Acta Informatica 12, 339–369 (1979)

A Method for Transforming Grammars into *LL*(*k*) Form

Eljas Soisalon-Soininen and Esko Ukkonen

Department of Computer Science, University of Helsinki,
Tukholmankatu 2, SF-00250 Helsinki 25, Finland

Summary. A new method for transforming grammars into equivalent $LL(k)$ grammars is studied. The applicability of the transformation is characterized by defining a subclass of $LR(k)$ grammars, called predictive $LR(k)$ grammars, with the property that a grammar is predictive $LR(k)$ if and only if the corresponding transformed grammar is $LL(k)$. Furthermore, it is shown that deterministic bottom-up parsing of a predictive $LR(k)$ grammar can be done by the $LL(k)$ parser of the transformed grammar. This parsing method is possible since the transformed grammar always 'left-to-right covers' the original grammar. The class of predictive $LR(k)$ grammars strictly includes the class of $LC(k)$ grammars (the grammars that can be parsed deterministically in the left-corner manner). Thus our transformation is more powerful than the one previously available, which transforms $LC(k)$ grammars into $LL(k)$ form.

Subject Classifications: $LL(k)$ grammars, parsing of context-free grammars, grammatical covering.

1. Introduction

Left-corner parsing refers to a class of parsing procedures in which the productions are recognized in a particular order different from both bottom-up and top-down. Each production is recognized after its leftmost descendant but before its other descendants. Left-corner parsing has its origin in the parsing procedure of Irons [15], and procedures in this class have since appeared frequently in the compiler literature [1, 3, 4–6, 10, 14, 25].

Deterministic left-corner parsing is formalized by Rosenkrantz and Lewis [25] (see also [1]). They define a class of grammars, called $LC(k)$ grammars, and demonstrate that a certain canonical pushdown automaton, which does left-corner parsing, is deterministic precisely for $LC(k)$ grammars. In addition, they suggest a grammatical rewriting procedure that yields an $LL(k)$ grammar [1, 22] if and only if the original grammar is $LC(k)$. This rewriting technique has earlier been described in different formats by several authors: Rosenkrantz [24] and

0001-5903/79/0012/0339/$06.20

Wood [30] used it to convert grammars into a certain normal form, whereas Foster [7] eliminated left recursion; the format of this transformation used by Rosenkrantz and Lewis was suggested by Griffiths and Petrick [11]. However, while it is precise, the definition of $LC(k)$ grammars is rather complicated. In fact, no formal proofs of the results mentioned above have appeared in the literature.

In this paper, which is based on a preliminary report [28], we study a class of grammars that is defined by restricting the $LR(k)$ grammars by Knuth [16], and that is a slight generalization of $LC(k)$ grammars. Inspired by this new class of grammars, called predictive $LR(k)$ grammars and abbreviated as $PLR(k)$ grammars, we also suggest a new definition for the class of $LC(k)$ grammars, and using this definition, we relate $LC(k)$ grammars to $LR(k)$ grammars.

Our primary motivation in defining $PLR(k)$ grammars is to provide an exact characterization of a new grammatical transformation for achieving $LL(k)$ grammars, which are grammars that can be parsed in a particularly easy manner. That is, we present a grammatical rewriting technique and show that when this technique is applied to a grammar, the resulting grammar is $LL(k)$ if and only if the original grammar is $PLR(k)$. The transformation presented in this paper resembles a method introduced by Hammer [12, 13] for achieving $LL(k)$ grammars since both transformations have the property that when parsing a string top-down with respect to the transformed grammar, the bottom-up parse of the string with respect to the original grammar can be obtained. However, our transformation technique has some important abvantages over the method of Hammer, although our technique is applicable to a somewhat smaller number of grammars. (The fact that Hammer's method always yields an $LL(k)$ grammar when the original grammar is $PLR(k)$ can be demonstrated by arguments similar to those used in showing the same result for $LC(k)$ grammars [12].) We have been able to give a precise grammatical condition under which the transformation yields the desired result. Also in contrast to the method of Hammer, which is based on an intricate parser construction, our transformation is a single process that is applied directly to the grammar. Moreover, even if the resulting grammar may sometimes be uneconomically large, considerable improvement is possible using simple techniques.

Other grammatical transformations that may yield $LL(k)$ grammars are given in [7, 8, 18, 19]. See also [9, 21, 29].

The present paper is organized as follows. Section 2 contains some terminology and a brief review of the theory of deterministic parsing. In Section 3 we give the definitions of $LC(k)$ and $PLR(k)$ grammars, and relate these classes of grammars both to each other and to $LR(k)$ grammars. Section 4 is devoted to analysing $PLR(k)$ grammars. We introduce our transformation technique and show that the transformed grammar generates the same language as the original grammar and that it is $LL(k)$ if and only if the original grammar is $PLR(k)$. We also define a deterministic bottom-up parsing algorithm for $PLR(k)$ grammars in terms of top-down parsing of the transformed grammar. Such a parsing method is possible since the transformed grammar left-to-right covers the original grammar in the sense of [23].

2. Background

In this section we shall briefly review the basic notions of context-free grammars and deterministic parsing. We mainly follow the treatment of [1].

If R is a binary relation on a set W then R^0 denotes the identity relation on W and R^n the composite relation $R^{n-1} \circ R$ for $n>0$. The transitive closure $\bigcup_{n=1}^{\infty} R^n$ of the relation R is denoted by R^+, and the reflexive transitive closure $R^+ \cup R^0$ is denoted by R^*. If V is a set the elements of which are called characters or symbols, then V^0 denotes the set $\{\varepsilon\}$, where ε is the so-called empty string, V^n the concatenation $V^{n-1}V$ for $n>0$, and V^* the set $\bigcup_{n=0}^{\infty} V^n$ consisting of all strings over V. The length of a string w in V^* is denoted by $|w|$. If k is a non-negative integer we stipulate that $k{:}w$ is equal to w if $|w| \leqq k$, and to the string formed by the first k symbols of w if $|w|>k$.

A quadruple $G=(N, T, P, S)$ is a *(context-free) grammar*, if N and T are finite disjoint sets, P is a finite subset of $N \times (N \cup T)^*$, and S is in N. Elements of the set N are called *nonterminals* and denoted by capital Latin letters A, B, C, S. Elements of the set T are called *terminals* and denoted by small Latin letters a, b, c. The symbols X, Y and Z denote elements of the set $N \cup T$ (and ε if explicitly indicated). The elements (A, α) of P are called *productions* and denoted by $A \rightarrow \alpha$, where the nonterminal A is called the *left-hand side* of the production, the string α the *right-hand side*, and $1{:}\alpha$ the *left corner* of the production. In particular, a production the right-hand side of which is ε is called an *ε-production*. The symbol S is called the *start symbol*.

Terminal strings, i.e. elements of T^*, are denoted by small Latin letters from the end of the alphabet $t, u, \ldots, z$, whereas small Greek letters $\alpha, \beta, \ldots, \omega$ denote elements of $(N \cup T)^*$. We use $\Rightarrow$ to represent the relation $\{(\alpha A \beta, \alpha\omega\beta) \mid \alpha, \beta \in (N \cup T)^* \text{ and } (A, \omega) \in P\}$ on $(N \cup T)^*$. In particular, the subrelations $\{(wA\alpha, w\omega\alpha) \mid w \in T^*, \ \alpha \in (N \cup T)^* \text{ and } (A, \omega) \in P\}$ and $\{(\alpha A w, \alpha\omega w) \mid \alpha \in (N \cup T)^*, \ w \in T^* \text{ and } (A, \omega) \in P\}$ are denoted by $\underset{L}{\Rightarrow}$ and $\underset{R}{\Rightarrow}$, respectively. If $\alpha \Rightarrow^* \beta$ we say that in the grammar G the string α *derives* the string β or that β is obtained by a *derivation* from α. In particular, if $\alpha \underset{L}{\Rightarrow}^* \beta$ or $\alpha \underset{R}{\Rightarrow}^* \beta$, we say that β is obtained by a *leftmost* or, respectively, by a *rightmost* derivation from α. We write $\alpha\underline{\gamma} \underset{L}{\Rightarrow}^* \beta\underline{\gamma}$ or $\underline{\gamma}\alpha \underset{R}{\Rightarrow}^* \underline{\gamma}\beta$, if we wish to express that $\alpha\gamma \underset{L}{\Rightarrow}^* \beta\gamma$ and $\alpha \underset{L}{\Rightarrow}^* \beta$ or, respectively, $\gamma\alpha \underset{R}{\Rightarrow}^* \gamma\beta$ and $\alpha \underset{R}{\Rightarrow}^* \beta$.

Strings obtained by a derivation, a leftmost derivation or a rightmost derivation from the start symbol S are called *sentential forms*, *left sentential forms* or, respectively, *right sentential forms* of the grammar G. The set of terminal strings derived by the start symbol S is called the *language generated* by the grammar G and is denoted by $L(G)$.

A sequence Π of productions is called a *left* or a *top-down parse* of γ from β in the grammar G, if either $\Pi=\varepsilon$ and $\gamma=\beta$, or if there exist a string $wA\alpha$, a production $A \to \omega$ of G and a sequence Π' of productions such that $wA\alpha \underset{L}{\Rightarrow} w\omega\alpha = \gamma$, $\Pi = \Pi'(A, \omega)$ and Π' is a left parse of $wA\alpha$ from β. Correspondingly, a sequence Π of productions is called a *right* or a *bottom-up parse* of γ to β in the grammar G, if either $\Pi = \varepsilon$ and $\gamma = \beta$, or if there exist a string $\alpha A w$, a production $A \to \omega$ of G and a sequence Π' of productions such that $\alpha A w \underset{R}{\Rightarrow} \alpha\omega w = \gamma$, $\Pi = (A, \omega)\Pi'$ and Π' is a right parse of $\alpha A w$ to β. A *left* or a *top-down parser* of G is an algorithm that determines for each string in $L(G)$ a left parse of it from the start symbol S and reports error for strings not in the language. Correspondingly, a *right* or a *bottom-up parser* of G is an algorithm that determines for each string in $L(G)$ a right parse of it to the start symbol and reports error for other strings.

A grammar $G = (N, T, P, S)$ is said to be *unambiguous*, if each string in the language $L(G)$ has a unique left parse from S. Otherwise, G is *ambiguous*. It is well-known that a string in $L(G)$ has a unique left parse from S if and only if it has a unique right parse to S. Thus, the grammar G is unambiguous if and only if each string in $L(G)$ has a unique right parse to the start symbol.

In a grammar $G = (N, T, P, S)$, a symbol X in $N \cup T$ is said to be *useless* if X does not appear in any sentential form of G that derives strings in T^*. As is well-known, there is an algorithm for removing useless symbols from context-free grammars (see e.g. [1]), and we assume throughout this paper that grammars do not contain useless symbols.

The concepts of deterministic top-down and bottom-up parsing have led to the formulation of grammatical conditions under which these particular parsing techniques are applicable (e.g. [1]). For the so-called $LL(k)$ grammars, first defined by Lewis and Stearns [22], deterministic top-down parsing is possible. We define the class of $LL(k)$ grammars following e.g. [1, 17, 20].

Let k be a non-negative integer. A grammar $G = (N, T, P, S)$ is said to be $LL(k)$ if in the grammar G the conditions

$$S \underset{L}{\Rightarrow}{}^* xA\alpha \underset{L}{\Rightarrow} x\beta\alpha \underset{L}{\Rightarrow}{}^* xy_1,$$
$$S \underset{L}{\Rightarrow}{}^* xA\alpha \underset{L}{\Rightarrow} x\gamma\alpha \underset{L}{\Rightarrow}{}^* xy_2,$$
$$k : y_1 = k : y_2$$

always imply that $\beta = \gamma$.

A language $L \subseteq T^*$ is said to be $LL(k)$ if there is an $LL(k)$ grammar G such that $L = L(G)$.

In the sequel we will make use of some well-known properties of $LL(k)$ grammars; we shall now review them here. First of all, each $LL(k)$ grammar is unambiguous, since it follows immediately from the definition that each string in a language generated by an $LL(k)$ grammar has a unique left parse from the start symbol. Another important property is that an $LL(k)$ grammar cannot have left-recursive nonterminals, i.e. nonterminals A that satisfy the condition $A \Rightarrow{}^+ A\alpha$ for some α. The proof of this property can be found e.g. in [2, 20].

Finally, in the following lemma we delineate a useful consequence of the $LL(k)$ condition; this lemma is merely a restatement of Lemma 8.1 in [2].

Lemma 2.1. *Let $G=(N, T, P, S)$ be an $LL(k)$ grammar. If there exist in G derivations*

$$S \underset{L}{\Rightarrow}{}^* xA\alpha \underset{L}{\Rightarrow} x\beta\alpha \underset{L}{\Rightarrow}{}^* xy_1 \tag{2.1}$$

and

$$S \underset{L}{\Rightarrow}{}^* x'B\delta \underset{L}{\Rightarrow} x'\gamma\delta \underset{L}{\Rightarrow}{}^* x'x''y_2 = xy_2 \tag{2.2}$$

such that $k:y_1 = k:y_2$, then either (1) $xA\alpha = x'B\delta$ *or* (2) *derivation* (2.1) *is of the form*

$$S \underset{L}{\Rightarrow}{}^* x'B\delta \underset{L}{\Rightarrow} x'\gamma\delta \underset{L}{\Rightarrow}{}^* xA\alpha \underset{L}{\Rightarrow}{}^* xy_1$$

or $x'=x$ and derivation (2.2) *is of the form*

$$S \underset{L}{\Rightarrow}{}^* xA\alpha \underset{L}{\Rightarrow} x\beta\alpha \underset{L}{\Rightarrow}{}^* xB\delta \underset{L}{\Rightarrow}{}^* xy_2. \quad \square$$

For $LR(k)$ grammars, first defined by Knuth [16], deterministic bottom-up parsing is possible. In defining this class of grammars we use the method provided in [1].

