Pregroup Grammars with Letter Promotions

Wojciech Buszkowski[1,2] and Zhe Lin[1,3]

[1] Adam Mickiewicz University in Poznań, Poland
[2] University of Warmia and Mazury in Olsztyn, Poland
[3] Sun Yat-sen University in Guangzhou, China
buszko@amu.edu.pl pennyshaq@gmail.com

Abstract. We study pregroup grammars with letter promotions $p^{(m)} \Rightarrow q^{(n)}$. We show that the Letter Promotion Problem for pregroups is solvable in polynomial time, if the size of $p^{(n)}$ is counted as $|n|+1$. In Mater and Fix [11], the problem is shown to be NP-hard, but their proof assumes the binary (or decimal, etc.) representation of n in $p^{(n)}$, which seems less natural for applications. We reduce the problem to a graph-theoretic problem, which is subsequently reduced to the emptiness problem for context-free languages. As a consequence, the following problems are in P: the word problem for pregroups with letter promotions and the membership problem for pregroup grammars with letter promotions.

1 Introduction and Preliminaries

Pregroups, introduced in Lambek [8], are ordered algebras $(M, \leq, \cdot, l, r, 1)$ such that $(M, \leq, \cdot, 1)$ is a partially ordered monoid (hence $\cdot$ is monotone in both arguments), and l, r are unary operations on M, fulfilling the following conditions:

$$a^l a \leq 1 \leq a a^l \,,\ a a^r \leq 1 \leq a^r a \,, \tag{1}$$

for all $a \in M$. The operation $\cdot$ is referred to as *product*. The element a^l (resp. a^r) is called the *left* (resp. *right*) *adjoint* of a.

The following laws are valid in pregroups:

$$1^l = 1 = 1^r \,, \tag{2}$$

$$(a^l)^r = a = (a^r)^l \,, \tag{3}$$

$$(ab)^l = b^l a^l \,,\ (ab)^r = b^r a^r \,, \tag{4}$$

$$a \leq b \text{ iff } b^l \leq a^l \text{ iff } b^r \leq a^l \,. \tag{5}$$

In any pregroup, one defines $a\backslash b = a^r b$, $a/b = ab^l$, and proves that $\cdot, \backslash, /$ satisfy the residuation law:

$$ab \leq c \text{ iff } b \leq a\backslash c \text{ iff } a \leq c/b, \tag{6}$$

for all elements a, b, c. Consequently, pregroups are a special class of residuated monoids, i.e. models of the Lambek calculus **L*** [3, 2].

Lambek [8] (also see [9, 10]) offers (free) pregroups as a computational machinery for lexical grammars, alternative to the Lambek calculus. The latter is widely recognized as a basic logic of categorial grammars [17, 2]; linguists usually employ the system **L** of the Lambek calculus, which is complete with respect to residuated semigroups (it is weaker than **L***).

The logic of pregroups is called Compact Bilinear Logic (**CBL**). It arises from Bilinear Logic (Noncommutative **MLL**) by collapsing 'times' and 'par', whence also 0 and 1. **CBL** is stronger than **L***; for instance, $(p/((q/q)/p))/p \leq p$ is valid in pregroups but not in residuated monoids [3], whence it is provable in **CBL**, but not in **L***. By the same example, **CBL** is stronger than Bilinear Logic, since the latter is a conservative extension of **L***.

Let M be a pregroup. For $a \in M$, one defines $a^{(n)}$ as follows: $a^{(0)} = a$; if n is negative, then $a^{(n)} = a^{l \cdots l}$ (l is iterated $|n|$ times); if n is positive, then $a^{(n)} = a^{r \cdots r}$ (r is iterated n times). The following laws can easily be proved:

$$(a^{(n)})^l = a^{(n-1)}\,,\ (a^{(n)})^r = a^{(n+1)}\,,\ \text{for all } n \in \mathbf{Z}\,, \tag{7}$$

$$a^{(n)}a^{(n+1)} \leq 1 \leq a^{(n+1)}a^{(n)}\,,\ \text{for all } n \in \mathbf{Z}\,, \tag{8}$$

$$(a^{(m)})^{(n)} = a^{(m+n)}\,,\ \text{for all } m, n \in \mathbf{Z}\,, \tag{9}$$

$$a \leq b \text{ iff } a^{(n)} \leq b^{(n)}\,,\ \text{for all even } n \in \mathbf{Z}\,, \tag{10}$$

$$a \leq b \text{ iff } b^{(n)} \leq a^{(n)}\,,\ \text{for all odd } n \in \mathbf{Z}\,, \tag{11}$$

where **Z** denotes the set of integers.

CBL can be formalized as follows. Let $(P, \leq)$ be a nonempty finite poset. Elements of P are called *atoms*. *Terms* are expressions of the form $p^{(n)}$ such that $p \in P$ and n is an integer. One writes p for $p^{(0)}$. *Types* are finite strings of terms. Terms are denoted by t, u and types by X, Y, Z. The relation $\Rightarrow$ on the set of types is defined by the following rules:

(CON) $X, p^{(n)}, p^{(n+1)}, Y \Rightarrow X, Y$,
(EXP) $X, Y \Rightarrow X, p^{(n+1)}, p^{(n)}, Y$,
(POS) $X, p^{(n)}, Y \Rightarrow X, q^{(n)}, Y$, if $p \leq q$, for even n, and $q \leq p$, for odd n,

called Contraction, Expansion, and Poset rules, respectively (the latter are called Induced Steps in Lambek [8]). To be precise, $\Rightarrow$ is the reflexive and transitive closure of the relation defined by these rules. The pure **CBL** is based on a trivial poset $(P, =)$.

