon, 35)$

$\delta(c, \overline{SaB}) = (\varepsilon, 16)$ $\qquad$ End of example. □

Note. The left part parse which is defined in Definition 2.2 may be called a *top-down* left part parse. If we replace in Construction 2.1 the output $k_1 \ldots k_n$ and $k_0 \ldots k_n$ by $k_n \ldots k_1$ and $k_n \ldots k_0$, respectively, then the parse which is obtained may be called a *bottom-up* left part parse.

CLAIM 2.1. R is a simple DPDT.

Proof. Obviously R has no ε-rules. That R transduces with empty pushdown list follows from the following lemma. It remains to show that δ is well-defined, that is, that R is deterministic. This can easily be done by verifying that the assumption that R is nondeterministic leads to contradictions with the properties of a simple chain grammar. □

LEMMA 2.1. Let G be a simple chain grammar, let Q be the simple SDTS for G as in Definition 2.2 and let R be the DPDT obtained for G by Construction 2.1. Then

$$(S,S) \overset{*}{\Rightarrow} (w,\pi) \text{ in } Q \text{ iff } (w,\overline{S},\varepsilon) \overset{*}{\vdash} (\varepsilon,\varepsilon,\pi) \text{ in } R.$$

Proof. First we show the *only if part* of this lemma. Therefore we have the following claim.

CLAIM 2.2. Let $A \to \alpha X \varphi$ be in P and let $X \underset{l}{\Rightarrow} X_1\psi_1 \underset{l}{\Rightarrow} \dots \underset{l}{\Rightarrow} X_{n-1}\psi_{n-1} \underset{l}{\Rightarrow} Y\psi$ be a possible derivation, where $XX_1\dots X_{n-1}Y \in N^*V$, and $\psi_i,\psi \in V^*$, $1 \le i \le n-1$. Then $(Y,Y') \overset{m}{\Rightarrow} (y,\pi)$, for some $m \ge 0$, $y \in T^+$, $\pi \in \Delta^*$ and with $Y' = Y$ if $Y \in N$ and $Y' = \varepsilon$ otherwise, implies

$(y,\overline{A\alpha},\varepsilon) \overset{*}{\vdash} (\varepsilon,(\overline{A\alpha k X_0}\dots\overline{X_{n-1}k'X_n})\hat{}, k\dots k'\pi)$, where $X_0 = X$ and $X_n = Y$.

Proof. Suppose $m = 0$, then $y = Y \in T$ and $\pi = \varepsilon$. Then, since $A \to \alpha X \varphi$ and $X \overset{*}{\underset{l}{\Rightarrow}} Y\psi$, we have that $\delta(y,\overline{A\alpha}) = ((\overline{A\alpha k X}\dots\overline{X_{n-1}k'y})\hat{}, k\dots k')$, where $X\dots X_{n-1}y \in CH(X)$, and it follows that

$$(y,\overline{A\alpha},\varepsilon) \vdash ((\overline{A\alpha k X_0}\dots\overline{X_{n-1}k'X_n})\hat{}, k\dots k'\pi).$$

Now assume $m > 0$ and assume the claim holds for all $m' < m$ (induction hypothesis). Then, if $j.Y \to Y_1Y_2\dots Y_q$ is the first production which is used, we have the following derivation. $(Y,Y) \Rightarrow (Y_1Y_2\dots Y_q,(Y_1\dots Y_{q-1})'jY_q') \overset{*}{\Rightarrow} (y_1y_1\dots y_q,\pi_1\pi_2\dots\pi_{q-1}j\pi_q)$, where $(Y_i,Y_i') \overset{m_i}{\Longrightarrow} (y_i,\pi_i)$, $1 \le i \le q$, $y_i \in T^*$ and $m_i < m$. Then we can use the induction hypothesis from which we obtain

$$(y_1,\overline{A\alpha},\varepsilon) \overset{*}{\vdash} (\varepsilon,(\overline{A\alpha k X}\dots\overline{Yk_1Y_1})\hat{}, k\dots k_1\pi_1),$$

and

$$(y_i,\overline{YY_1\dots Y_{i-1}},\varepsilon) \overset{*}{\vdash} (\varepsilon,\overline{YY_1\dots Y_{i-1}k_iY_i}\hat{}, k_i\pi_i)$$

for $1 < i \le q$.