If $G=(N, T, P, S)$ is a grammar, then the *augmented grammar* for G is the grammar $G'=(N\cup\{S'\}, T\cup\{\perp\}, P\cup\{S'\to\perp S\}, S')$, where S' is not in N and $\perp$ is not in T. (The symbol $\perp$ not used in [1] is introduced only for technical reasons and is not necessary for defining $LR(k)$ grammars.)

Let k be a non-negative integer. A grammar $G=(N, T, P, S)$ is said to be $LR(k)$ if the augmented grammar G' for G the three conditions

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z_1 \underset{R}{\Rightarrow} \alpha\beta z_1, \tag{2.3}$$

$$S' \underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \alpha'\beta' z_2 = \alpha\beta z_2', \tag{2.4}$$

$$k:z_1 = k:z_2' \tag{2.5}$$

always imply that $\alpha A = \alpha' B$ and $z_2 = z_2'$. A production $A\to\beta$ of G satisfies the $LR(k)$ condition if for fixed A and β conditions (2.3), (2.4) and (2.5) always imply that $\alpha A = \alpha' B$ and $z_2 = z_2'$.

A language $L\subseteq T^*$ is said to be $LR(k)$ if there is an $LR(k)$ grammar G such that $L=L(G)$.

We shall now consider some consequences of the $LR(k)$ definition. First of all, each $LR(k)$ grammar is unambiguous, since it follows immediately from the definition that each string in a language generated by an $LR(k)$ grammar has a unique right parse to the start symbol. In the second place, we wish to prove a useful variation of the $LR(k)$ condition, and to do this we first state the following lemma, which is a modification of Lemma 5.2 in [1].

Lemma 2.2. *A grammar G is $LR(k)$, if in the augmented grammar G' for G the conditions*

$$S' \underset{R}{\Rightarrow}{}^{*} \alpha A z_1 \underset{R}{\Rightarrow} \alpha\beta z_1,$$
$$S' \underset{R}{\Rightarrow}{}^{*} \alpha' B z_2 \underset{R}{\Rightarrow} \alpha'\beta' z_2 = \alpha\beta y_2 z_2,$$
$$k:z_1 = k:y_2 z_2$$

always imply that $\alpha A = \alpha' B$ *and* $z_2 = y_2 z_2$. □

The following theorem will be used e.g. in demonstrating that the class of $LC(k)$ grammars is included in the class of $LR(k)$ grammars.

Theorem 2.1. *Let* $G = (N, T, P, S)$ *be a grammar and let* k *be a non-negative integer. The grammar* G *satisfies the* $LR(k)$ *condition if in the augmented grammar* G' *the conditions*

$$S' \underset{R}{\Rightarrow}{}^{*} \alpha A z_1 \underset{R}{\Rightarrow} \alpha\beta z_1, \tag{2.6}$$

$$S' \underset{R}{\Rightarrow}{}^{*} \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\beta'}\gamma z_2 \underset{R}{\Rightarrow}{}^{*} \underline{\alpha'\beta'} y_2 z_2, \tag{2.7}$$

$$\alpha\beta = \alpha'\beta', \quad |\beta'| > 0 \quad and \quad k:z_1 = k:y_2 z_2 \tag{2.8}$$

always imply that $\alpha A = \alpha' B$ *and* $\beta = \beta'\gamma$. *Conversely, if a production* $A \to \beta$, $\beta \neq \varepsilon$, *of* G *satisfies the* $LR(k)$ *condition, then conditions* (2.6), (2.7) *and* (2.8) *imply that* $\alpha A = \alpha' B$ *and* $\beta = \beta'\gamma$.

Proof. We first assume that the grammar G is not $LR(k)$. Then there exist in G' two derivations

$$S' \underset{R}{\Rightarrow}{}^{*} \alpha A z_1 \underset{R}{\Rightarrow} \alpha\beta z_1 \tag{2.9}$$

and

$$S' \underset{R}{\Rightarrow}{}^{*} \alpha' B z_2 \underset{R}{\Rightarrow} \alpha'\gamma z_2 = \alpha\beta z_2' \tag{2.10}$$

such that $\alpha A \neq \alpha' B$ or $z_2 \neq z_2'$, although $k:z_1 = k:z_2'$. In addition, we may assume by Lemma 2.2 that $z_2' = y_2 z_2$ for some y_2. Now there are two cases to consider depending on whether $|\alpha\beta| \leqq |\alpha'|$ or not.

Case 1. Let us first consider the case in which $|\alpha\beta| > |\alpha'|$. Then derivation (2.10) can be written as a derivation

$$S' \underset{R}{\Rightarrow}{}^{*} \alpha' B z_2 \underset{R}{\Rightarrow} \alpha'\beta' y_2 z_2 = \alpha\beta y_2 z_2.$$

But this form of derivation (2.10) together with derivation (2.9) leads to an immediate violation of the condition of the theorem.

Case 2. If $|\alpha\beta| \leqq |\alpha'|$, derivation (2.10) can be written in the form

$$S' \underset{R}{\Rightarrow}{}^{*} \alpha'' C w_2 \underset{R}{\Rightarrow} \underline{\alpha''\beta''}\varphi w_2 \underset{R}{\Rightarrow}{}^{*} \underline{\alpha''\beta''} u B z_2$$
$$\underset{R}{\Rightarrow} \alpha''\beta'' u v z_2 = \alpha''\beta'' y_2 z_2,$$

where $\alpha''\beta'' = \alpha\beta$ and $|\beta''| > 0$. (Note that the string β'' can be chosen such that $|\beta''| > 0$, since $S' \to \bot S$ is the only production of G' the left-hand side of which is

S'.) But since $k:z_1=k:y_2z_2$, we should be able to decide that $\alpha A=\alpha''C$ and $\beta=\beta''\varphi$ if the condition of the theorem holds. However, since φ cannot be ε while deriving a string beginning with uB, we conclude that either $\alpha A\neq\alpha''C$ or $\beta\neq\beta''\varphi$, as desired.

To prove the second part of the theorem, we assume that there exist in G' derivations

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z_1 \underset{R}{\Rightarrow} \alpha\beta z_1 \tag{2.11}$$

and

$$S' \underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\beta'}\gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\beta'} y_2 z_2=\alpha\beta y_2 z_2, \tag{2.12}$$

where $\beta\neq\varepsilon$ and $\beta'\neq\varepsilon$, such that $\alpha A\neq\alpha' B$ or $\beta\neq\beta'\gamma$, although $k:z_1=k:y_2z_2$. Now if γ is the terminal string y_2, then we immediately have a contradiction of the $LR(k)$ condition. (Note that in the case $\alpha A=\alpha' B$ the inequality $\beta\neq\beta'\gamma$ implies that $z_2\neq y_2z_2$.) Otherwise derivation (2.12) can be written as a derivation

$$\begin{aligned} S' &\underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\beta'}\gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\beta'} u C v z_2 \\ &\underset{R}{\Rightarrow} \alpha'\beta' u x v z_2=\alpha'\beta' y_2 z_2, \end{aligned}$$

which, together with derivation (2.11), also violates the $LR(k)$ condition, since $\beta\neq\varepsilon$ implies that $\alpha'\beta' uC$ $(=\alpha\beta uC)\neq\alpha A$. The proof of the theorem is thus complete. □

3. Definition of $LC(k)$ and $PLR(k)$ Grammars

To intuitively characterize the classes of grammars to be defined, we first illustrate the deterministic bottom-up and left-corner parsing algorithms, i.e. the parsing algorithms that apply to $LR(k)$ and $LC(k)$ grammars, respectively. Consider the derivation tree shown in Fig. 1.

In the bottom-up parsing algorithm for $LR(k)$ grammars the productions for the nonterminals in the tree are recognized in the order B, A, C, E, D, F, S. Each production in the tree is recognized after its descendants but before its ancestors and its right siblings and their descendants. In the left-corner parsing algorithm for $LC(k)$ grammars the productions are recognized in the order A, B, S, C, D,

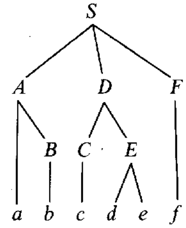

Fig. 1. Derivation tree

E, F. Each production is recognized after its left corner but before any of the siblings of the left corner (or their descendants). The $PLR(k)$ grammars are $LR(k)$ grammars for which the left-hand sides of the productions can be recognized in the same order as the whole productions in left-corner parsing, but the right-hand sides are recognized in the order of the bottom-up parse. For example, when the left-corner parsing algorithm has recognized the productions for A, B, S, C and D, then in the bottom-up parsing algorithm for $PLR(k)$ grammars the left-hand sides A, B, S, C and D are determined although productions have been recognized only for B, A and C.

Since the $LC(k)$ grammars are those that can be deterministically parsed in the left-corner manner, they can, roughly speaking, be characterized as follows. A grammar is $LC(k)$ if for each terminal string in the language each production in its derivation can be identified by inspecting the string from its beginning to k symbols beyond the left corner of the production. (It is additionally required in [25] that if the left corner is a terminal then the production is already identified after k symbols beyond the beginning of the production. This asymmetry is apparently introduced only for the technical reason that the transformation of grammars on which this formalization of deterministic left-corner parsing is based would otherwise not rewrite all $LC(k)$ grammars in $LL(k)$ form.) The $PLR(k)$ grammars are then those for which the left-hand side of each production can be identified by inspecting the terminal string from its beginning to k symbols beyond the left corner but the right-hand side is identified by inspecting the terminal string from its beginning to k symbols beyond the end of the production.

To illustrate these two definitions suppose that there is a production $A \to X\alpha$, where X is a terminal or a nonterminal. Suppose furthermore that

$$S \Rightarrow^* wAz, \qquad X \Rightarrow^* x, \qquad \alpha \Rightarrow^* y,$$

and consider the terminal string $wxyz$. The production $A \to X\alpha$ can be recognized with certainty after scanning

(i) wx and $k:yz$, if the grammar is $LC(k)$,

(ii) wxy and $k:z$, if the grammar is $LR(k)$ or $PLR(k)$.

However, if the grammar is $PLR(k)$ then the left-hand side A of the production $A \to X\alpha$ can be recognized with certainty after scanning wx and $k:yz$.

We shall first give a formal definition of $LC(k)$ grammars, and then extend it to define $PLR(k)$ grammars. Let k be a non-negative integer. A grammar $G = (N, T, P, S)$ is said to be $LC(k)$, if each ε-production of G satisfies the $LR(k)$ condition and if in the augmented grammar G' for each production $A \to X\beta$, $X\beta \neq \varepsilon$, the conditions

$$S' \underset{R}{\Rightarrow}^* \alpha A z_1 \underset{R}{\Rightarrow} \underline{\alpha X}\beta z_1 \underset{R}{\Rightarrow}^* \underline{\alpha X} y_1 z_1,$$

$$S' \underset{R}{\Rightarrow}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\alpha'' X}\gamma z_2 \underset{R}{\Rightarrow}^* \underline{\alpha'\alpha'' X} y_2 z_2,$$

$$\alpha'\alpha'' = \alpha \quad \text{and} \quad k:y_1 z_1 = k:y_2 z_2$$

always imply that $\alpha A = \alpha' B$ and $\beta = \gamma$.

It should be noted that our definition of $LC(k)$ grammars is not quite equivalent to that of Rosenkrantz and Lewis in [25]. An $LC(k+1)$ grammar as defined by Rosenkrantz and Lewis can be $LC(k)$ in our sense, even if it is not $LC(k)$ by their definition. But our definition could be changed to an equivalent definition by dividing it into two parts depending on whether the left corner X of the production $A \to X\beta$ is a nonterminal or a terminal. For a nonterminal X the definition remains as it is, but for a terminal X the condition $k:y_1 z_1 = k:y_2 z_2$ is replaced by $k:X y_1 z_1 = k:X y_2 z_2$.

Since for $LC(k)$ grammars the whole right-hand side of a production – in addition to the left-hand side – must be recognized after seeing the left-corner and k symbols lookahead, the implication $\beta = \gamma$ is needed in the definition. By substituting for this requirement the condition that the grammar is $LR(k)$ we obtain the definition of $PLR(k)$ grammars.

Let k be a non-negative integer. A grammar $G=(N, T, P, S)$ is said to be $PLR(k)$, if G is $LR(k)$ and in the augmented grammar G' for each production $A \to X\beta$, $X\beta \neq \varepsilon$, the conditions

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z_1 \underset{R}{\Rightarrow} \underline{\alpha X} \beta z_1 \underset{R}{\Rightarrow}{}^* \underline{\alpha X} y_1 z_1,$$

$$S' \underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\alpha'' X} \gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\alpha'' X} y_2 z_2,$$

$$\alpha'\alpha'' = \alpha \quad \text{and} \quad k:y_1 z_1 = k:y_2 z_2$$

always imply that $\alpha A = \alpha' B$.

A language $L \subseteq T^*$ is said to be $LC(k)$ (respectively $PLR(k)$) if there is an $LC(k)$ (respectively $PLR(k)$) grammar such that $L = L(G)$.