An *assignment* in a pregroup M is a mapping $\mu : P \mapsto M$ such that $\mu(p) \leq \mu(q)$ in M whenever $p \leq q$ in $(P, \leq)$. Clearly any assignment μ is uniquely extendible to a homomorphism of the set of types into M; one sets $\mu(\epsilon) = 1$, $\mu(p^{(n)}) = (\mu(p))^{(n)}$, $\mu(XY) = \mu(X)\mu(Y)$. The following completeness theorem is true: $X \Rightarrow Y$ holds in **CBL** if and only if, for any pregroup M and any assignment μ of P in M, $\mu(X) \leq \mu(Y)$ [3].

A *pregroup grammar* assigns a finite set of types to each word from a finite lexicon Σ. Then, a nonempty string $v_1 \ldots v_n$ ($v_i \in \Sigma$) is assigned type X, if there exist types $X_1, \ldots, X_n$ initially assigned to words $v_1, \ldots, v_n$, respectively, such that $X_1, \ldots, X_n \Rightarrow X$ in **CBL**. For instance, if 'goes' is assigned type $\pi_3^{(1)} s_1$ and 'he' type π_3, then 'he goes' is assigned type s_1 (statement in the present tense). π_k represents the $k-$th person pronoun. For the past tense, the person is irrelevant; so, π represents pronoun (any person), and one assumes $\pi_k \leq \pi$, for $k = 1, 2, 3$. Now, if 'went' is assigned type $\pi^{(1)} s_2$, then 'he went' is assigned type s_2 (statement in the past tense), and similarly for 'I went', 'you went'. Assuming $s_i \leq s$, for $i = 1, 2$, one can assign type s (statement) to all sentences listed above. These examples come from [8].

A pregroup grammar is formally defined as a quintuple $G = (\Sigma, P, I, s, R)$ such that Σ is a finite alphabet (lexicon), P is a finite set (of atoms), s is a designated atom (the principal type), I is a finite relation between elements of Σ and types on P, and R is a partial ordering on P. One writes $p \leq q$ for pRq, if R is fixed. *The language* of G, denoted $L(G)$, consists of all strings $x \in \Sigma^+$ such that G assigns type s to x (see the above paragraph). Pregroup grammars are weakly equivalent to $\epsilon-$free context-free grammars [3]; hence, the former provide a lexicalization of the latter.

As shown in [1], every pregroup grammar can be fully lexicalized; there exists a polynomial time transformation which sends any pregroup grammar to an equivalent pregroup grammar on a trivial poset $(P, =)$. Actually, an exponential time procedure is quite obvious: it suffices to apply all possible (POS)-transitions to the lexical types in I [5].

Lambek [8] proves a normalization theorem for **CBL** (also called: Lambek Switching Lemma). One introduces new rules:

(GCON) $X, p^{(n)}, q^{(n+1)}, Y \Rightarrow X, Y$,
(GEXP) $X, Y \Rightarrow X, p^{(n+1)}, q^{(n)}, Y$,

if either n is even and $p \leq q$, or n is odd and $q \leq p$. These rules are called Generalized Contraction and Generalized Expansion, respectively. Clearly they are derivable in **CBL**: (GCON) amounts to (POS) followed by (CON), and (GEXP) amounts to (EXP) followed by (POS). Lambek's normalization theorem states: if $X \Rightarrow Y$ in **CBL**, then there exist types Z, U such that $X \Rightarrow Z$, by a finite number of instances of (GCON), $Z \Rightarrow U$, by a finite number of instances of (POS), and $U \Rightarrow Y$, by a finite number of instances of (GEXP). Consequently, if Y is a term or $Y = \epsilon$, then $X \Rightarrow Y$ in **CBL** if and only if X can be reduced to Y without (GEXP) (hence, by (CON) and (POS) only). The normalization theorem is equivalent to the cut-elimination theorem for a sequent system of **CBL** [4].

This yields the polynomial time complexity of the provability problem for **CBL** [3, 4]. For any type X, define X^l and X^r as follows:

$$\epsilon^l = \epsilon^r = \epsilon\,,\ (t_1 t_2 \cdots t_k)^\alpha = (t_k)^\alpha \cdots (t_2)^\alpha (t_1)^\alpha\,, \tag{12}$$

for $\alpha \in \{l, r\}$, where t^α is defined according to (7): $(p^{(n)})^l = p^{(n-1)}$, $(p^{(n)})^r = p^{(n+1)}$. In **CBL** the following equivalences hold:

$$X \Rightarrow Y \text{ iff } X, Y^r \Rightarrow \epsilon \text{ iff } Y^l, X \Rightarrow \epsilon\,, \tag{13}$$

for all types X, Y. We prove the first equivalence. Assume $X \Rightarrow Y$. Then, $X, Y^r \Rightarrow Y, Y^r \Rightarrow \epsilon$, by an obvious congruence property of $\Rightarrow$ and a finite number of (CON). Assume $X, Y^r \Rightarrow \epsilon$. Then, $X \Rightarrow X, Y^r, Y \Rightarrow Y$, by a finite number of (EXP) and a congruence property of $\Rightarrow$. In a similar way, one proves: $X \Rightarrow Y$ iff $Y^l, X \Rightarrow \epsilon$.

In order to verify whether $X \Rightarrow Y$ in **CBL** one verifies whether $X, Y^r \Rightarrow \epsilon$; the latter holds if and only if XY^r can be reduced to ϵ by a finite number of instances of (GCON). An easy modification of the CYK-algorithm for context-free grammars yields a polynomial time algorithm, solving this problem (also see [12]). Furthermore, every pregroup grammar can be transformed into an equivalent context-free grammar in polynomial time [3, 5]. Francez and Kaminsky [6] show that pregroup grammars augmented with partial commutation can generate some non-context-free languages.