Then we have

$$(y_1y_2\ldots y_q,\overline{A\alpha},\varepsilon) \overset{*}{\vdash} (y_2\ldots y_q,(\overline{A\alpha kX}\ldots\overline{Yk_1Y_1})\hat{}, k\ldots k_1\pi_1)$$

$$\overset{*}{\vdash} (\varepsilon,(\overline{A\alpha kX}\ldots\overline{YY_1\ldots Y_{q-1}k_qY_q})\hat{}, k\ldots k_1\pi_1k_2\pi_2\ldots k_q\pi_q) =$$

$$= (\varepsilon,(\overline{A\alpha kX_0}\ldots\overline{X_{n-1}k'X_n})\hat{}, k\ldots k'\pi_1\pi_2\ldots\pi_{q-1}j\pi_q),$$

where $X_0 = X$ and $X_n = Y$, which was to be proved. □

Now let $A \to \alpha\Omega\rho$ and suppose $(C,C) \overset{*}{\Rightarrow} (x,\pi)$, then from Claim 2.2 it follows that $(x,\overline{A\alpha},\varepsilon) \overset{*}{\vdash} (\varepsilon,\overline{A\alpha kC}\hat{},k\pi)$.

Notice that the claim also holds for $\alpha = \varepsilon$ and $A = S$. Now let $1.\ S \to Z_1Z_2\ldots Z_n$ be the first production which is used in a derivation $w = z_1z_2\ldots z_n \in T^*$, where $(Z_i,Z_i') \overset{*}{\Rightarrow} (z_i,\pi_i)$, $1 \le i \le n$. Then it follows that

$$(S,S) \Rightarrow (Z_1Z_2\ldots Z_n,(Z_1\ldots Z_{n-1})'1Z_n') \overset{*}{\Rightarrow} (w,\pi_1\ldots\pi_{n-1}1\pi_n)$$

implies

$$(w,\overline{S},\varepsilon) \overset{*}{\vdash} (\varepsilon,\varepsilon,\pi_1\ldots\pi_{n-1}1\pi_n),$$

which was to be proved.

Now we come to the *if part* of the lemma.

CLAIM 2.3. $(w,\overline{A\alpha X},\varepsilon) \overset{m}{\vdash} (\varepsilon,\varepsilon,\pi)$ implies $(A,A) \overset{*}{\Rightarrow} (\alpha Xw,\alpha'X'\pi)$, where $\alpha'X'$ is equal to αX with terminal symbols deleted.

Proof. The proof is by induction on m. Let $w = ax$, $a \in T$ and $x \in T^*$. If $m = 1$ then $x = \varepsilon$, hence $w = a$. In this case

$$\delta(a,\overline{A\alpha X}) = ((\overline{A\alpha Xk_0X_0}\ \overline{X_0k_1X_1}\ldots\overline{X_{n-1}k_nX_n})\hat{}, k_0k_1\ldots k_n),$$

where

$$X_0X_1\ldots X_n \in CH(X_0),\ X_n = a,\ (\overline{A\alpha Xk_0X_0}\ \overline{X_0k_1X_1}\ldots\overline{X_{n-1}k_nX_n})\hat{} = \varepsilon$$

and $\pi' = k_1\ldots k_n$ is the left part parse associated with $X_0 \overset{*}{\Rightarrow} X_n$. Thus,

$$(a,\overline{A\alpha X},\varepsilon) \vdash (\varepsilon,\varepsilon,k_0\pi') \text{ implies } (A,A) \Rightarrow (\alpha XX_0,\alpha'X'k_0X_0') \overset{*}{\Rightarrow} (\alpha Xa,\alpha'X'k_0\pi').$$