Example 3.1. Consider the grammar $G=(\{S, A\}, \{a, b, c\}, P, S)$, where P consists of the productions

$$S \to aAb$$
$$S \to aAc$$
$$A \to Aa$$
$$A \to a.$$

This grammar is not $LC(k)$ for any k. To see this, let k be a positive integer and consider the derivations

$$S' \underset{R}{\Rightarrow} \perp S \underset{R}{\Rightarrow} \perp aAb \underset{R}{\Rightarrow}{}^* \perp a a^n b$$

and

$$S' \underset{R}{\Rightarrow} \perp S \underset{R}{\Rightarrow} \perp aAc \underset{R}{\Rightarrow}{}^* \perp a a^n c,$$

where $n \geq k$. Now, although $k:a^n b = k:a^n c$, the right-hand sides aAb and aAc are not equal, which violates the $LC(k)$ condition. Intuitively, the grammar is not $LC(k)$, because for each string in the language it is necessary to inspect the last symbol of the string – which may be arbitrarily far to the right – to determine the first production in the left-corner parse.

On the other hand, it is easily seen that this grammar is $PLR(1)$. For instance, the special $PLR(1)$ condition required in addition to the $LR(1)$-ness is satisfied for the production $A \to Aa$, since the conditions

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z_1 \underset{R}{\Rightarrow} \alpha A a z_1,$$

$$S' \underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\alpha'' A}\gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\alpha'' A} y_2 z_2,$$

$$\alpha'\alpha'' = \alpha \quad \text{and} \quad 1{:}az_1 = 1{:}y_2 z_2$$

imply that B is equal to A and α' is equal to α, $\alpha = \perp a$. Note that $1{:}y_2 z_2$ can also be the terminal b or c, and then $B = S$ and $\alpha' = \perp$, which means that the grammar cannot be $PLR(0)$. □

To prove that $LC(k)$ grammars are also $PLR(k)$ grammars we first show that $LC(k)$ grammars are $LR(k)$ grammars.

Theorem 3.1. *An $LC(k)$ grammar is $LR(k)$.*

Proof. Assume that an $LC(k)$ grammar $G = (N, T, P, S)$ is not $LR(k)$. Then there must exist a production $A \to \beta$, $\beta \neq \varepsilon$, of G, which does not satisfy the $LR(k)$ condition. This means, by Theorem 2.1, that there are in the augmented grammar G' two rightmost derivations

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z_1 \underset{R}{\Rightarrow} \alpha \beta z_1 \tag{3.1}$$

and

$$S' \underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\beta'}\gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\beta'} y_2 z_2, \tag{3.2}$$

where $|\beta'| > 0$, such that $\alpha A \neq \alpha' B$ or $\beta \neq \beta'\gamma$, although $\alpha'\beta' = \alpha\beta$ and $k{:}z_1 = k{:}y_2 z_2$. Now if $|\alpha'| \leq |\alpha|$, we can conclude that the production $A \to \beta$ used in (3.1) violates the $LC(k)$ condition. That is, if β is denoted by $X\delta$, then β' may be denoted by $\alpha'' X \delta$ such that we have in G' the derivations

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z_1 \underset{R}{\Rightarrow} \underline{\alpha X} \delta z_1 \underset{R}{\Rightarrow}{}^* \underline{\alpha X} x z_1$$

and

$$S' \underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\alpha'' X} \delta \gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\alpha'' X} x y_2 z_2,$$

where $\alpha'\alpha'' = \alpha$. But since $k{:}z_1 = k{:}y_2 z_2$ implies that $k{:}xz_1 = k{:}xy_2 z_2$, and $X\delta \neq \alpha'' X \delta\gamma$ implies in the case $\alpha A = \alpha' B$ that $\delta \neq \delta\gamma$, we have because of these derivations a contradiction of the assumption that the grammar G is $LC(k)$.

On the other hand, if $|\alpha'| > |\alpha|$, we may conclude, as in the above case, that the grammar G is not $LC(k)$ because of the production $B \to \beta'\gamma$ used in derivation (3.2). The proof of the theorem is complete. □

Because $LC(k)$ grammars are $LR(k)$ grammars, it is immediate from the definitions of $LC(k)$ and $PLR(k)$ grammars that an $LC(k)$ grammar is also a $PLR(k)$ grammar. In addition, this inclusion is proper for each non-negative integer k. The grammar given in Example 3.1 is not $LC(k)$ for any k but is $PLR(1)$ and, for example, the grammar with productions

$S \to ab$

$S \to ac$

is not $LC(0)$ although it is $PLR(0)$.

In the following theorem we shall give an exact characterization of the relationship between the classes of $LC(k)$ and $PLR(k)$ grammars. We first define that a grammar is *left-factored* if it has no two productions $A \to \alpha\beta_1$ and $A \to \alpha\beta_2$ such that α is not the empty string.

Theorem 3.2. *A left-factored grammar is $PLR(k)$ if and only if it is $LC(k)$.*

Proof. As noted above all $LC(k)$ grammars are $PLR(k)$ grammars. Consider then a $PLR(k)$ grammar G that is left-factored. Since G is $PLR(k)$, any two derivations

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z_1 \underset{R}{\Rightarrow} \underline{\alpha X} \beta z_1 \underset{R}{\Rightarrow}{}^* \underline{\alpha X} y_1 z_1,$$

$X\beta \neq \varepsilon$, and

$$S' \underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\alpha'' X} \gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\alpha'' X} y_2 z_2 = \alpha X y_2 z_2$$

in G' together with the condition $k:y_1 z_1 = k:y_2 z_2$ imply that $\alpha A = \alpha' B$. Thus to demonstrate that G is $LC(k)$ it is sufficient to show that these derivations also imply that $\beta = \gamma$. But since $A = B$ and $\alpha = \alpha'$, we know by these derivations that G' has productions $A \to X\beta$ and $A \to X\gamma$. Since G was assumed to be left-factored, this means that $\beta = \gamma$. $\square$

Left-factoring, i.e. transforming grammars into left-factored form for the same language, is a standard method in attempting to obtain $LL(k)$ grammars (see e.g. [1, 7, 21]). For instance, we may repeat the following procedure until the grammar is left-factored (the procedure is applied to the grammar obtained by the preceding application): Select a group, if there are any, of productions of the form $A \to \alpha\beta_1$, $A \to \alpha\beta_2 \ldots, A \to \alpha\beta_n$, $|\alpha| > 0$, $n \geqq 2$, and replace it by the productions $A \to \alpha A'$ and $A' \to \beta_1$, $A' \to \beta_2, \ldots, A' \to \beta_n$, where A' is a new nonterminal.

It can easily be concluded that the $PLR(k)$-ness of a grammar is not affected by left-factoring and that left-factoring cannot produce a $PLR(k)$ grammar from a non-$PLR(k)$ grammar. Thus by Theorem 3.2 $PLR(k)$ grammars are precisely those that can be transformed into $LC(k)$ form by left-factoring.

Example 3.2. Left-factoring the grammar of Example 3.1, we obtain the grammar with productions

$S \to aAA'$

$A' \to b$

$A' \to c$

$A \to Aa$

$A \to a.$

This grammar is $LC(1)$, since the original grammar was $PLR(1)$. $\square$

We have

Corollary 3.1. *The classes of $LC(k)$ and $PLR(k)$ languages are equal.*

It is well-known that $LC(k)$ languages are equal to $LL(k)$ languages (this result is proved in [27]). Thus, by Corollary 3.1, the class of $PLR(k)$ languages is also equal to the class of $LL(k)$ languages. Finally, we note that since $LL(k)$ languages are a proper subclass of $LR(k)$ languages (e.g. [1]), so are $PLR(k)$ languages also. Thus, since $PLR(k)$ grammars are $LR(k)$ grammars, the class of $PLR(k)$ grammars is a proper subclass of the class of $LR(k)$ grammars.

4. Analysis of *PLR(k)* Grammars

In this section we shall develop a general test to determine whether a given grammar is $PLR(k)$, and shall also produce a deterministic bottom-up parser for $PLR(k)$ grammars. Our method is rather unconventional: we shall show that the problem of whether a grammar is $PLR(k)$ can be reduced to the problem of whether a certain transformed grammar is $LL(k)$, and also that the deterministic top-down parser for such an $LL(k)$ grammar can be used as a determinstic bottom-up parser for the original grammar.

4.1. The Transformation

We now describe a rewriting technique that will transform a $PLR(k)$ grammar, $k>0$, into an $LL(k)$ grammar for the same language. Let $G=(N, T, P, S)$ be a grammar and let $G'=(N', T', P', S')$ be the augmented grammar for G. We first define a relation, denoted R_{Lc}, on the set $N \cup T \cup \{\varepsilon\}$ as follows: $A\, R_{Lc} X$, if there is in P a production $A \to \alpha$ such that $X = 1:\alpha$. The *transformed grammar* for G is then defined to be the grammar $T(G)=(N'', T, P'', [S', \perp])$, where the set P'' has been constructed as follows (the set N'' consists of all symbols of the form $[A, \alpha]$ that appear in the productions):

For each production

$$A \to X_1 X_2 \ldots X_n$$

in $P'(n \geqq 0;\ X_1 \ldots X_n = \varepsilon$, if $n=0)$ the set P'' contains

(i) the production

$$[A, X_1 \ldots X_i] \to a[A, X_1 \ldots X_i a]$$

for each i, $1 \leqq i < n$, and for each a in T such that $X_{i+1} R_{Lc}^* a$;

(ii) the production

$$[A, X_1 \ldots X_i Y] \to [B, Y][A, X_1 \ldots X_i B]$$

for each i, $1 \leqq i < n$, and for each B in N and Y in $N \cup T \cup \{\varepsilon\}$ such that $B R_{Lc} Y$ and $X_{i+1} R_{Lc}^* B$;

(iii) the production

$[A, X_1 \dots X_n] \to \varepsilon$

We now illustrate the given transformation technique.

Example 4.1. As a simple example consider the grammar with productions

$$S \to Sa$$
$$S \to a$$

The transformed grammar for this grammar has the productions

$$\begin{array}{llll}
[S', \perp] & \to a[S', \perp a] & [S', \perp S] & \to [S, S][S', \perp S] \\
[S', \perp a] & \to [S, a][S', \perp S] & [S, S] & \to a[S, Sa] \\
[S, a] & \to \varepsilon & [S, Sa] & \to \varepsilon \\
[S', \perp S] & \to \varepsilon & &
\end{array}$$

Example 4.2. Consider the grammar with productions

$$\begin{array}{l}
S \to i \leftarrow A \\
S \to i \leftarrow B \\
A \to A * P \\
A \to P \\
B \to A = A \\
P \to (A) \\
P \to i
\end{array}$$

The transformed grammar $T(G)$ has the productions

$$\begin{array}{llll}
[S', \perp] & \to i[S', \perp i] & [A, A*P] & \to \varepsilon \\
[S', \perp i] & \to [S, i][S', \perp S] & [B, A] & \to =[B, A=] \\
[S', \perp S] & \to \varepsilon & [B, A=] & \to ([B, A=(] \\
[S, i] & \to \leftarrow [S, i\leftarrow] & [B, A=] & \to i[B, A=i] \\
[S, i\leftarrow] & \to ([S, i\leftarrow(] & [B, A=(] & \to [P, (][B, A=P] \\
[S, i\leftarrow] & \to i[S, i\leftarrow i] & [B, A=i] & \to [P, i][B, A=P] \\
[S, i\leftarrow(] & \to [P, (][S, i\leftarrow P] & [B, A=P] & \to [A, P][B, A=A] \\
[S, i\leftarrow i] & \to [P, i][S, i\leftarrow P] & [B, A=A] & \to [A, A][B, A=A] \\
[S, i\leftarrow P] & \to [A, P][S, i\leftarrow A] & [B, A=A] & \to \varepsilon \\
[S, i\leftarrow A] & \to [A, A][S, i\leftarrow A] & [P, (] & \to ([P, ((] \\
[S, i\leftarrow A] & \to [B, A][S, i\leftarrow B] & [P, (] & \to i[P, (i] \\
[S, i\leftarrow A] & \to \varepsilon & [P, i] & \to \varepsilon \\
[S, i\leftarrow B] & \to \varepsilon & [P, ((] & \to [P, (][P, (P] \\
[A, P] & \to \varepsilon & [P, (i] & \to [P, i][P, (P] \\
[A, A] & \to *[A, A*] & [P, (P] & \to [A, P][P, (A] \\
[A, A*] & \to ([A, A*(] & [P, (A] & \to [A, A][P, (A] \\
[A, A*] & \to i[A, A*i] & [P, (A] & \to)[P, (A)] \\
[A, A*(] & \to [P, (][A, A*P] & [P, (A)] & \to \varepsilon \\
[A, A*i] & \to [P, i][A, A*P] & &
\end{array}$$

This grammar can be simplified by eliminating nonterminals that appear on the left-hand side of only one production (excluding $[S', \perp]$). The simplified grammar has the productions

$$
\begin{array}{llll}
[S', \perp] & \to i\leftarrow[S, i\leftarrow] & [B, A=] & \to ([P, (][B, A=A] \\
[S, i\leftarrow] & \to ([P, (][S, i\leftarrow A] & [B, A=] & \to i[B, A=A] \\
[S, i\leftarrow] & \to i[S, i\leftarrow A] & [B, A=A] & \to *[A, A*][B, A=A] \\
[S, i\leftarrow A] & \to *[A, A*][S, i\leftarrow A] & [B, A=A] & \to \varepsilon \\
[S, i\leftarrow A] & \to =[B, A=] & [P, (] & \to ([P, (][P, (A] \\
[S, i\leftarrow A] & \to \varepsilon & [P, (] & \to i[P, (A] \\
[A, A*] & \to ([P, (] & [P, (A] & \to *[A, A*][P, (A] \\
[A, A*] & \to i & [P, (A] & \to)
\end{array}
$$

It is easily verified that both new grammars are $LL(1)$ and generate exactly the same language as the given grammar. □

In the case of a subset of Algol 60 containing 53 onterminals and 102 productions the transformed grammar has 91 nonterminals and 313 productions after the improvement described in Example 4.2.