We have formalized **CBL** with special assumptions. Assumptions $p \leq q$ in nontrivial posets express different forms of subtyping, as shown in the above examples.

It is interesting to consider **CBL** enriched with more general assumptions. Mater and Fix [11] show that **CBL** enriched with finitely many assumptions of the general form $X \Rightarrow Y$ can be undecidable (the word problem for groups is reducible to systems of that kind). For assumptions of the form $t \Rightarrow u$ (called *letter promotions*) they prove a weaker form of Lambek's normalization theorem for the resulting calculus (for sequents $X \Rightarrow \epsilon$ only).

A complete system of **CBL** with letter promotions is obtained by modifying (POS) to the following Promotion Rules:

(PRO) $X, p^{(m+k)}, Y \Rightarrow X, q^{(n+k)}, Y$, if either k is even and $p^{(m)} \Rightarrow q^{(n)}$ is an assumption, or k is odd and $q^{(n)} \Rightarrow p^{(m)}$ is an assumption.

The Letter Promotion Problem for pregroups (LPPP) is the following: given a finite set R, of letter promotions, and terms t, u, verify whether $t \Rightarrow u$ in **CBL** enriched with all promotions from R as assumptions.

To formulate the problem quite precisely, we need some formal notions. Let R denote a finite set of letter promotions. We write $R \vdash_{\mathbf{CBL}} X \Rightarrow Y$, if X can be transformed into Y, using finitely many instances of (CON), (EXP) and (PRO), restricted to the assumptions from R. Now, the problem under consideration amounts to verifying whether $R \vdash_{\mathbf{CBL}} t \Rightarrow u$, for given R, t, u.

Since the formalism is based on no fixed poset, we have to explain what are atoms (atomic types). We fix a denumerable set P of atoms. Terms and types are defined as above. $P(R)$ denotes the set of atoms appearing in assumptions from R. By an assignment in M we mean now a mapping $\mu : P \mapsto M$. We prove a standard completeness theorem.

Theorem 1. *$R \vdash_{\mathbf{CBL}} X \Rightarrow Y$ if, and only if, for any pregroup M and any assignment μ in M, if all assumptions from R are true in (M, μ), then $X \Rightarrow Y$ is true in (M, μ).*

Proof. The 'only if' part is easy. For the 'if' part one constructs a special pregroup M whose elements are equivalence classes of the relation: $X \sim Y$ iff

$R \vdash_{CBL} X \Rightarrow Y$ and $R \vdash_{CBL} Y \Rightarrow X$. One defines: $[X] \cdot [Y] = [XY]$, $[X]^\alpha = [X^\alpha]$, for $\alpha \in \{l, r\}$, $[X] \leq [Y]$ iff $R \vdash_{CBL} X \Rightarrow Y$. For $\mu(p) = [p]$, $p \in P$, one proves: $X \Rightarrow Y$ is true in (M, μ) iff $R \vdash_{CBL} X \Rightarrow Y$. □

Mater and Fix [11] claim that LPPP is NP-complete. Actually, their paper only provides a proof of NP-hardness; even the decidability of LPPP does not follow from their results.

The NP-hardness is proved by a reduction of the following Subset Sum Problem to LPPP: given a nonempty finite set of integers $S = \{k_1, \ldots, k_m\}$ and an integer k, verify whether there exists a subset $X \subseteq S$ such that the sum of all integers from X equals k. The latter problem is NP-complete, if integers are represented in a binary (or decimal, etc.) code; see [7]. For the reduction, one considers $m + 1$ atoms $p_0, \ldots, p_m$ and the promotions R: $p_{i-1} \Rightarrow p_i$, for all $i = 1, \ldots, m$, and $p_{i-1} \Rightarrow (p_i)^{(2k_i)}$, for all $i = 1, \ldots, m$. Then, the Subset Sum Problem has a solution if and only if $p_0 \Rightarrow (p_m)^{(2k)}$ is derivable from R. Clearly the reduction assumes the binary representation of n in $p^{(n)}$.

In linguistic applications, it is more likely that R contains many promotions $p^{(m)} \Rightarrow q^{(n)}$, but all integers in them are relatively small. In Lambek's original setting, these integers are equal to 0. It is known that in pregroups: $a \leq a^{ll}$ iff a is surjective (i.e. $ax = b$ has a solution, for any b), and $a^{ll} \leq a$ iff a is injective (i.e. $ax = ay$ implies $x = y$) [3]. One can postulate these properties by promotions: $p \Rightarrow p^{(-2)}$, $p^{(-2)} \Rightarrow p$. Let n be the atomic type of negation 'not', then $nn \Rightarrow \epsilon$ expresses the double negation law on the syntactic level, and this promotion is equivalent to $n \Rightarrow n^{(-1)}$. All linguistic examples in [8, 10] use at most three (usually, one or two) iterated left or right adjoints. Accordingly, binary encoding is not very useful for such applications.

It seems more natural to look at $p^{(n)}$ as an abbreviated notation for $p^{l \cdots l}$ or $p^{r \cdots r}$, where adjoints are iterated $|n|$ times, and take $|n| + 1$ as the proper complexity measure of this term. Under this proviso, we prove below that LPPP is polynomial time decidable. As a consequence, the provability problem for **CBL** with letter promotions has the same complexity. Accordingly, we prove the decidability of both problems, and the polynomial time complexity of them (under the proviso). (The final comments of [11] suggest that a practically useful version of LPPP may have a lower complexity.)