Now let $m > 1$. Let the first step be done with the transition

$$\delta(a,\overline{A\alpha X}) = ((\overline{A\alpha Xk_0X_0}\ \overline{X_0k_1X_1}\ldots\overline{X_{n-1}k_nX_n})\hat{},k_0k_1\ldots k_n),$$

where $X_n = a$. Then,

$$(ax,\overline{A\alpha X},\varepsilon) \vdash (x,(\overline{A\alpha Xk_0X_0}\ \overline{X_0k_1X_1}\ldots\overline{X_{n-1}k_nX_n})\hat{},\ k_0k_1\ldots k_n) \overset{*}{\vdash} (\varepsilon,\varepsilon,\pi).$$

Obviously there exist $x_i \in T^*$, $0 \le i \le n$, such that $x = x_n x_{n-1} \ldots x_2 x_1 x_0$ and $\pi_i \in \Delta^*$, $0 \le i \le n$, such that $\pi = k_0 k_1 \ldots k_n \pi_n \pi_{n-1} \ldots \pi_2 \pi_1 \pi_0$, where $\pi_i = x_i = \varepsilon$ if $k_i \neq \varepsilon$ (notice that in case $k_0 \neq \varepsilon$ or $k_i \neq \varepsilon$, $1 \le i \le n$, then $\overline{A\alpha X k_0 X_0}\hat{} = \varepsilon$ or $\overline{X_{i-1} k_i X_i}\hat{} = \varepsilon$, respectively) and such that, for those k_i's not equal ε,

$$(x_i, \overline{X_{i-1} k_i X_i}, \varepsilon) \overset{m_i}{\vdash} (\varepsilon, \varepsilon, \pi_i)$$

and

$$(x_0, \overline{A\alpha X k_0 X_0}, \varepsilon) \overset{m_0}{\vdash} (\varepsilon, \varepsilon, \pi_0).$$

Since $m_0, m_i < m$ we obtain

$$(X_{i-1}, X_{i-1}) \overset{*}{\Rightarrow} (X_i x_i, k_i X_i' \pi_i) \qquad 1 \le i \le n \qquad (*)$$

and

$$(A, A) \overset{*}{\Rightarrow} (\alpha X X_0 x_0, \alpha' X' k_0 X_0' \pi_0) \qquad (**)$$

Notice that the cases $k_i = \varepsilon$ and $k_i \neq \varepsilon$ can be taken together. From (*) and (**) it follows immediately that

$$(A, A) \overset{*}{\Rightarrow} (\alpha X a x, \alpha' X' \pi).$$ □

Now let $(w, \overline{S}, \varepsilon) \overset{m}{\vdash} (\varepsilon, \varepsilon, \pi)$. The first step, with $w = ax$, yields

$$(ax, \overline{S}, \varepsilon) \vdash (x, \overline{(Sk_0X_0}\ \overline{X_0 k_1 X_1} \ldots \overline{X_{n-1} k_n X_n})\hat{}, k_0 k_1 \ldots k_n) \overset{*}{\vdash} (\varepsilon, \varepsilon, \pi),$$

where, again $X_n = a$ and the other notations are as usual. From Claim 2.3, with an analogous partition of x and π as in its proof, we obtain

$(S, S) \overset{*}{\Rightarrow} (X_0 x_0, k_0 X_0' \pi_0)$, and for $1 \le i \le n$,

$(X_{i-1}, X_{i-1}) \overset{*}{\Rightarrow} (X_i x_i, k_i X_i' \pi_i)$, hence

$$(S, S) \overset{*}{\Rightarrow} (X_n x_n \ldots x_1, k_0 k_1 \ldots k_n \pi_n \ldots \pi_1 \pi_0) = (w, \pi),$$

which had to be proved. □

Now the following corollary is immediate.

COROLLARY 2.1. Each simple chain grammar has a simple DPDT which acts as a left part parser.