4.2. Properties of the Transformation

In this section we shall show that the language generated by the transformed grammar equals the language generated by the original grammar, and also that for $k>0$ the transformed grammar is $LL(k)$ if and only if the original grammar is $PLR(k)$. We begin with several lemmas to delineate the correspondence between the rightmost derivations in the given grammar and the leftmost derivations in the transformed grammar. In what follows $G=(N, T, P, S)$ denotes the given grammar, G' its augmented grammar, and $T(G)$ the transformed grammar.

We first prove some properties of the leftmost derivations in the transformed grammar.

Lemma 4.1. *In the transformed grammar $T(G)$ a derivation*

$$[A, X] \underset{L}{\Rightarrow}{}^* x[B, Y_1 \dots Y_r]\tau, \qquad r \geqq 1,$$

is of the form

$$[A, X] \underset{L}{\Rightarrow}{}^* y[B, Y_1]\underline{\tau} \underset{L}{\Rightarrow}{}^* yz[B, Y_1 \dots Y_r]\underline{\tau},$$

where $yz=x$.

Proof. We prove the lemma by induction on the length of the derivation $[A, X] \underset{L}{\Rightarrow}{}^* x[B, Y_1 \dots Y_r]\tau$. First, if $[A, X] \underset{L}{\Rightarrow}{}^0 x[B, Y_1 \dots Y_r]\tau$, the lemma holds trivially.

Let n be a non-negative integer and assume that the lemma holds for derivations $[A, X] \underset{L}{\Rightarrow}{}^i x[B, Y_1 \dots Y_r]\tau$, where $i \leqq n$. Suppose that

$$[A, X] \underset{L}{\Rightarrow}{}^{n+1} x[B, Y_1 \dots Y_r]\tau,$$

and suppose further that $r>1$ (the lemma holds trivially in the case $r=1$). The symbol $[B, Y_1 \dots Y_r]$ may have been introduced into the derivation $[A, X] \underset{L}{\Rightarrow}^{n+1} x[B, Y_1 \dots Y_r]\tau$ in two ways. Either this derivation is of the form

$$[A, X] \underset{L}{\Rightarrow}^{n} x'[B, Y_1 \dots Y_{r-1}]\tau \underset{L}{\Rightarrow} x' Y_r [B, Y_1 \dots Y_{r-1} Y_r]\tau,$$

where $x' Y_r = x$, or this derivation is of the form

$$\begin{aligned}[A, X] &\underset{L}{\Rightarrow}^{i_1} x'[B, Y_1 \dots Y_{r-1} Z]\tau \\ &\underset{L}{\Rightarrow} x'[Y_r, Z]\underline{[B, Y_1 \dots Y_{r-1} Y_r]\tau} \\ &\underset{L}{\Rightarrow}^{i_2} x'x''\underline{[B, Y_1 \dots Y_r]\tau},\end{aligned}$$

where $x'x''=x$ and $i_1+i_2=n$. In both cases, using the induction hypothesis, we can immediately conclude that the lemma holds for the derivation involved. □

Lemma 4.2. *If there is in $T(G)$ a derivation*

$$[A, \alpha] \underset{L}{\Rightarrow}^{*} \tau \underset{L}{\Rightarrow} x,$$

where x is in T^* and the right-hand side of the production used in the first derivation step from $[A, \alpha]$ is of the form $a[A, \alpha a]$, $[B, \varepsilon][A, \alpha B]$ or ε, then $\tau = x[A, \alpha\beta]$ for some β such that $A \to \alpha\beta$ is a production of G'.

Proof. Let $\tau_0 \underset{L}{\Rightarrow} \tau_1 \underset{L}{\Rightarrow} \dots \underset{L}{\Rightarrow} \tau_n$, $n \geqq 1$, be a derivation in $T(G)$ such that $\tau_0 = [A, \alpha]$, $\tau_n = x$ and $\tau_1 = a[A, \alpha a]$, $[B, \varepsilon][A, \alpha B]$ or ε. If $\tau_1 = \varepsilon$ then necessarily $n=1$ and the lemma is true since in this case $[A, \alpha] \to \varepsilon$ is a production of $T(G)$ and hence $A \to \alpha$ is a production of G'. Otherwise it follows from the construction of $T(G)$ that $\tau_i = \eta_i[A, \alpha\beta_i]$ for some η_i and β_i, β_i is in $(N \cup T)^*$, $i = 1, 2, \dots, n-1$. On the other hand, the productions of the form $[B, \gamma] \to \varepsilon$, such that $B \to \gamma$ is a production of G', are the only productions of $T(G)$ the right-hand side of which is a terminal string. Thus τ_{n-1} is equal to $x[A, \alpha\beta]$ for some β such that $A \to \alpha\beta$ is a production of G'. □

The following two lemmas delineate the 'local' correspondence between the rightmost derivations in G and the leftmost derivations in $T(G)$.

Lemma 4.3. *Let $A \to X_1 \dots X_q$, $q \geqq 0$, be a production of G'. For each X_i, $1 \leqq i \leqq \max(1, q)$, (if $q=0$, then $X_1 \dots X_q = \varepsilon$ and hence $X_1 = \varepsilon$) and for each Y in $N \cup T \cup \{\varepsilon\}$ such that $X_i R^*_{Lc} Y$*

$$\begin{cases} A \underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}^{*} \underline{X_1 \dots X_{i-1}} Yx, & \\ \text{where } Y \text{ is derived from } X_i & \text{if } i>1 \\ A \underset{R}{\Rightarrow} \underline{X_1} X_2 \dots X_q \underset{R}{\Rightarrow}^{*} \underline{X_1} x, & \text{otherwise} \end{cases} \tag{4.1}$$

holds in G' if and only if in $T(G)$

$$\begin{cases} [A, X_1 \dots X_{i-1} Y] \underset{L}{\Rightarrow}^{*} x[A, X_1 \dots X_q] \underset{L}{\Rightarrow} x, & \text{if } i>1 \\ [A, X_1] \underset{L}{\Rightarrow}^{*} x[A, X_1 \dots X_q] \underset{L}{\Rightarrow} x, & \text{otherwise.} \end{cases} \tag{4.2}$$

Proof. We prove the lemma by induction on the length of derivation (4.1), as well as on the length of derivation (4.2). First, by the construction of $T(G)$, for a symbol X_i in the right-hand side of any production $A \to X_1 \dots X_q$ of G' there exists a derivation (4.1) of length 1 in G' if there exists a derivation (4.2) of length 1 in $T(G)$. Conversely, if there is a derivation (4.1) of length 1 in G', then the symbols $X_{i+1}, X_{i+2}, \dots, X_q$ are all terminals and $X_{i+1} \dots X_q = x$, and there is in $T(G)$ a derivation (4.2) of length $1 + |X_{i+1} \dots X_q|$. (That is, $[A, X_1 \dots X_i]$ produces $x[A, X_1 \dots X_q]$ by the productions $[A, X_1 \dots X_j] \to X_{j+1}[A, X_1 \dots X_j X_{j+1}]$ where $j = i, \dots, q-1$.)

Let n be a positive integer. Assume first that (4.1) implies (4.2) for each X_i in the right-hand side of any production $A \to X_1 \dots X_q$ of G', if the length of derivation (4.1) is less than or equal to n. Then let the length of derivation (4.1) for an arbitrary X_i in the right-hand side of an arbitrary production $A \to X_1 \dots X_q$ of G' be equal to $n+1$. There are two cases to consider depending on whether a production has been applied to X_i in derivation (4.1) or not.

Case 1. We first consider the case in which $i > 1$ and derivation (4.1) is of the form

$$\begin{aligned} A &\underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}{}^* \underline{X_1 \dots X_{i-1}} Bz \\ &\underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} Y\alpha z \underset{R}{\Rightarrow}{}^* \underline{X_1 \dots X_{i-1}} Yyz \\ &= X_1 \dots X_{i-1} Yx \end{aligned}$$

for some B in N, $X_i R_{Lc}^* B$, and α in $(N \cup T)^*$. Now, applying the induction hypothesis to the derivations

$$A \underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}{}^* \underline{X_1 \dots X_{i-1}} Bz \quad \text{and} \quad B \underset{R}{\Rightarrow} \underline{Y}\alpha \underset{R}{\Rightarrow}{}^* \underline{Y}y,$$

we obtain in $T(G)$ the derivations

$$[A, X_1 \dots X_{i-1} B] \underset{L}{\Rightarrow}{}^* z[A, X_1 \dots X_q] \underset{L}{\Rightarrow} z \tag{4.3}$$

and

$$[B, Y] \underset{L}{\Rightarrow}{}^* y[B, Y\alpha] \Rightarrow y. \tag{4.4}$$

On the other hand, $T(G)$ must have the production

$$[A, X_1 \dots X_{i-1} Y] \to [B, Y][A, X_1 \dots X_{i-1} B],$$

and thus, using derivations (4.3) and (4.4), we conclude that

$$[A, X_1 \dots X_{i-1} Y] \underset{L}{\Rightarrow}{}^* yz[A, X_1 \dots X_q] \underset{L}{\Rightarrow} yz = x,$$

as desired.

Case 2. Another possibility is that derivation (4.1) is of the form

$$A \underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}{}^* \underline{X_1 \dots X_{i-1}} X_i x, \tag{4.5}$$

i.e. no production has been applied to X_i. In this case, consider the symbols $X_{i+1}, \ldots, X_q$, and let X_j, $i+1 \leqq j \leqq q$, be the first nonterminal in this sequence, i.e. X_j is in N and $X_{i+1}, \ldots, X_{j-1}$ are in T. (The case $i=q$ and the case that no such X_j can be found are ruled out, since the length of derivation (4.1) is assumed to be $n+1>1$.) Now, since $X_{i+1}, \ldots, X_{j-1}$ are terminal symbols, derivation (4.5) can be written as

$$A \underset{R}{\Rightarrow} \underline{X_1 \ldots X_{j-1}} X_j \ldots X_q \underset{R}{\Rightarrow}^* \underline{X_1 \ldots X_{j-1}} Z y = X_1 \ldots X_i x,$$

where Z is a terminal or ε and $X_j R_{Lc}^+ Z$. We can apply Case 1 to this form of derivation (4.5), and we have in $T(G)$ the derivation

$$[A, X_1 \ldots X_{j-1} Z] \underset{L}{\Rightarrow}^* y[A, X_1 \ldots X_q] \underset{L}{\Rightarrow} y.$$

On the other hand, since $X_{i+1}, \ldots, X_{j-1}$ are terminals and Z is a terminal or ε, we can conclude directly by the construction of $T(G)$ that

$$[A, X_1 \ldots X_{i-1} X_i] \underset{L}{\Rightarrow}^* X_{i+1} \ldots X_{j-1} Z [A, X_1 \ldots X_{j-1} Z].$$

Hence,

$$\begin{aligned}[A, X_1 \ldots X_{i-1} X_i] &\underset{L}{\Rightarrow}^* X_{i+1} \ldots X_{j-1} Z y [A, X_1 \ldots X_q] \\ &\underset{L}{\Rightarrow} X_{i+1} \ldots X_{j-1} Z y = x,\end{aligned}$$

and the proof of Case 2 is complete.

Assume in the second place that for any symbol X_i in any production $A \to X_1 \ldots X_q$ of G' a derivation (4.2) implies a derivation (4.1), if the length of derivation (4.2) is less than or equal to n. Then let the length of derivation (4.2) for an arbitrary X_i in the right-hand side of an arbitrary production $A \to X_1 \ldots X_q$ of G' be equal to $n+1$. To show that there is a derivation (4.1) in this case also, we have to consider two cases depending on how derivation (4.2) begins.

Case 1. The first step of derivation (4.2) is

$$[A, X_1 \ldots X_{i-1} Y] \underset{L}{\Rightarrow} [B, Y][A, X_1 \ldots X_{i-1} B]$$

for some B in N, $X_i R_{Lc}^* B$. In this case $i>1$ and, by Lemma 4.2, derivation (4.2) may be written as

$$\begin{aligned}[A, X_1 \ldots X_{i-1} Y] &\underset{L}{\Rightarrow} [B, Y] \underline{[A, X_1 \ldots X_{i-1} B]} \\ &\underset{L}{\Rightarrow}^* y[B, Y\alpha] \underline{[A, X_1 \ldots X_{i-1} B]} \underset{L}{\Rightarrow} y[A, X_1 \ldots X_{i-1} B] \\ &\underset{L}{\Rightarrow}^* yz[A, X_1 \ldots X_q] \underset{L}{\Rightarrow} yz = x\end{aligned}$$

for some α in $(N \cup T)^*$. Then, applying the induction hypothesis to the derivations $[B, Y] \underset{L}{\Rightarrow}^* y[B, Y\alpha] \underset{L}{\Rightarrow} y$ and

$$[A, X_1 \ldots X_{i-1} B] \underset{L}{\Rightarrow}^* z[A, X_1 \ldots X_q] \underset{L}{\Rightarrow} z,$$

we obtain in G' the derivations

$$B \underset{R}{\Rightarrow} \underline{Y}\alpha \underset{R}{\Rightarrow}{}^{*} \underline{Y}y$$

and

$$A \underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}}X_i \dots X_q \underset{R}{\Rightarrow}{}^{*} \underline{X_1 \dots X_{i-1}}Bz.$$

Thus we have in G' the derivation

$$\begin{aligned} A &\underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}}X_i \dots X_q \\ &\underset{R}{\Rightarrow}{}^{*} \underline{X_1 \dots X_{i-1}}\,Yyz = X_1 \dots X_{i-1}\,Yx, \end{aligned}$$

where $X_i R_{Lc}^{*} Y$, as desired.