Oehrle [15] and Moroz [14] provide some cubic parsing algorithms for pregroup grammars (the former uses some graph-theoretic ideas; the latter modifies Savateev's algorithm for the unidirectional Lambek grammars [16]). These algorithms can be adjusted for pregroup grammars with letter promotions. Pregroup grammars with (finitely many) letter promotions are weakly equivalent to $\epsilon-$free context-free grammars. We do not elaborate these matters here, since they are rather routine variants of results obtained elsewhere; also see [3, 5, 14].

2 The Normalization Theorem

We provide a full proof of the Lambek-style normalization theorem for **CBL** with letter promotions, which yields a simpler formulation of LPPP.

We write $t \Rightarrow_R u$, if $t \Rightarrow u$ is an instance of (PRO), restricted to the assumptions from R (X, Y are empty). We write $t \Rightarrow_R^* u$, if there exist terms $t_0, \ldots, t_k$ such that $k \geq 0$, $t_0 = t$, $t_k = u$, and $t_{i-1} \Rightarrow_R t_i$, for all $i = 1, \ldots, k$. Hence $\Rightarrow_R^*$ is the reflexive and transitive closure of $\Rightarrow_R$.

It is expedient to introduce derivable rules of Generalized Contraction and Generalized Expansion for **CBL** with letter promotions.

(GCON$-R$) $X, p^{(m)}, q^{(n+1)}, Y \Rightarrow X, Y$, if $p^{(m)} \Rightarrow_R^* q^{(n)}$,
(GEXP$-R$) $X, Y \Rightarrow X, p^{(n+1)}, q^{(m)}, Y$, if $p^{(n)} \Rightarrow_R^* q^{(m)}$.

These rules are derivable in **CBL** with assumptions from R, and (CON), (EXP) are special instances of them. We also treat any iteration of (PRO)-steps as a single step:

(PRO$-R$) $X, t, Y \Rightarrow X, u, Y$, if $t \Rightarrow_R^* u$.

The following normalization theorem has been proved in [11], for the particular case $Y = \epsilon$: if $X \Rightarrow \epsilon$ is provable, then X reduces to ϵ by (GCON$-R$) only. This easily follows from Theorem 2 and does not directly imply Lemma 1. Here we prove the full version (this result is essential for further considerations).

Theorem 2. *If $R \vdash_{\mathbf{CBL}} X \Rightarrow Y$, then there exist Z, U such that $X \Rightarrow Z$ by a finite number of instances of (GCON$-R$), $Z \Rightarrow U$ by a finite number of instances of (PRO$-R$), and $U \Rightarrow Y$ by a finite number of instances of (GEXP$-R$).*

Proof. By *a derivation* of $X \Rightarrow Y$ in **CBL** from the set of assumptions R, we mean a sequence $X_0, \ldots, X_k$ such that $X = X_0$, $Y = X_k$ and, for any $i = 1, \ldots, k$, $X_{i-1} \Rightarrow X_i$ is an instance of (GCON$-R$), (GEXP$-R$) or (PRO$-R$); k is *the length* of this derivation. We show that every derivation $X_0, \ldots, X_k$ of $X \Rightarrow Y$ in **CBL** from R can be transformed into a derivation of the required form (a normal derivation) whose length is at most k. We proceed by induction on k.

For $k = 0$ and $k = 1$ the initial derivation is normal; for $k = 0$, one takes $X = Z = U = Y$, and for $k = 1$, if $X \Rightarrow Y$ is an instance of (GCON$-R$), one takes $Z = U = Y$, if $X \Rightarrow Y$ is an instance of (GEXP$-R$), one takes $X = Z = U$, and if $X \Rightarrow Y$ is an instance of (PRO$-R$), one takes $X = Z$ and $U = Y$.

Assume $k > 1$. The derivation $X_1, \ldots, X_k$ is shorter, whence it can be transformed into a normal derivation $Y_1, \ldots, Y_l$ such that $X_1 = Y_1$, $X_k = Y_l$ and $l \leq k$. If $l < k$, then $X_0, Y_1, \ldots, Y_l$ is a derivation of $X \Rightarrow Y$ of length less than k, whence it can be transformed into a normal derivation, by the induction hypothesis. So assume $l = k$.

Case 1. $X_0 \Rightarrow X_1$ is an instance of (GCON$-R$). Then $X_0, Y_1, \ldots, Y_l$ is a normal derivation of $X \Rightarrow Y$ from R.

Case 2. $X_0 \Rightarrow X_1$ is an instance of (GEXP$-R$), say $X_0 = UV$, $X_1 = Up^{(n+1)}q^{(m)}V$, and $p^{(n)} \Rightarrow_R^* q^{(m)}$. We consider two subcases.

Case 2.1 No (GCON$-R$)-step of $Y_1, \ldots, Y_l$ acts on the designated occurrences of $p^{(n+1)}, q^{(m)}$. If also no (PRO$-R$)-step of $Y_1, \ldots, Y_l$ acts on these designated terms, then we drop $p^{(n+1)}q^{(m)}$ from all types appearing in (GCON$-R$)-