3. ON THE COVERING OF SIMPLE CHAIN GRAMMARS

As previously mentioned there is no transformation from the class of simple chain grammars to the class of simple deterministic grammars such that we can obtain a left cover or a left-to- right cover. With the results of Section 2 we can now show, in a way analogous to the argument in [4], that each simple chain grammar G has an eq equivalent simple deterministic (simple LL(1)) grammar G' such that left parses with respect to G' can be mapped on left part parses with respect to G, that is, G'[l/lp]G. Since this result follows immediately (as a more restricted case) from some general results in [4] we can confine ourselves to a sketch of the proof.

THEOREM 3.1. Let G be a simple chain grammar. Then G can be transformed to a simple deterministic grammar G', such that G'[l/lp]G.

Proof. (sketch). For a simple chain grammar G = (N,T,P,S) we can construct, with Construction 2.1, a simple DPDT R which acts a left part parser. It is obvious how to construct from R a simple deterministic grammar G' = (N',T,P',S'). Then, in G'

$S' \underset{l}{\overset{\pi'}{\Longrightarrow}} w$ iff

$$(w,\bar{S},\varepsilon) \overset{*}{\vdash} (\varepsilon,\varepsilon,h(\pi')),$$

where h is the cover-homomorphism defined as follows. The production of G' obtained from a rule

$$\delta(a,A) = (X_1X_2\ldots X_k,y)$$

is mapped on y.

Then it is straightforward to show that the conditions in Definition 1.3 for a *left-to-left part* cover are satisfied. □

4. CONCLUSIONS

The class of simple chain grammars can be considered as a generalization of the class of strict deterministic grammars of degree 1 [8]. The strict deterministic grammars of degree 1 form a proper subclass of the class of simple chain grammars. Therefore, results obtained for simple chain grammars also hold for strict deterministic grammars of degree 1. Although each simple chain grammar can be transformed to a simple LL(1) grammar, this transformation is not a left or a left-to-right cover. The main motivation for writing this paper was to provide an answer to this "cover-problem".

To obtain both a positive cover result and an optimal parsing method we introduced a new type of parse. This leads to the following observation. To obtain a more simple parsing method (simple in the sense of the type of device which can be used, here a simple DPDT) we have to introduce a less simple type of parse. Maybe this is not too surprising. Anyway, the simple chain grammars and the results presented here provide a clear illustration of such an observation.

Left part parsing, resulting in either a top-down or a bottom-up left part parse, can be done for any (ε-free) non-left recursive context-free grammar. A straightforward generalization of Construction 2.1 will make this clear. The parser is then a (nondeterministic) pushdown transducer without ε-rules and with one state only.

ACKNOWLEDGEMENTS

I thank Marja Verburg for her careful typing of this paper.

REFERENCES

1. A.V. AHO and J.D. ULLMAN, *The Theory of Parsing, Translation and Compiling*, Vol. 1 and 2, Prentice-Hall, Englewood Cliffs, N.J., 1972 and 1973.

2. D.B. BENSON, *Some preservation properties of normal form grammars*, Siam J. of Comput. 6 (1977), pp. 381-402.

3. A.J. KORENJAK and J.E. HOPCROFT, *Simple deterministic languages*, in "7th Ann. Sympos. on Sw. and Aut. Theory, IEEE 1966", pp. 36-46.

4. A. NIJHOLT, *On the covering of parsable grammars*, J. Comput. System Sci.15 (1977), pp. 99-110.

5. A. NIJHOLT, *A left part theorem for grammatical trees*, IR-22, Dept. of Mathematics, Vrije Universiteit Amsterdam, august 1977.

6. A. NIJHOLT, *Simple chain grammars*, Proceedings of the 4th Coll. on Automata, Languages and Programming 1977 (eds. A. Salomaa and M. Steinby) pp. 352-364, Lecture Notes in Computer Science 52, Springer Verlag, Berlin.

7. D.J. ROSENKRANTZ and P.M. LEWIS, *Deterministic left-corner parsing*, in "11th Ann. Sympos. on Sw. and Aut. Theory, IEEE 1970", pp. 139-152.

8. M.A. HARRISON and I.M. HAVEL, *Real-time strict deterministic languages*, Siam J. of Comput. 4 (1972), pp. 333-349.