Case 2. The first step of derivation (4.2) is

$$[A, X_1 \dots X_{i-1} X_i] \underset{L}{\Rightarrow} a[A, X_1 \dots X_{i-1} X_i a]$$

for some a in T such that $X_{i+1} R_{Lc}^{*} a$. In this case $q \geqq 2$ and derivation (4.2) may be written as

$$\begin{aligned} [A, X_1 \dots X_i] &\underset{L}{\Rightarrow} a[A, X_1 \dots X_i a] \\ &\underset{L}{\Rightarrow}{}^{*} ay[A, X_1 \dots X_q] \underset{L}{\Rightarrow} ay = x. \end{aligned}$$

Applying the induction hypothesis to the derivation

$$[A, X_1 \dots X_i a] \underset{L}{\Rightarrow}{}^{*} y[A, X_1 \dots X_q] \underset{L}{\Rightarrow} y,$$

we conclude that there is in G' a derivation

$$\begin{aligned} A &\underset{R}{\Rightarrow} \underline{X_1 \dots X_i}X_{i+1} \dots X_q \\ &\underset{R}{\Rightarrow}{}^{*} \underline{X_1 \dots X_i}ay = X_1 \dots X_i x. \end{aligned}$$

The proof of Case 2 is thus complete. $\square$

Lemma 4.4. *Let $A \to X_1 \dots X_q$, $q \geqq 2$, be a production of G'. For all X_i, $2 \leqq i \leqq q$, and for all Y in $N \cup T \cup \{\varepsilon\}$ such that $X_i R_{Lc}^{*} Y$, there exists in G' a derivation*

$$\begin{aligned} &A \underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}}X_i \dots X_q \underset{R}{\Rightarrow}{}^{*} \underline{X_1 \dots X_{i-1}}Yy, \\ &\underline{X_1} \dots X_{i-1}\,Yy \underset{R}{\Rightarrow}{}^{*} \underline{X_1}xy, \end{aligned} \tag{4.6}$$

where Y is derived from X_i, if and only if there exists in $T(G)$ a derivation

$$[A, X_1] \underset{L}{\Rightarrow}{}^{*} x[A, X_1 \dots X_{i-1} Y] \underset{L}{\Rightarrow}{}^{*} xy[A, X_1 \dots X_q] \underset{L}{\Rightarrow} xy. \tag{4.7}$$

Proof. We prove the lemma by induction on the length of the derivation segment $\underline{X_1} \dots X_{i-1}\,Yy \underset{R}{\Rightarrow}{}^{*} \underline{X_1}xy$ in (4.6), as well as on the length of the derivation segment $[A, X_1] \underset{L}{\Rightarrow}{}^* x[A, X_1 \dots X_{i-1} Y]$ in (4.7). Initially, the case that the length of one of these two derivation segments is equal to 0 is directly concluded by Lemma 4.3.

Let n be a non-negative integer. Assume first that (4.6) implies (4.7) for any symbol X_i, $i \geqq 2$, in any production $A \to X_1 \dots X_q$, $q \geqq 2$, provided that the length of the derivation segment $\underline{X_1} \dots X_{i-1} Yy \underset{R}{\Rightarrow}{}^* \underline{X_1} xy$ in (4.6) is less than or equal to n. Then let the length of this derivation segment in (4.6) for an arbitrary X_i, $i \geqq 2$, be equal to $n+1$.

Consider the symbols $Y_1 = X_1$, $Y_2 = X_2, \dots, Y_{i-1} = X_{i-1}$, $Y_i = Y$, and let Y_j, $1 \leqq j \leqq i$, be the last nonterminal in this sequence, i.e. Y_j is in N and $Y_{j+1}, \dots, Y_i$ are not in N. (The case in which no such Y_j exists is ruled out, since the length of the derivation segment $Y_1 \dots Y_{i-1} Y_i y = \underline{X_1} \dots X_{i-1} Yy \underset{R}{\Rightarrow}{}^* \underline{X_1} xy$ is assumed to be $n+1 \geqq 1$.) Since $Y_{j+1}, \dots, Y_i$ are terminal symbols (Y_i may also be ε), derivation (4.6) can be written in the form

$$\begin{aligned}
A &\underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}{}^* \underline{X_1 \dots X_{i-1}} Yy \\
&= Y_1 \dots Y_{j-1} Y_j Y_{j+1} \dots Y_i y \\
&\underset{R}{\Rightarrow} \underline{Y_1 \dots Y_{j-1}} Z\alpha Y_{j+1} \dots Y_i y \underset{R}{\Rightarrow}{}^* \underline{Y_1 \dots Y_{j-1}} Z x_2 Y_{j+1} \dots Y_i y, \\
&\quad \underline{Y_1} \dots Y_{j-1} Z x_2 Y_{j+1} \dots Y_i y \underset{R}{\Rightarrow}{}^* \underline{Y_1} x_1 x_2 Y_{j+1} \dots Y_i y = X_1 xy
\end{aligned} \tag{4.8}$$

for some string $Z\alpha$ in $(N \cup T)^*$ such that $Y_j \to Z\alpha$ is a production of G. Since in (4.8) the length of the derivation segment

$$\underline{Y_1} \dots Y_{j-1} Z x_2 Y_{j+1} \dots Y_i y \underset{R}{\Rightarrow}{}^* \underline{Y_1} x_1 x_2 Y_{j+1} \dots Y_i y,$$

where $Y_1 \dots Y_{j-1} = X_1 \dots X_{j-1}$, is equal to n, and $X_j R_{Lc}^* Z$, we obtain by applying the induction hypothesis the derivation

$$[A, X_1] \underset{L}{\Rightarrow}{}^* x_1 [A, X_1 \dots X_{j-1} Z]$$

in $T(G)$. Further, it follows from (4.8) by the construction of $T(G)$ and Lemma 4.3 that

$$\begin{aligned}
[A, X_1 \dots X_{j-1} Z] &\underset{L}{\Rightarrow} [Y_j, Z][A, X_1 \dots X_{j-1} Y_j] \\
&\underset{L}{\Rightarrow}{}^* x_2 [A, X_1 \dots X_{j-1} Y_j] \\
&\underset{L}{\Rightarrow}{}^* x_2 Y_{j+1} \dots Y_i [A, X_1 \dots X_{j-1} Y_j Y_{j+1} \dots Y_i] \\
&= x_2 Y_{j+1} \dots Y_i [A, X_1 \dots X_{i-1} Y].
\end{aligned}$$

Thus we have the derivation

$$\begin{aligned}
[A, X_1] &\underset{L}{\Rightarrow}{}^* x_1 x_2 Y_{j+1} \dots Y_i [A, X_1 \dots X_{i-1} Y] \\
&= x[A, X_1 \dots X_{i-1} Y],
\end{aligned}$$

which, by Lemma 4.3, can be made to proceed as desired:

$$x[A, X_1 \dots X_{i-1} Y] \underset{L}{\Rightarrow}{}^* xy[A, X_1 \dots X_q] \underset{L}{\Rightarrow} xy.$$

Assume in the second place that (4.7) implies (4.6) for any symbol X_i, $i \geqq 2$, in any production $A \to X_1 \dots X_q$, $q \geqq 2$, provided that the length of the derivation segment $[A, X_1] \underset{L}{\Rightarrow}^* x[A, X_1 \dots X_{i-1} Y]$ in (4.7) is less than or equal to n. Then let the length of this derivation segment in (4.7) for an arbitrary X_i, $i \geqq 2$, be equal to $n+1$. To show that there is a respective derivation (4.6), we consider three different cases depending on whether the symbol Y appearing in $[A, X_1 \dots X_{i-1} Y]$ is a nonterminal, a terminal or the empty string ε.

Case 1. If Y is a nonterminal, the symbol $[A, X_1 \dots X_{i-1} Y]$ is introduced into derivation (4.7) in the following way:

$$\begin{aligned}
[A, X_1] &\underset{L}{\Rightarrow}^* x_1 [A, X_1 \dots X_{i-1} Z] \\
&\underset{L}{\Rightarrow} x_1 [Y, Z] \underline{[A, X_1 \dots X_{i-1} Y]} \\
&\underset{L}{\Rightarrow}^* x_1 x_2 \underline{[A, X_1 \dots X_{i-1} Y]} \\
&\underset{L}{\Rightarrow}^* x_1 x_2 y [A, X_1 \dots X_q] \underset{L}{\Rightarrow} x_1 x_2 y = xy,
\end{aligned}$$

where Z is in $N \cup T \cup \{\varepsilon\}$ and $X_i R_{Lc}^* Z$. Applying the induction hypothesis to the derivation

$$\begin{aligned}
[A, X_1] &\underset{L}{\Rightarrow}^* x_1 [A, X_1 \dots X_{i-1} Z] \\
&\underset{L}{\Rightarrow}^* x_1 x_2 y [A, X_1 \dots X_q] \underset{L}{\Rightarrow} x_1 x_2 y
\end{aligned}$$

we conclude that there is in G' a derivation

$$\begin{aligned}
A &\underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}^* \underline{X_1 \dots X_{i-1}} Z x_2 y, \\
&\underline{X_1} \dots X_{i-1} Z x_2 y \underset{R}{\Rightarrow}^* \underline{X_1} x_1 x_2 y = X_1 x y.
\end{aligned}$$

Further, by Lemma 4.3 we have in G' the derivations

$$Y \underset{R}{\Rightarrow}^* Z x_2 \quad \text{and} \quad A \Rightarrow \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}^* \underline{X_1 \dots X_{i-1}} Y y.$$

Hence, we conclude that in the grammar G'

$$\begin{aligned}
A &\underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}^* \underline{X_1 \dots X_{i-1}} Y y, \\
&\underline{X_1} \dots X_{i-1} Y y \underset{R}{\Rightarrow}^* \underline{X_1} x_1 x_2 y = X_1 x y,
\end{aligned}$$

as desired.

Case 2. If Y is a terminal symbol, derivation (4.7) can be written in the form

$$\begin{aligned}
[A, X_1] &\underset{L}{\Rightarrow}^* x' [A, X_1 \dots X_{i-1}] \underset{L}{\Rightarrow} x' Y [A, X_1 \dots X_{i-1} Y] \\
&\underset{L}{\Rightarrow}^* x' Y y [A, X_1 \dots X_q] \underset{L}{\Rightarrow} x' Y y = xy.
\end{aligned}$$

By Lemma 4.3 we have in G' the derivation

$$A \underset{R}{\Rightarrow} \underline{X_1 \dots X_{i-1}} X_i \dots X_q \underset{R}{\Rightarrow}^* \underline{X_1 \dots X_{i-1}} Y y$$

which is the desired derivation in the case $i=2$ (then $x'=\varepsilon$ and $Yy=xy$). Otherwise the induction hypothesis implies that this derivation can be continued by the derivation

$$\underline{X_1} \ldots X_{i-1} Yy \underset{R}{\Rightarrow}{}^* \underline{X_1} x' Yy,$$

where $x' Yy = xy$, as desired.

Case3. If Y is ε, then X_{i-1} is a nonterminal or a terminal. In addition, $i \geqq 3$, since the length of the derivation

$$[A, X_1] \underset{L}{\Rightarrow}{}^* x[A, X_1 \ldots X_{i-1} Y]$$

was assumed to be greater than zero. Thus we may conclude from Case 1 and Case 2 that

$$\begin{aligned} A &\underset{R}{\Rightarrow} \underline{X_1 \ldots X_{i-2}} X_{i-1} \ldots X_q \\ &\underset{R}{\Rightarrow}{}^* \underline{X_1 \ldots X_{i-2}} X_{i-1} y, \\ \underline{X_1} \ldots X_{i-1} y &\underset{R}{\Rightarrow}{}^* \underline{X_1} xy, \end{aligned}$$

where $X_1 \ldots X_{i-1} y = X_1 \ldots X_{i-1} Y_y$, as desired. □

Now we are ready to turn our attention to the 'global' correspondence between the rightmost derivations in the given grammar and the leftmost derivations in the transformed grammar.

We first define that if τ is a string of nonterminals of $T(G)$ such that τ is of the form $[A, \alpha_1 B_1][A_2, \alpha_2 B_2] \ldots [A_n, \alpha_n B_n]$, $n \geqq 1$, then $(\tau)_T = \alpha_n \alpha_{n-1} \ldots \alpha_1$. In particular, we define that $(\varepsilon)_T = \varepsilon$.

Lemma 4.5. *There exists in G' a rightmost derivation*

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z \underset{R}{\Rightarrow} \alpha \underline{X_1} X_2 \ldots X_q z \underset{R}{\Rightarrow}{}^* \alpha \underline{X_1} yz \underset{R}{\Rightarrow}{}^* \perp xyz \tag{4.9}$$

if and only if there exists in $T(G)$ a leftmost derivation

$$[S', \perp] \underset{L}{\Rightarrow}{}^* x[A, X_1] \underline{\tau} \underset{L}{\Rightarrow}{}^* xy[A, X_1 \ldots X_q] \underline{\tau} \underset{L}{\Rightarrow} xy\tau \underset{L}{\Rightarrow}{}^* xyz, \tag{4.10}$$

where $(\tau)_T = \alpha$. (Here $q \geqq 0$; if $q=0$, then $X_1 \ldots X_q = \varepsilon$ and hence $X_1 = \varepsilon$.)