steps and (PRO$-R$)-steps of $Y_1,\ldots,Y_l$, then introduce them by a single instance of (GEXP$-R$), and continue the (GEXP$-R$)-steps of $Y_1,\ldots,Y_l$; this yields a normal derivation of $X \Rightarrow Y$ of length k. Otherwise, let $Y_{i-1} \Rightarrow Y_i$ be the first (PRO$-R$)-step of $Y_1,\ldots,Y_l$ which acts on $p^{(n+1)}$ or $q^{(m)}$. If it acts on $p^{(n+1)}$, then there exist a term $r^{(m')}$ and types T,W such that $Y_{i-1} = Tp^{(n+1)}W$, $Y_i = Tr^{(m')}W$ and $p^{(n+1)} \Rightarrow_R^* r^{(m')}$. Then, $r^{(m'-1)} \Rightarrow_R^* p^{(n)}$, whence $r^{(m'-1)} \Rightarrow_R^* q^{(m)}$, and we can replace the derivation $X_0,Y_1,\ldots,Y_l$ by a shorter derivation: first apply (GEXP$-R$) of the form $U,V \Rightarrow U,r^{(m')},q^{(m)},V$, then derive $Y_1,\ldots,Y_{i-1}$ in which $p^{(n+1)}$ is replaced by $r^{(m')}$, drop Y_i, and continue $Y_{i+1},\ldots,Y_l$. By the induction hypothesis, this derivation can be transformed into a normal derivation of length less than k. If $Y_{i-1} \Rightarrow Y_i$ acts on $q^{(m)}$, then there exist a term $r^{(m')}$ and types T,W such that $Y_{i-1} = Tq^{(m)}W$, $Y_i = Tr^{(m')}W$ and $q^{(m)} \Rightarrow_R^* r^{(m')}$. Then, $p^{(n)} \Rightarrow_R^* r^{(m')}$, and we can replace the derivation $X_0,Y_1,\ldots,Y_l$ by a shorter derivation: first apply (GEXP$-R$) of the form $U,V \Rightarrow U,p^{(n+1)},r^{(m')},V$, then derive $Y_1,\ldots,Y_{i-1}$ in which $q^{(m)}$ is replaced by $r^{(m')}$, drop Y_i, and continue $Y_{i+1},\ldots,Y_l$. Again we apply the induction hypothesis.

Case 2.2. Some (GCON$-R$)-step of $Y_1,\ldots,Y_l$ acts on (some of) the designated occurrences of $p^{(n+1)},q^{(m)}$. Let $Y_{i-1} \Rightarrow Y_i$ be the first step of that kind. There are three possibilities. (I) This step acts on both $p^{(n+1)}$ and $q^{(m)}$. Then, the derivation $X_0,Y_1,\ldots,Y_l$ can be replaced by a shorter derivation: drop the first application of (GEXP$-R$), then derive $Y_1,\ldots,Y_{i-1}$ in which $p^{(n+1)}q^{(m)}$ is omitted, drop Y_i, and continue $Y_{i+1},\ldots,Y_l$. We apply the induction hypothesis. (II) This step acts on $p^{(n+1)}$ only. Then, $Y_{i-1} = Tr^{(m')}p^{(n+1)}q^{(m)}W$, $Y_i = T,q^{(m)},W$ and $r^{(m')} \Rightarrow_R^* p^{(n)}$. The derivation $X_0,Y_1,\ldots,Y_l$ can be replaced by a shorter derivation: drop the first application of (GEXP$-R$), then derive $Y_1,\ldots,Y_{i-1}$ in which $p^{(n+1)}q^{(m)}$ is omitted, derive Y_i by a (PRO$-R$)-step (notice $r^{(m')} \Rightarrow_R^* q^{(m)}$), and continue $Y_{i+1},\ldots,Y_l$. We apply the induction hypothesis. (III) This step acts on $q^{(m)}$ only. Then, $Y_{i-1} = Tp^{(n+1)}q^{(m)}r^{(m'+1)}W$, $Y_i = Tp^{(n+1)}W$ and $q^{(m)} \Rightarrow_R^* r^{(m')}$. The derivation $X_0,Y_1,\ldots,Y_l$ can be replaced by a shorter derivation: drop the first application of (GEXP$-R$), then derive $Y_1,\ldots,Y_{i-1}$ in which $p^{(n+1)}q^{(m)}$ is dropped, derive Y_i by a (PRO$-R$)-step (notice $r^{(m'+1)} \Rightarrow_R^* p^{(n+1)}$), and continue $Y_{i+1},\ldots,Y_l$. We apply the induction hypothesis.

Case 3. $X_0 \Rightarrow X_1$ is an instance of (PRO$-R$), say $X_0 = UtV$, $X_1 = UuV$ and $t \Rightarrow_R^* u$. We consider two subcases.

Case 3.1. No (GCON$-R$)-step of $Y_1,\ldots,Y_l$ acts on the designated occurrence of u. Then $X_0,Y_1,\ldots,Y_l$ can be transformed into a normal derivation of length k: drop the first application of (PRO$-R$), apply all (GCON$-R$)-steps of $Y_1,\ldots,Y_l$ in which the designated occurrences of u are replaced by t, apply a (PRO$-R$)-step which changes t into u, and continue the remaining steps of $Y_1,\ldots,Y_l$.

Case 3.2. Some (GCON$-R$)-step of $Y_1,\ldots,Y_l$ acts on the designated occurrence of u. Let $Y_{i-1} \Rightarrow Y_i$ be the first step of that kind. There are two possibilities. (I) $Y_{i-1} = Tuq^{(n+1)}W$, $Y_i = TW$ and $u \Rightarrow_R^* q^{(n)}$. Since $t \Rightarrow_R^* q^{(n)}$, then $X, Y_1, \dots, Y_l$ can be transformed into a shorter derivation: drop the first application of (PRO$-R$), derive $Y_1, \dots, Y_{i-1}$ in which the designated occurrences of u are replaced by t, derive Y_i by a (GCON$-R$)-step of the form $T, t, q^{(n+1)}, W \Rightarrow T, W$, and continue $Y_{i+1}, \dots, Y_l$. We apply the induction hypothesis. (II) $u = q^{(n+1)}$, $Y_{i-1} = Tp^{(m)}uW$, $Y_i = TW$ and $p^{(m)} \Rightarrow_R^* q^{(n)}$. Let $t = r^{(n')}$. We have $q^{(n)} \Rightarrow_R^* r^{(n'-1)}$, whence $p^{(m)} \Rightarrow_R^* r^{(n'-1)}$. The derivation $X_0, Y_1, \dots, Y_l$ can be transformed into a shorter derivation: drop the first aplication of (PRO$-R$), derive $Y_1, \dots, Y_{i-1}$ in which the designated occurrences of u are replaced by t, derive Y_i by a (GCON$-R$)-step of the form $T, p^{(m)}, r^{(n')}, W \Rightarrow T, W$, and continue $Y_{i+1}, \dots, Y_l$. We apply the induction hypothesis. □

As a consequence, we obtain:

Lemma 1. *$R \vdash_{\mathbf{CBL}} t \Rightarrow u$ if, and only if, $t \Rightarrow_R^* u$.*

Proof. The 'if' part is obvious. The 'only if' part employs Theorem 2. Assume $R \vdash_{\mathbf{CBL}} t \Rightarrow u$. There exists a normal derivation of $t \Rightarrow u$ from R. The first step of this derivation cannot be (GCON$-R$), whence (GCON$-R$) is not applied at all; the last step cannot be (GEXP$-R$), whence (GEXP$-R$) cannot be applied at all. Consequently, each step of the derivation is a (PRO$-R$)-step (with X, Y empty). whence the derivation reduces to a single (PRO$-R$)-step. This yields $t \Rightarrow_R^* u$. □

Accordingly, LPPP amounts to verifying whether $t \Rightarrow_R^* u$, for any given R, t, u.

3 LPPP and Weighted Graphs

We reduce LPPP to a graph-theoretic problem. In the next section, the second problem is reduced to the emptiness problem for context-free languages. Both reductions are polynomial, and the third problem is solvable in polynomial time. This yields the polynomial time complexity of LPPP.

We define a finite weighted directed graph $G(R)$. $P(R)$ denotes the set of atoms occurring in promotions from R. The vertices of $G(R)$ are elements p_0, p_1, for all $p \in P(R)$. For any integer n, we set $\pi(n) = 0$, if n is even, and $\pi(n) = 1$, if n is odd. We also set $\pi^*(n) = 1 - \pi(n)$. For any promotion $p^{(m)} \Rightarrow q^{(n)}$ from R, $G(R)$ contains an arc from $p_{\pi(m)}$ to $q_{\pi(n)}$ with weight $n - m$ and an arc from $q_{\pi^*(n)}$ to $p_{\pi^*(m)}$ with weight $m - n$. Thus, each promotion from R gives rise to two weighted arcs in $G(R)$.

An arc from v to w of weight k is represented as the triple (v, k, w). As usual, a *route* from a vertex v to a vertex w in $G(R)$ is defined as a sequence of arcs $(v_0, k_1, v_1), \dots, (v_{r-1}, k_r, v_r)$ such that $v_0 = v$, $v_r = w$, and the target of each but the last arc equals the source of the next arc. The *length* of this route is r, and its *weight* is $k_1 + \cdots + k_r$. We admit a trivial route from v to v of length 0 and weight 0.

Lemma 2. *If $p^{(m)} \Rightarrow_R q^{(n)}$, then $(p_{\pi(m)}, n - m, q_{\pi(n)})$ is an arc in $G(R)$.*

Proof. Assume $p^{(m)} \Rightarrow_R q^{(n)}$. We consider two cases.

(I) $m = m' + k$, $n = n' + k$, k is even, and $p^{(m')} \Rightarrow q^{(n')}$ belongs to R. Then $(p_{\pi(m')}, n' - m', q_{\pi(n')})$ is an arc in $G(R)$. We have $\pi(m) = \pi(m')$, $\pi(n) = \pi(n')$ and $n - m = n' - m'$, which yields the thesis.

(II) $m = m' + k$, $n = n' + k$, k is odd, and $q^{(n')} \Rightarrow p^{(m')}$ belongs to R. Then $(p_{\pi^*(m')}, n' - m', q_{\pi^*(n')})$ is an arc in $G(R)$. We have $\pi^*(m') = \pi(m)$, $\pi^*(n') = \pi(n)$ and $n - m = n' - m'$, which yields the thesis. □

Lemma 3. *Let $(v, r, q_{\pi(n)})$ be an arc in $G(R)$. Then, there is some $p \in P(R)$ such that $v = p_{\pi(n-r)}$ and $p^{(n-r)} \Rightarrow_R q^{(n)}$.*

Proof. We consider two cases.

(I) $(v, r, q_{\pi(n)})$ equals the arc $(p_{\pi(m')}, n' - m', q_{\pi(n')})$, and $p^{(m')} \Rightarrow q^{(n')}$ belongs to R. Then $r = n' - m'$ and $\pi(n) = \pi(n')$. We have $n = n' + k$, for an even integer k, whence $n - r = m' + k$. This yields $\pi(n - r) = \pi(m')$ and $p^{(n-r)} \Rightarrow_R q^{(n)}$.