Proof. We prove the lemma by induction on the length of the derivation segment $S' \underset{R}{\Rightarrow}{}^* \alpha A z$ in (4.9), as well as on the length of the derivation segment $[S', \perp] \underset{L}{\Rightarrow}{}^* x[A, X_1]\tau$ in (4.10). By Lemma 4.3, $S' \underset{R}{\Rightarrow} \perp S \underset{R}{\Rightarrow}{}^* \perp x$ if and only if $[S', \perp] \underset{L}{\Rightarrow}{}^* x[S', \perp S] \underset{L}{\Rightarrow} x$. Hence we conclude the initial step of the induction, when the lengths of the derivation segments involved are 0.

Let n be a non-negative integer. Assume first that (4.9) implies (4.10) whenever the length of the derivation segment $S' \underset{R}{\Rightarrow}{}^* \alpha A z$ in (4.9) is less than or equal to n. Then consider the case in which the length of this derivation segment in (4.9) is equal to $n+1$. Assume that the symbol A in the right sentential form $\alpha A z$ in (4.9) has been produced by the production $B \to Y_1 \ldots Y_{i-1} C Y_{i+1} \ldots Y_r$,

$r \geqq 2$, $i \geqq 2$, of G' such that $CR^*_{Lc}A$ (this production always exists, since the unique production for the start symbol S' is $S' \to \bot S$):

$$\begin{aligned} S' \underset{R}{\Rightarrow}{}^* \gamma B z_2 &\underset{R}{\Rightarrow} \underline{\gamma Y_1 \dots Y_{i-1}} C Y_{i+1} \dots Y_r z_2 \\ &\underset{R}{\Rightarrow}{}^* \underline{\gamma Y_1 \dots Y_{i-1}} A z_1 z_2 \\ &\underset{R}{\Rightarrow} \underline{\gamma Y_1 \dots Y_{i-1} X_1} \dots X_q z_1 z_2 \\ &\underset{R}{\Rightarrow}{}^* \underline{\gamma Y_1 \dots Y_{i-1} X_1} y z_1 z_2, \\ \underline{\gamma Y_1} \dots Y_{i-1} X_1 y z_1 z_2 &\underset{R}{\Rightarrow}{}^* \underline{\gamma Y_1} x_2 y z_1 z_2 \underset{R}{\Rightarrow}{}^* \bot x_1 x_2 y z_1 z_2, \end{aligned} \tag{4.11}$$

where $\gamma Y_1 \dots Y_{i-1} = \alpha$, $x_1 x_2 = x$ and $z_1 z_2 = z$. Applying the induction hypothesis to the derivation

$$\begin{aligned} S' &\underset{R}{\Rightarrow}{}^* \gamma B z_2 \underset{R}{\Rightarrow} \underline{\gamma Y_1} \dots Y_r z_2 \\ &\underset{R}{\Rightarrow}{}^* \underline{\gamma Y_1} x_2 y z_1 z_2 \underset{R}{\Rightarrow}{}^* \bot x_1 x_2 y z_1 z_2 \end{aligned}$$

we conclude that there exists in $T(G)$ a derivation

$$\begin{aligned} [S', \bot] &\underset{L}{\Rightarrow}{}^* x_1 [B, Y_1] \underline{\eta} \underset{L}{\Rightarrow}{}^* x_1 x_2 y z_1 [B, Y_1 \dots Y_r] \underline{\eta} \\ &\underset{L}{\Rightarrow} x_1 x_2 y z_1 \eta \underset{L}{\Rightarrow}{}^* x_1 x_2 y z_1 z_2, \end{aligned}$$

such that $(\eta)_T = \gamma$. By Lemma 4.4 it further follows from (4.11) that we have in $T(G)$ the derivations

$$[B, Y_1] \underset{L}{\Rightarrow}{}^* x_2 y [B, Y_1 \dots Y_{i-1} A] \underset{L}{\Rightarrow}{}^* x_2 y z_1 [B, Y_1 \dots Y_r]$$

and

$$[B, Y_1] \underset{L}{\Rightarrow}{}^* x_2 [B, Y_1 \dots Y_{i-1} X_1] \underset{L}{\Rightarrow}{}^* x_2 y z_1 [B, Y_1 \dots Y_r].$$

Moreover, by the construction of $T(G)$, derivation (4.11) implies that $T(G)$ has the production

$$[B, Y_1 \dots Y_{i-1} X_1] \to [A, X_1][B, Y_1 \dots Y_{i-1} A].$$

Since we additionally know by Lemma 4.3 that

$$[A, X_1] \underset{L}{\Rightarrow}{}^* y [A, X_1 \dots X_q] \underset{L}{\Rightarrow} y,$$

we obtain in $T(G)$ the derivation

$$\begin{aligned} [S', \bot] &\underset{L}{\Rightarrow}{}^* x_1 x_2 [A, X_1] \underline{[B, Y_1 \dots Y_{i-1} A] \eta} \\ &\underset{L}{\Rightarrow}{}^* x_1 x_2 y [A, X_1 \dots X_q] \underline{[B, Y_1 \dots Y_{i-1} A] \eta} \\ &\underset{L}{\Rightarrow} x_1 x_2 y [B, Y_1 \dots Y_{i-1} A] \eta \\ &\underset{L}{\Rightarrow}{}^* x_1 x_2 y z_1 z_2, \end{aligned}$$

where $x_1 x_2 = x$, $z_1 z_2 = z$ and $([B, Y_1 \dots Y_{i-1} A] \eta)_T = \gamma Y_1 \dots Y_{i-1} = \alpha$, as desired.

Assume in the second place that a derivation (4.10) implies a derivation (4.9) whenever the length of the derivation segment $[S', \perp] \underset{L}{\Rightarrow}^* x[A, X_1]\tau$ in (4.10) is less than or equal to n. Then let the length of this derivation segment in (4.10) be equal to $n+1$.

Since $n+1 \geqq 1$, we can conclude by the construction of $T(G)$ that derivation (4.10) is of the form

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}^* x[B, Y_1 \dots Y_{i-1} X_1]\eta \\ &\underset{L}{\Rightarrow} x[A, X_1][B, Y_1 \dots Y_{i-1} A]\eta \underset{L}{\Rightarrow}^* xyz,\end{aligned}$$

where $[B, Y_1 \dots Y_{i-1} A]\eta = \tau$ and $i \geqq 2$. Hence, applying Lemma 4.1 and Lemma 4.2, we conclude that derivation (4.10) may be written as

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}^* x_1[B, Y_1]\eta \underset{L}{\Rightarrow}^* x_1 x_2 [B, Y_1 \dots Y_{i-1} X_1]\eta \\ &\underset{L}{\Rightarrow} x_1 x_2 [A, X_1]\underline{[B, Y_1 \dots Y_{i-1} A]\eta} \\ &\underset{L}{\Rightarrow}^* x_1 x_2 y [A, X_1 \dots X_q]\underline{[B, Y_1 \dots Y_{i-1} A]\eta} \\ &\underset{L}{\Rightarrow} x_1 x_2 y [B, Y_1 \dots Y_{i-1} A]\underline{\eta} \underset{L}{\Rightarrow}^* x_1 x_2 y z_1 [B, Y_1 \dots Y_r]\underline{\eta} \\ &\underset{L}{\Rightarrow} x_1 x_2 y z_1 \eta \underset{L}{\Rightarrow}^* x_1 x_2 y z_1 z_2,\end{aligned} \tag{4.12}$$

where $x_1 x_2 = x$, $z_1 z_2 = z$, $r \geqq 2$, and $2 \leqq i \leqq r$. Now, applying the induction hypothesis to the derivation

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}^* x_1[B, Y_1]\underline{\eta} \underset{L}{\Rightarrow}^* x_1 x_2 y z_1 [B, Y_1 \dots Y_r]\underline{\eta} \\ &\underset{L}{\Rightarrow} x_1 x_2 y z_1 \eta \underset{L}{\Rightarrow}^* x_1 x_2 y z_1 z_2\end{aligned}$$

we conclude that there is in G' a derivation

$$\begin{aligned}S' &\underset{R}{\Rightarrow}^* \gamma B z_2 \underset{R}{\Rightarrow} \underline{\gamma Y_1} \dots Y_r z_2 \underset{R}{\Rightarrow}^* \underline{\gamma Y_1} x_2 y z_1 z_2 \\ &\underset{R}{\Rightarrow}^* \perp x_1 x_2 y z_1 z_2,\end{aligned}$$

where $\gamma = (\eta)_T$. Furthermore, since it follows from (4.12) by Lemma 4.4 that we have the derivation

$$\begin{aligned}B &\underset{R}{\Rightarrow} \underline{Y_1 \dots Y_{i-1}} Y_i \dots Y_r \underset{R}{\Rightarrow}^* \underline{Y_1 \dots Y_{i-1}} A z_1 \\ &\underset{R}{\Rightarrow} \underline{Y_1 \dots Y_{i-1} X_1} \dots X_q z_1 \underset{R}{\Rightarrow}^* \underline{Y_1 \dots Y_{i-1} X_1} y z_1, \\ \underline{Y_1} &\dots Y_{i-1} X_1 y z_1 \underset{R}{\Rightarrow}^* \underline{Y_1} x_2 y z_1,\end{aligned}$$

we conclude that there exists in G' a derivation

$$\begin{aligned}S' &\underset{R}{\Rightarrow}^* \gamma Y_1 \dots Y_{i-1} A z_1 z_2 \\ &\underset{R}{\Rightarrow} \underline{\gamma Y_1 \dots Y_{i-1} X_1} \dots X_q z_1 z_2 \\ &\underset{R}{\Rightarrow}^* \underline{\gamma Y_1 \dots Y_{i-1} X_1} y z_1 z_2 \underset{R}{\Rightarrow}^* \perp x_1 x_2 y z_1 z_2,\end{aligned}$$

where $x_1 x_2 = x$, $z_1 z_2 = z$ and $\gamma Y_1 \dots Y_{i-1} = ([B, Y_1 \dots Y_{i-1} A]\eta)_T = (\tau)_T$, as desired. □

We have now established the correspondence between the rightmost derivations in the given grammar G and the leftmost derivations in the transformed grammar $T(G)$, and we are able to prove the following major results of the present paper. Theorem 4.1 states that $L(T(G)) = L(G)$, and Theorem 4.2 and Theorem 4.3 imply that for a positive integer k the given grammar G is $PLR(k)$ if and only if the transformed grammar $T(G)$ is $LL(k)$.

Theorem 4.1. *For a grammar* G, $L(T(G)) = L(G)$.

Proof. By Lemma 4.3 there is in G' a derivation $S' \underset{R}{\Rightarrow}{}^* \perp w$ if and only if there is in $T(G)$ a derivation $[S', \perp] \underset{L}{\Rightarrow}{}^* w$. Since $L(G') = \{\perp\} L(G)$, we conclude the theorem. $\square$

Theorem 4.2. *Let k be a positive integer. If a grammar G is $PLR(k)$, then the transformed grammar $T(G)$ is $LL(k)$.*

Proof. We assume that for a $PLR(k)$ grammar G the transformed grammar $T(G)$ is not $LL(k)$. Then there exist in $T(G)$ two leftmost derivations

$$[S', \perp] \underset{L}{\Rightarrow}{}^* x[A, \alpha Y]\tau \underset{L}{\Rightarrow} x\eta_1\tau \underset{L}{\Rightarrow}{}^* xz' \tag{4.13}$$

and

$$[S', \perp] \underset{L}{\Rightarrow}{}^* x[A, \alpha Y]\tau \underset{L}{\Rightarrow} x\eta_2\tau \underset{L}{\Rightarrow}{}^* xz'' \tag{4.14}$$

such that $\eta_1 \neq \eta_2$, although $k\!:\!z' = k\!:\!z''$. This situation may occur in three different ways depending on how the derivations are continued after the left sentential form $x[A, \alpha Y]\tau$. (Note that $\eta_1 \neq \eta_2$ implies that $\alpha Y \neq \varepsilon$; in addition, together with the condition $k\!:\!z' = k\!:\!z''$ where $k > 0$, the inequality $\eta_1 \neq \eta_2$ implies that both η_1 and η_2 cannot be of the form $a[A, \alpha Ya]$.)

Case 1. The right-hand sides η_1 and η_2 in the above derivations are both of the form $[B, Y][A, \alpha B]$, where B is a nonterminal of G. In this case there are in $T(G)$ derivations

$$\begin{aligned} [S', \perp] &\underset{L}{\Rightarrow}{}^* x[A, \alpha Y]\tau \underset{L}{\Rightarrow} x[B_1, Y]\underline{[A, \alpha B_1]\tau} \\ &\underset{L}{\Rightarrow}{}^* xy_1[B_1, Y\beta_1]\underline{[A, \alpha B_1]\tau} \\ &\underset{L}{\Rightarrow} xy_1[A, \alpha B_1]\tau \underset{L}{\Rightarrow}{}^* xy_1z_1 \end{aligned}$$

and

$$\begin{aligned} [S', \perp] &\underset{L}{\Rightarrow}{}^* x[A, \alpha Y]\tau \underset{L}{\Rightarrow} x[B_2, Y]\underline{[A, \alpha B_2]\tau} \\ &\underset{L}{\Rightarrow}{}^* xy_2[B_2, Y\beta_2]\underline{[A, \alpha B_2]\tau} \\ &\underset{L}{\Rightarrow} xy_2[A, \alpha B_2]\tau \underset{L}{\Rightarrow}{}^* xy_2z_2, \end{aligned}$$

where $B_1 \neq B_2$ and $k\!:\!y_1z_1 = k\!:\!y_2z_2$. Applying Lemma 4.5 to these derivations, we conclude that there exist in G' derivations

$$S' \underset{R}{\Rightarrow}{}^* \gamma B_1 z_1 \underset{R}{\Rightarrow} \underline{\gamma Y}\beta_1 z_1 \underset{R}{\Rightarrow}{}^* \underline{\gamma Y} y_1 z_1 \underset{R}{\Rightarrow}{}^* xy_1z_1$$

and

$$S' \underset{R}{\Rightarrow}{}^* \gamma B_2 z_2 \underset{R}{\Rightarrow} \underline{\gamma Y} \beta_2 z_2 \underset{R}{\Rightarrow}{}^* \underline{\gamma Y} y_2 z_2 \underset{R}{\Rightarrow}{}^* x y_2 z_2,$$

where $\gamma = ([A, \alpha B_1]\tau)_T = ([A, \alpha B_2]\tau)_T$. We therefore have a contradiction of the assumption that G is $PLR(k)$, since $B_1 \neq B_2$ although $k: y_1 z_1 = k: y_2 z_2$.