(II) $(v, r, q_{\pi(n)})$ equals $(p_{\pi^*(m')}, n' - m', q_{\pi^*(n')})$, and $q^{(n')} \Rightarrow p^{(m')}$ belongs to R. Then $r = n' - m'$ and $\pi(n) = \pi^*(n')$. We have $n = n' + k$, for an odd integer k, whence $n - r = m' + k$. This yields $\pi(n - r) = \pi^*(m')$ and $p^{(n-r)} \Rightarrow_R q^{(n)}$. □

Theorem 3. *Let $p, q \in P(R)$. Then, $p^{(m)} \Rightarrow_R^* q^{(n)}$ if and only if there exists a route from $p_{\pi(m)}$ to $q_{\pi(n)}$ of weight $n - m$ in $G(R)$.*

Proof. The 'only if' part easily follows from Lemma 2. The 'if' part is proved by induction on the length of a route from $p_{\pi(m)}$ to $q_{\pi(n)}$ in $G(R)$, using Lemma 3. For the trivial route, we have $p = q$ and $n - m = 0$, whence $n = m$; so, the trivial derivation yields $p_{\pi(m)}^{(m)} \Rightarrow_R^* p_{\pi(m)}^{(m)}$. Assume that $(p_{\pi(m)}, r_1, v_1)$, (v_1, r_2, v_2), $\ldots, (v_k, r_{k+1}, q_{\pi(n)})$ is a route of length $k + 1$ and weight $n - m$ in $G(R)$. By Lemma 3, there exists $s \in P$ such that $v_k = s_{\pi(n-r_{k+1})}$ and $s^{(n-r_{k+1})} \Rightarrow_R q^{(n)}$. The weight of the initial subroute of length k is $n - m - r_{k+1}$, which equals $n - r_{k+1} - m$. By the induction hypothesis $p^{(m)} \Rightarrow_R^* s^{(n-r_{k+1})}$, which yields $p^{(m)} \Rightarrow_R^* q^{(n)}$. □

We return to LPPP. To verify whether $R \vdash p^{(m)} \Rightarrow q^{(n)}$ we consider two cases. If $p, q \in P(R)$, then, by Lemma 1 and Theorem 3, the answer is YES iff there exists a route in $G(R)$, as in Theorem 3. Otherwise, $R \vdash p^{(m)} \Rightarrow q^{(n)}$ iff $p = q$ and $m = n$.

4 Main Results

We have reduced LPPP to the following problem: given a finite weighted directed graph G with integer weights, two vertices v, w and an integer k, verify whether there exists a route from v to w of weight k in G. Caution: integers are represented in unary notation, e.g. 5 is the string of five digits.

We present a polynomial time reduction of this problem to the emptiness problem for context-free languages. Since a trivial route exists if and only if $v = w$ and $k = 0$, then we may restrict the problem to nontrivial routes.

First, the graph G is transformed into a non-deterministic FSA $M(G)$ in the following way. The alphabet of $M(G)$ is $\{+,-\}$. We describe the graph of $M(G)$. The states of $M(G)$ are vertices of G and some auxiliary states. If (v',n,w') is an arc in G, $n>0$, then we link v' with w' by n transitions $v' \to s_1 \to s_2 \to \cdots \to s_n = w'$, all labeled by $+$, where $s_1,\ldots,s_{n-1}$ are new states; similarly for $n<0$ except that now the transitions are labeled by $-$. For $n=0$, we link v' with w' by two transitions $v' \to s \to w'$, the first one labeled by $+$, and the second one by $-$, where s is a new state. The final state is w. If $k=0$, then v is the start state. If $k \neq 0$, then we add new states $i_1,\ldots,i_k$ with transitions $i_1 \to i_2 \to \cdots \to i_k$ and $i_k \to v$, all labeled by $-$, if $k>0$, and by $+$, if $k<0$; the start state is i_1. The following equivalence is obvious: there exists a nontrivial route from v to w of weight k in G iff there exists a nontrivial route from the start state to the final state in $M(G)$ which visits as many pluses as minuses.

Let L be the context-free language, consisting of all nonempty strings on $\{+,-\}$ which contain as many pluses as minuses. The right-hand side of the above equivalence is equivalent to $L(M(G)) \cap L \neq \emptyset$.

A CFG for L consists of the following production rules: $S \mapsto SS$, $S \mapsto +S-$, $S \mapsto -S+$, $S \mapsto +-$, $S \mapsto -+$. We transform it to a CFG in the Chomsky Normal Form (i.e. all rules are of the form $A \mapsto BC$ or $A \mapsto a$) in a constant time. The latter is modified to a CFG for $L(M(G)) \cap L$ in a routine way. The new variables are of the form (q,A,q'), where q,q' are arbitrary states of $M(G)$, A is a variable of the former grammar. The initial symbol is (q_0,S,q_f), where q_0 is the start state and q_f the final state of $M(G)$. The new production rules are:

(1) $(q_1,A,q_3) \mapsto (q_1,B,q_2)(q_2,C,q_3)$ for any rule $A \mapsto BC$ of the former grammar,

(2) $(q_1,A,q_2) \mapsto a$, whenever $A \mapsto a$ is a rule of the former grammar, and $M(G)$ admits the transition from q_1 to q_2, labeled by $a \in \{+,-\}$.

The size of a graph G is defined as the sum of the following numbers: the number of vertices, the number of arcs, and the sum of absolute values of weights of arcs. The time of the construction of $M(G)$ is $O(n^2)$, where n is the size of G. A CFG for $L(M(G)) \cap L$ can be constructed in time $O(n^3)$, where n is the size of $M(G)$, defined as the number of transitions. The emptiness problem for a context-free language can be solved in time $O(n^2)$, where n is the size of the given CFG for the language, defined as the sum of the number of variables and the number of rules. Since the construction of $G(R)$ can be performed in linear time, we have proved the following theorem.

Theorem 4. *LPPP is solvable in polynomial time.*

As a consequence, the provability problem for **CBL** enriched with letter promotions (the word problem for pregroups with letter promotions) is solvable in polynomial time. First, $X \Rightarrow Y$ is derivable iff $X, Y^{(1)} \Rightarrow \epsilon$ is so. By Theorem 2, $X \Rightarrow \epsilon$ is derivable iff X can be reduced to ϵ by generalized contractions $Y,t,u,Z \Rightarrow Y,Z$ such that t,u appear in X and $t,u \Rightarrow \epsilon$ is derivable. The latter is equivalent to $t \Rightarrow_R^* u^{(-1)}$. By Theorem 4, the required instances of generalized contractions can be determined in polynomial time on the basis of R and X.