Case 2. Either one of the right-hand sides η_1 and η_2 in derivations (4.13) and (4.14), respectively, is $[B, Y][A, \alpha B]$ for some nonterminal B of G. We assume that η_1 is of this form; then we can write derivation (4.13) as

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x[A, \alpha Y]\tau \underset{L}{\Rightarrow} x[B, Y]\underline{[A, \alpha B]\tau} \\ &\underset{L}{\Rightarrow}{}^* x y_1 [B, Y\beta]\underline{[A, \alpha B]\tau} \\ &\underset{L}{\Rightarrow} x y_1 [A, \alpha B]\tau \underset{L}{\Rightarrow}{}^* x y_1 z_1,\end{aligned}$$

where $y_1 z_1 = z'$. Applying Lemma 4.1 and Lemma 4.2 we conclude that derivation (4.14) can be written (note that $|\alpha| > 0$) as

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x'[A, 1:\alpha]\underline{\tau} \underset{L}{\Rightarrow}{}^* x'x''[A, \alpha Y]\underline{\tau} \\ &\underset{L}{\Rightarrow}{}^* x'x'' y_2 [A, \alpha Y\varphi]\underline{\tau} \\ &\underset{L}{\Rightarrow} x'x'' y_2 \tau \underset{L}{\Rightarrow}{}^* x'x'' y_2 z_2,\end{aligned}$$

where $x'x'' = x$ and $y_2 z_2 = z''$.

If we apply Lemma 4.5 to the former of the above derivations, and Lemma 4.4 and Lemma 4.5 to the latter, then we obtain in G' the derivation

$$S' \underset{R}{\Rightarrow}{}^* \gamma B z_1 \underset{R}{\Rightarrow} \underline{\gamma Y} \beta z_1 \underset{R}{\Rightarrow}{}^* \underline{\gamma Y} y_1 z_1 \underset{R}{\Rightarrow}{}^* x y_1 z_1,$$

where $\gamma = ([A, \alpha B]\tau)_T$, and, respectively, the derivation

$$S' \underset{R}{\Rightarrow}{}^* \delta A z_2 \underset{R}{\Rightarrow} \underline{\delta \alpha Y} \varphi z_2 \underset{R}{\Rightarrow}{}^* \underline{\delta \alpha Y} y_2 z_2 \underset{R}{\Rightarrow}{}^* x y_2 z_2,$$

where $\delta = (\tau)_T$. But since $\gamma = \delta\alpha$, we can conclude that $\gamma B \neq \delta A$, because the length of the string α is greater than or equal to 1. Since $k: y_1 z_1 = k: y_2 z_2$, we thus have a contradiction of the assumption that the grammar G is $PLR(k)$.

Case 3. Neither of the right-hand sides η_1 and η_2 in (4.13) and (4.14), respectively, is of the form $[B, Y][A, \alpha B]$. This case includes the subcase that both η_1 and η_2 are of the form $[B, \varepsilon][A, \alpha YB]$, the proof of which parallels the proof of Case 1 and is therefore omitted. Otherwise either of the right-hand sides η_1 and η_2 must be the empty string. Assuming that $\eta_1 = \varepsilon$ and applying Lemma 4.1 to (4.13) and Lemmas 4.1 and 4.2 to (4.14), we conclude that derivations (4.13) and (4.14) may be written as

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x'[A, 1:\alpha Y]\underline{\tau} \underset{L}{\Rightarrow}{}^* x'x''[A, \alpha Y]\underline{\tau} \\ &\underset{L}{\Rightarrow} x'x''\tau \underset{L}{\Rightarrow}{}^* x'x''z'\end{aligned}$$

and, respectively,

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x'[A, 1{:}\alpha Y]\underline{\tau} \underset{L}{\Rightarrow}{}^* x'x''[A, \alpha Y]\underline{\tau} \\ &\underset{L}{\Rightarrow}{}^* x'x''y_2[A, \alpha Y\beta]\underline{\tau} \\ &\underset{L}{\Rightarrow} x'x''y_2\tau \underset{L}{\Rightarrow}{}^* x'x''y_2z_2,\end{aligned}$$

where $x'x''=x$, $|\beta|>0$ and $y_2z_2=z''$. Applying Lemma 4.5 to the former and Lemmas 4.4 and 4.5 to the latter derivation we obtain in G' the derivations

$$S' \underset{R}{\Rightarrow}{}^* \gamma Az' \underset{R}{\Rightarrow} \gamma\alpha Yz' \underset{R}{\Rightarrow}{}^* xz'$$

and

$$S' \underset{R}{\Rightarrow}{}^* \gamma Az_2 \underset{R}{\Rightarrow} \underline{\gamma\alpha Y}\beta z_2 \underset{R}{\Rightarrow}{}^* \underline{\gamma\alpha Y}y_2z_2 \underset{R}{\Rightarrow}{}^* xy_2z_2,$$

where $\gamma=(\tau)_T$. But since $\gamma\alpha Y\beta \neq \gamma\alpha Y$ and $k{:}z'=k{:}y_2z_2$, we know by Theorem 2.1 that the grammar G cannot be $LR(k)$, which by definition contradicts the assumption that G is $PLR(k)$. □

We now prove the converse of Theorem 4.2:

Theorem 4.3. *Let k be a non-negative integer. A grammar G is $PLR(k)$ if the transformed grammar $T(G)$ is $LL(k)$.*

Proof. We assume that a grammar G is not $PLR(k)$ and we show that the transformed grammar $T(G)$ cannot then be $LL(k)$. We first note that if G is ambiguous then by Theorem 4.4 to be given in Sect. 4.3 also $T(G)$ is ambiguous, which means that $T(G)$ is not $LL(k)$.

In the rest of the proof we assume that G is not ambiguous. We have two cases to consider depending on whether G contradicts the special condition of the definition of $PLR(k)$ grammars or only the $LR(k)$ condition. We first consider the former case.

Case 1. There exist in G' two rightmost derivations

$$S' \underset{R}{\Rightarrow}{}^* \alpha Az_1 \underset{R}{\Rightarrow} \underline{\alpha X}\beta z_1 \underset{R}{\Rightarrow}{}^* \underline{\alpha X}y_1z_1 \underset{R}{\Rightarrow}{}^* \perp xy_1z_1, \tag{4.15}$$

$X\beta \neq \varepsilon$, and

$$S' \underset{R}{\Rightarrow}{}^* \alpha' Bz_2 \underset{R}{\Rightarrow} \underline{\alpha'\alpha'' X}\gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\alpha'' X}y_2z_2 \underset{R}{\Rightarrow}{}^* \perp xy_2z_2 \tag{4.16}$$

such that $\alpha A \neq \alpha' B$, although $\alpha'\alpha''=\alpha$ and $k{:}y_1z_1=k{:}y_2z_2$. This case is further divided into two subcases depending on whether $|\alpha A|=|\alpha' B|$ or not.

Case 1a. The strings αA and $\alpha' B$ are of equal length. In this case, since $\alpha'\alpha''=\alpha$ in the above derivations, the strings α' and α must be equal, $\alpha''=\varepsilon$, and the inequality $\alpha A \neq \alpha' B$ implies that $A \neq B$. Thus, applying Lemma 4.5 to derivations (4.15) and (4.16), we conclude that there exist in $T(G)$ derivations

$$[S', \perp] \underset{L}{\Rightarrow}{}^* x[A, X]\underline{\tau_1} \underset{L}{\Rightarrow}{}^* xy_1[A, X\beta]\underline{\tau_1} \underset{L}{\Rightarrow} xy_1\tau_1 \underset{L}{\Rightarrow}{}^* xy_1z_1 \tag{4.17}$$

and

$$[S', \perp] \underset{L}{\Rightarrow}{}^* x[B, X]\underline{\tau_2} \underset{L}{\Rightarrow}{}^* xy_2[B, X\gamma]\underline{\tau_2} \underset{L}{\Rightarrow} xy_2\tau_2 \underset{L}{\Rightarrow}{}^* xy_2z_2, \tag{4.18}$$

where $[A, X] \neq [B, X]$ and $(\tau_1)_T = (\tau_2)_T = \alpha$. But since $k:y_1z_1 = k:y_2z_2$, we can conclude by Lemma 2.1 that if $T(G)$ were $LL(k)$ then either derivation (4.17) would be of the form

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x[B, X]\tau_2 \underset{L}{\Rightarrow}{}^+ x[A, X]\underline{\tau_1} \\ &\underset{L}{\Rightarrow}{}^* xy_1[A, X\beta]\underline{\tau_1} \underset{L}{\Rightarrow}{}^* xy_1z_1\end{aligned} \tag{4.19}$$

or derivation (4.18) would be of the form

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x[A, X]\tau_1 \underset{L}{\Rightarrow}{}^+ x[B, X]\underline{\tau_2} \\ &\underset{L}{\Rightarrow}{}^* xy_2[B, X\gamma]\underline{\tau_2} \underset{L}{\Rightarrow}{}^* xy_2z_2.\end{aligned} \tag{4.20}$$

We shall subsequently show that this is impossible, which means that the transformed grammar $T(G)$ cannot satisfy the $LL(k)$ condition.

Suppose that there is in $T(G)$ a derivation of the form (4.19). (The case in which there is a derivation of the form (4.20) is handled similarly.) Then, since derivation (4.19) can also, by Lemma 4.2, be expressed as

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x[B, X]\underline{\tau_2} \underset{L}{\Rightarrow}{}^* xy'_1[B, X\beta']\underline{\tau_2} \\ &\underset{L}{\Rightarrow} xy'_1\tau_2 \underset{L}{\Rightarrow}{}^* xy'_1z'_1 = xy_1z_1,\end{aligned}$$

and $(\tau_2)_T = \alpha$, we conclude by Lemma 4.5 that there exists in G' a derivation

$$S' \underset{R}{\Rightarrow}{}^* \alpha Bz'_1 \underset{R}{\Rightarrow} \underline{\alpha X}\beta' z'_1 \underset{R}{\Rightarrow}{}^* \underline{\alpha X} y'_1 z'_1 \underset{R}{\Rightarrow}{}^* \perp xy'_1z'_1.$$

But together with derivation (4.15) this means, since $A \neq B$ and $X \neq \varepsilon$, that the string $\alpha X y_1 z_1 = \alpha X y'_1 z'_1$ and hence the string $\perp xy_1z_1 = \perp xy'_1z'_1$ has in G' at least two different right parses to S'. Thus we have a contradiction of the assumption that the grammar G is unambiguous, and the proof of Case 1a is complete.

Case 1b. The strings αA and $\alpha' B$ in (4.15) and (4.16), respectively, are not of equal length. As in Case 1a we conclude, applying Lemma 4.5 to derivation (4.15), that there is in $T(G)$ a derivation

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x[A, X]\underline{\tau_1} \underset{L}{\Rightarrow}{}^* xy_1[A, X\beta]\underline{\tau_1} \\ &\underset{L}{\Rightarrow} xy_1\tau_1 \underset{L}{\Rightarrow}{}^* xy_1z_1.\end{aligned}$$

Since $\alpha'\alpha'' = \alpha$, the length of $\alpha' B$ must be less than the length of αA and $|\alpha''| > 0$. Thus, applying Lemma 4.4 and Lemma 4.5 to derivation (4.16), we conclude that there exists in $T(G)$ a derivation

$$\begin{aligned}[S', \perp] &\underset{L}{\Rightarrow}{}^* x'[B, 1:\alpha'']\underline{\tau_2} \underset{L}{\Rightarrow}{}^* x'x''[B, \alpha'' X]\underline{\tau_2} \\ &\underset{L}{\Rightarrow}{}^* x'x''y_2[B, \alpha'' X\gamma]\underline{\tau_2} \underset{L}{\Rightarrow} x'x''y_2\tau_2 \\ &\underset{L}{\Rightarrow}{}^* x'x''y_2z_2,\end{aligned}$$

where $x'x''=x$. Since $|\alpha''|>0$, we conclude that $[A,X]\neq[B,\alpha''X]$. But since $k:y_1z_1=k:y_2z_2$, we can continue the proof using these two derivations, as in Case 1a, and obtain a contradiction of the $LL(k)$-ness of the grammar $T(G)$.

Case 2. If the grammar G contradicts the $LR(k)$ condition, then we conclude by Theorem 2.1 that there exist in G' two rightmost derivations

$$S' \underset{R}{\Rightarrow}{}^* \alpha A z_1 \underset{R}{\Rightarrow} \alpha\beta z_1 \underset{R}{\Rightarrow}{}^* \perp x z_1$$

and

$$S' \underset{R}{\Rightarrow}{}^* \alpha' B z_2 \underset{R}{\Rightarrow} \underline{\alpha'\beta'}\gamma z_2 \underset{R}{\Rightarrow}{}^* \underline{\alpha'\beta'} y_2 z_2 \underset{R}{\Rightarrow}{}^* \perp x y_2 z_2$$

such that $\alpha A\neq\alpha' B$ or $\beta\neq\beta'\gamma$, although $\alpha\beta=\alpha'\beta'$, $|\beta'|>0$ and $k:z_1=k:y_2z_2$. Applying Lemma 4.5 to the former and Lemmas 4.4 and 4.5 to the latter derivation, we obtain in $T(G)$ the derivation

$$[S',\perp] \underset{L}{\Rightarrow}{}^* x[A,\beta]\tau_1 \underset{L}{\Rightarrow} x\tau_1 \underset{L}{\Rightarrow}{}^* xz_1 \tag{4.21}$$

and, respectively, the derivation

$$\begin{aligned}[S',\perp] &\underset{L}{\Rightarrow}{}^* x'[B,1:\beta']\underline{\tau_2} \underset{L}{\Rightarrow}{}^* x'x''[B,\beta']\underline{\tau_2} \\ &\underset{L}{\Rightarrow}{}^* x'x''y_2[B,\beta'\gamma]\underline{\tau_2} \underset{L}{\Rightarrow} x'x''y_2\tau_2 \\ &\underset{L}{\Rightarrow}{}^* x'x''y_2z_2,\end{aligned} \tag{4.22}$$

where $x'x''=x$. Now if $\alpha A\neq\alpha' B$, we conclude that $A\neq B$ or $\beta\neq\beta'$, which means that $[A,\beta]\neq[B,\beta']$. But since $k:z_1=k:y_2z_2$, we can continue the proof using derivations (4.21) and (4.22), as in Case 1a, and conclude that the grammar $T(G)$ cannot be $LL(k)$, as desired. On the other hand if $\alpha A=\alpha' B$ and $\beta\neq\beta'\gamma$, then the symbols $[A,\beta]$ and $[B,\beta']$ must be equal but the right-hand sides by which these symbols are substituted in derivations (4.21) and (4.22) are different. Thus we conclude by definition that the grammar $T(G)$ is not $LL(k)$. □

From Theorem 4.2 and Theorem 4.3 we have:

Corollary 4.1. *Let k be a positive integer. A grammar G is PLR(k) if and only if the transformed grammar T(G) is LL(k).*

Rosenkrantz and Stearns [26] (see also [1]) have shown that there is an algorithm to decide whether a grammar is $LL(k)$ for a given k, $k>0$. Hence, by Corollary 4.1, it can be decided whether a grammar G is $PLR(k)$ for a given k, $k>0$, by first constructing the transformed grammar $T(G)$ and then applying the test of the $LL(k)$-ness to $T(G)$.

Finally, we note that obviously a grammar can be $PLR(0)$ although the transformed grammar is not $LL(0)$, since an $LL(0)$ grammar can generate only one terminal string. However, it is immediate from the proofs of Theorems 4.2 and 4.3 that a grammar is $PLR(0)$ if and only if the transformed grammar is $LL(1)$ and the $LL(0)$ condition is not violated unless the two violating productions are of the form $[A,\alpha]\to a[A,\alpha a]$. That is, if for a left-hand side $[A,\alpha]$ there exist more than one production, then all these productions are of the form $[A,\alpha]\to a[A,\alpha a]$.

4.3. $PLR(k)$ Parsing Scheme

It follows from Theorem 4.1 and Corollary 4.1 that if a grammar G is $PLR(k)$ then the language $L(G)$ can be recognized by $LL(k)$ parsing techniques (see e.g. [1]) applied to the grammar $T(G)$. In addition, the $LL(k)$ parser for the transformed grammar $T(G)$ can be used as a deterministic bottom-up parser for G by modifying the $LL(k)$ parser for $T(G)$ so that it does not emit productions of $T(G)$ but it does emit productions of G in the following way: the production $A \to \alpha$ of G is emitted whenever the production $[A, \alpha] \to \varepsilon$ is recognized by the parser.

The resulting parser for the $PLR(k)$ grammar G is called the *$PLR(k)$ parser* for G. The fact that the $PLR(k)$ parser for G is really a bottom-up parser for G follows from a grammatical covering property of $T(G)$.

For a formal definition of the covering property let $G_1 = (N_1, T, P_1, S_1)$ and $G_2 = (N_2, T, P_2, S_2)$ be grammars such that $L(G_1) = L(G_2)$, and let h be a homomorphism from P_2 to P_1. Following Nijholt [23] we say that G_2 *left-to-right covers* G_1 with respect to the homomorphism h

(i) if π is a left parse of a terminal string w from S_2 in the grammar G^2 then $h(\pi)$ is a right parse of w to S_1 in G_1, and

(ii) if π is a right parse of a terminal string w to S_1 in G_1 then there exists a left parse π' of w from S_2 in G_2 and $h(\pi') = \pi$.

To show that the transformed grammar $T(G)$ left-to-right covers G we first define a homomorphism h_T from the set of productions of $T(G)$ to the set of productions of G. We set $h_T([A, \alpha] \to \varepsilon) = (A \to \alpha)$ for each $[A, \alpha] \to \varepsilon$ where $[A, \alpha] \neq [S', \perp S]$, and $h_T([A, \alpha] \to \eta) = \varepsilon$ otherwise. Now it can be concluded in a way similar to that used to prove Lemma 4.3 that $T(G)$ left-to-right covers G with respect to h_T. In fact, augmenting this proof in the obvious way we can conclude that if $A \underset{R}{\Rightarrow} \underline{X}\alpha \underset{R}{\Rightarrow}{}^* \underline{X}x$ then for any right parse $\pi(A \to X\alpha)$ in G', where π is a right parse of x to α, there exists in $T(G)$ a left parse π_T of x from $[A, X]$ such that $h_T(\pi_T) = \pi(A \to X\alpha)$ if $(A \to X\alpha) \neq (S' \to \perp S)$ and $h_T(\pi_T) = \pi$ otherwise. The correspondence holds also conversely as is easily seen. Consequently, when $A = S'$ and $X = \perp$ this property and Theorem 4.1 yield the following theorem, which implies the correctness of $PLR(k)$ parsing.

Theorem 4.4. *The transformed grammar $T(G)$ left-to-right covers the given grammar G with respect to the homomorphism h_T.*

Example 4.3. To illustrate the $PLR(k)$ parsing, we consider the grammar given in Example 4.2:

$$\begin{aligned}
&S \to i \leftarrow A \\
&S \to i \leftarrow B \\
&A \to A * P \\
&A \to P \\
&B \to A = A \\
&P \to (A) \\
&P \to i
\end{aligned}$$

For instance, the left parse of the string $i \leftarrow i * i$ from $[S', \perp]$ in the transformed grammar (see Example 4.2) comprises the following productions listed in order from left to right:

$$[S', \perp] \to i[S', \perp i], \quad [S', \perp i] \to [S, i][S', \perp S], \quad [S, i] \to \leftarrow [S, i\leftarrow],$$
$$[S, i\leftarrow] \to i[S, i\leftarrow i], \quad [S, i\leftarrow i] \to [P, i][S, i\leftarrow P], \quad [P, i] \to \varepsilon,$$
$$[S, i\leftarrow P] \to [A, P][S, i\leftarrow A], \quad [A, P] \to \varepsilon, \quad [S, i\leftarrow A] \to [A, A][S, i\leftarrow A],$$
$$[A, A] \to *[A, A*], \quad [A, A*] \to i[A, A*i], \quad [A, A*i] \to [P, i][A, A*P],$$
$$[P, i] \to \varepsilon, \quad [A, A*P] \to \varepsilon, \quad [S, i\leftarrow A] \to \varepsilon, \quad [S', \perp S] \to \varepsilon.$$

Thus, the $PLR(1)$ parser for the given grammar in the case of the string $i\leftarrow i*i$ produces the sequence

$$(P \to i)(A \to P)(P \to i)(A \to A * P)(S \to i \leftarrow A).$$

This sequence of productions is, as desired, the right parse of $i\leftarrow i*i$ to the start symbol S in the given grammar. □

5. Conclusion

In this paper we described and studied the properties of a new method for transforming non-$LL(k)$ grammars into $LL(k)$ form. In particular we characterized the applicability of the transformation by defining a class of grammars, called $PLR(k)$ grammars, with the property that a given grammar is $PLR(k)$ if and only if the transformed grammar is $LL(k)$. At present there exists only one other transformation for which a similar characterization has been given ($LC(k)$ grammars by Rosenkrantz and Lewis [25]).

Inspired by the definition of $PLR(k)$ grammars, which seemed to form only a slight generalization of $LC(k)$ grammars, we also suggested a new definition of $LC(k)$ grammars. Using this new definition we showed that $PLR(k)$ grammars are exactly those that can be transformed into $LC(k)$ grammars by left-factoring.

Acknowledgements. We wish to thank Dr. Matti Linna, Prof. Arto Salomaa and Prof. Martti Tienari for their comments. This work was supported by a research project investigating translator writing systems. The project is led by Prof. Martti Tienari and sponsored by the Academy of Finland. The financial support of the Wihuri Foundation is also acknowledged.

References

1. Aho, A.V., Ullman, J.D.: The Theory of Parsing, Translation, and Compiling, Vol. I: Parsing. Englewood Cliffs, N.J.: Prentice-Hall 1972
2. Aho, A.V., Ullman, J.D.: The Theory of Parsing, Translation, and Compiling, Vol. II: Compiling. Englewood Cliffs, N.J.: Prentice-Hall 1973
3. Brosgol, B.M.: Deterministic translation grammars. Harvard University, Cambridge, Mass., Ph.D. Thesis, 1974
4. Burge, W.H.: Recursive Programming Techniques. Reading, Mass.: Addison-Wesley 1975
5. Cheatham, T.E., Jr.: The Theory and Construction of Compilers. Wakefield, Mass.: Computer Associates 1967
6. Demers, A.: Generalized left corner parsing. Conf. Record of the Fourth ACM Symposium on Principles of Programming Languages, pp. 170–182 (1977)
7. Foster, J.M.: A syntax improving program. Comput. J. **11**, 31–34 (1968)

8. Greibach, S.A.: A new normal-form theorem for context-free phrase structure grammars. J. Assoc. Comput. Mach. **12**:1, 42–52 (1965)
9. Griffiths, M.: $LL(1)$ grammars and analysers. In: Compiler Construction: An Advanced Course, Lecture Notes in Computer Science, Vol. 21 (F.L. Bauer, J. Eickel, eds.), pp. 57–84. Berlin-Heidelberg-New York: Springer 1974
10. Griffiths, T.V., Petrick, S.R.: On the relative efficiencies of context-free grammar recognizers. Comm. ACM. **8**, 289–300 (1965)
11. Griffiths, T.V., Petrick, S.R.: Top-down versus bottom-up analysis. In: Proc. of the IFIP congress 68, Vol. 1 (A.J.H. Morrell, ed.), pp. 80–85. Amsterdam: North-Holland 1969
12. Hammer, M.: A new grammatical transformation into deterministic top-down form. MIT, Mass., Project MAC Technical Report TR-119, 1974
13. Hammer, M.: A new grammatical transformation into $LL(k)$ form. Proc. of Sixth Annual ACM Symposium on Theory of Computing, pp. 266–275 (1974)
14. Hopgood, F.R.A.: Compiling Techniques. New York, N.Y.: American Elsevier 1969
15. Irons, E.T.: A syntax directed compiler for ALGOL 60. Comm. ACM. **4**, 51–55 (1961)
16. Knuth, D.E.: On the translation of languages from left to right. Information and Control **8**, 607–639 (1965)
17. Knuth, D.E.: Top-down syntax analysis. Acta Informat. **1**, 79–110 (1971)
18. Komor, T.: A note on left factored languages. Comput. J. **17**, 242–244 (1974)
19. Kurki-Suonio, R.: On top-to-bottom recognition and left recursion. Comm. ACM. **9**, 527–528 (1966)
20. Kurki-Suonio, R.: Notes on top-down languages. BIT **9**, 225–238 (1969)
21. Lewis, P.M., II, Rosenkrantz, D.J., Stearns, R.E.: Compiler Design Theory. Reading, Mass.: Addison-Wesley 1976
22. Lewis, P.M., II, Stearns, R.E.: Syntax directed transduction. J. Assoc. Comput. Mach. **15**, 464–488 (1968)
23. Nijholt, A.: On the covering of left-recursive grammars. Conf. Record of the Fourth ACM Symposium on Principles of Programming Languages, pp. 86–96 (1977)
24. Rosenkrantz, D.J.: Matrix equations and normal forms for context-free grammars. J. Assoc. Comput. Mach. **14**, 501–507 (1967)
25. Rosenkrantz, D.J., Lewis, P.M., II: Deterministic left corner parsing. IEEE Conf. Record of 11th Annual Symposium on Switching and Automata Theory, pp. 139–152 (1970)
26. Rosenkrantz, D.J., Stearns, R.E.: Properties of deterministic top-down grammars. Information and Control **17**, 226–256 (1970)
27. Soisalon-Soininen, E.: Characterization of $LL(k)$ languages by restricted $LR(k)$ grammars. Department of Computer Science, University of Helsinki, Helsinki, Report A-1977-3, 1977
28. Soisalon-Soininen, E., Ukkonen, E.: A characterization of $LL(k)$ languages. In: Automata, Languages and Programming, Third Colloquium (S. Michaelson, R. Milner, eds.), pp. 20–30. Edinburgh: Edinburgh University Press 1976
29. Stearns, R.E.: Deterministic top-down parsing. Proc. Fifth Annual Princeton Conf. on Information Sciences and Systems, pp. 182–188 (1971)
30. Wood, D.: The normal form theorem – another proof. Comput. J. **12**, 139–147 (1969)

Received January 20, 1978 / June 21, 1979