Corollary 1. *The word problem for pregroups with letter promotions is solvable in polynomial time.*

A pregroup grammar with letter promotions can be defined as a pregroup grammar in section 1 except that R is a finite set of letter promotions such that $P(R) \subseteq P$. $T^+(G)$ denotes the set of types appearing in I (of G) and $T(G)$ the set of terms occurring in the types from $T^+(G)$. One can compute all generalized contractions $t, u \Rightarrow \epsilon$, derivable from R in **CBL**, for arbitrary terms $t, u \in T(G)$. As shown in the above paragraph, this procedure is polynomial.

As in [3] for pregroup grammars, one can prove that pregroup grammars with letter promotions are equivalent to $\epsilon-$free context-free grammars. For $G = (\Sigma, P, I, s, R)$, one constructs a CFG G' (in an extended sense) in which the terminals are the terms from $T(G)$ and the nonterminals are the terminals and 1, the start symbol equals the principal type of G and the production rules are:

(P1) $u \mapsto t$, if $R \vdash_{CBL} t \Rightarrow u$,
(P2) $1 \mapsto t, u$, if $R \vdash_{CBL} t, u \Rightarrow \epsilon$,
(P3) $t \mapsto 1, t$ and $t \mapsto t, 1$, for any $t \in T(G)$.

By Theorem 2, G' generates precisely all strings $X \in (T(G))^+$ such that $R \vdash_{CBL} X \Rightarrow s$. $L(G) = f[g^{-1}[L(G')]]$, where $g : \Sigma \times T^+(G) \mapsto T^+(G)$ is a partial mapping, defined by $g((v, X)) = X$ whenever $(v, X) \in I$, and $f : \Sigma \times T^+(G) \mapsto \Sigma$ is a mapping, defined by $f((v, X)) = v$ (we extend f, g to homomorphisms of free monoids). Consequently, $L(G)$ is context-free, since the context-free languages are closed under homomorphisms and inverse homomorphisms.

Pregroup grammars with letter promotions can be transformed into equivalent context-free grammars in polynomial time (as in [5] for pregroup grammars), and the membership problem for the former is solvable in polynomial time. A parsing algorithm of complexity $O(n^3)$ can be designed, following the ideas of Oehrle [15] or Moroz [14]; see [13].

Acknowledgements. The polynomial time complexity of LPPP has been announced by the authors at the conference "Topology, Algebra and Categories in Logic", Amsterdam, 2009. The conference did not publish any proceedings.

References

1. Béchet, D., Foret, A.: Fully lexicalized pregroup grammars. In Leivant, D., de Queiroz, P., eds.: Logic, Language, Information and Computation. Volume 4576 of Lecture Notes in Computer Science. Springer (2007) 12–25
2. Buszkowski, W.: Mathematical linguistics and proof theory. In van Benthem, J., ter Meulen, A., eds.: Handbook of Logic and Language. Elsevier Science B. V. (1997) 683–736
3. Buszkowski, W.: Lambek grammars based on pregroups. In de Groote, P., Morrill, G., Retoré, C., eds.: Logical Aspects of Computational Linguistics. Volume 2099 of Lecture Notes in Artificial Intelligence. Springer (2001) 95–109
4. Buszkowski, W.: Sequent systems for compact bilinear logic. Mathematical Logic Quarterly **49**(5) (2003) 467–474

5. Buszkowski, W., Moroz, K.: Pregroup grammars and context-free grammars. In Casadio, C., Lambek, J., eds.: Computational Algebraic Approaches to Natural Language. Polimetrica (2008) 1–21
6. Francez, N., Kaminski, M.: Commutation-augmented pregroup grammars and mildly context-sensitive languages. Studia Logica **87**(2-3) (2007) 297–321
7. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, Reading (1979)
8. Lambek, J.: Type grammars revisited. In Lecomte, A., Lamarche, F., Perrier, G., eds.: Logical Aspects of Computational Linguistics. Volume 1582 of Lecture Notes in Artificial Intelligence. Springer (1999) 1–27
9. Lambek, J.: Type grammars as pregroups. Grammars **4** (2001) 21–39
10. Lambek, J.: From Word to Sentence: a computational algebraic approach to grammar. Polimetrica (2008)
11. Mater, A.H., Fix, J.D.: Finite presentations of pregroups and the identity problem. In: Proc. of Formal Grammar - Mathematics of Language, CSLI Publications (2005) 63–72
12. Moortgat, M., Oehrle, R.T.: Pregroups and type-logical grammar: Searching for convergence. In Casadio, C., Scott, P.J., Seely, R.A., eds.: Language and Grammar. Studies in Mathematical Linguistics and Natural Language. Volume 168 of CSLI Lecture Notes. CSLI Publications (2005) 141–160
13. Moroz, K.: Algorithmic problems for pregroup grammars. PhD thesis, Adam Mickiewicz University, Poznań (2010)
14. Moroz, K.: A Savateev-style parsing algorithm for pregroup grammars. Lecture Notes in Computer Science **5591** (2010) To appear
15. Oehrle, R.T.: A parsing algorithm for pregroup grammars. In: Categorial Grammars: an Efficient Tool for Natural Language Processing, University of Montpellier (2004) 59–75
16. Savateev, Y.: Unidirectional Lambek grammars in polynomial time. Theory of Computing Systems (2010) To appear
17. van Benthem, J.: Language in Action. Categories, Lambdas and Dynamic Logic. Studies in Logic and The Foundations of Mathematics. North-Holland, Amsterdam (1991)