Journal of Computer and System Sciences 78 (2012) 1899–1909

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Pregroup grammars with letter promotions: Complexity and context-freeness

Wojciech Buszkowski [a,*], Zhe Lin [a,b], Katarzyna Moroz [a]

[a] Faculty of Mathematics and Computer Science, Adam Mickiewicz University in Poznań, Poland
[b] Institute of Logic and Cognition, Sun Yat-sen University in Guangzhou, China

ARTICLE INFO

Article history:
Received 17 November 2010
Received in revised form 8 February 2011
Accepted 17 November 2011
Available online 23 December 2011

Keywords:
Type grammar
Pregroup
Context-free grammar
Computational complexity

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 (2005) [13], 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. We also prove that pregroup grammars with letter promotions are equivalent to context-free grammars. At the end, we obtain similar results for letter promotions with unit.

1. Introduction and preliminaries

Pregroup grammars were introduced by Lambek [10] together with the underlying notion of a pregroup. They are lexical grammars in the sense that the major linguistic information is contained in types assigned to lexical items (words) of the particular language. This resembles categorial grammars (or: type grammars). The basic difference consists in the structure of types and their logic; while categorial grammars apply functional types and substructural type logics, usually different variants of the Lambek calculus, pregroup types are elements of a free monoid, generated by iterated adjoints of some atoms, and the logic is a calculus of free pregroups, also called Compact Bilinear Logic (**CBL**). The major advantage of pregroup grammars in comparison with Lambek categorial grammars seems to be their lower complexity. **CBL** is polynomial, while the associative Lambek calculus is NP-complete [19]. As a consequence, although Lambek categorial grammars are equivalent to (ϵ-free) context-free grammars [18], there is no polynomial time transformation of a Lambek grammar into an equivalent CFG, nor does a polynomial time parsing algorithm exist, which can be applied directly to Lambek grammars (if $\mathrm{P} \neq \mathrm{NP}$). Such transformations and parsing algorithms are known for pregroup grammars [6,16].

In this paper we study pregroup grammars with letter promotions $p^{(m)} \Rightarrow q^{(n)}$, which generalize Lambek's poset arrows $p \Rightarrow q$ (see below). Our work has been inspired by [13], where the authors show that **CBL** with letter promotions is NP-hard (they claim NP-completeness). Here we prove that the word problem for pregroups with letter promotions remains in P, if one counts the size of $p^{(n)}$ as $|n|+1$, which seems natural, since $p^{(n)}$ represents the n-th iteration of an adjoint operation on p; below we discuss this question in detail. As a consequence, several other problems related to pregroup grammars with letter promotions are in P. As observed in [13], more general assumptions $X \Rightarrow Y$, where X, Y are finite strings of terms $p^{(n)}$, lead, in general, to undecidable problems.

* Corresponding author.
E-mail address: buszko@amu.edu.pl (W. Buszkowski).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2011.12.010

Pregroups, introduced in Lambek [10], are ordered algebras $(M, \leqslant, \cdot, l, r, 1)$ such that $(M, \leqslant, \cdot, 1)$ is a partially ordered monoid (hence $\cdot$ is monotone in both arguments), and l, r are unary operations on M, fulfilling the following conditions:

$$a^l a \leqslant 1 \leqslant a a^l, \qquad a a^r \leqslant 1 \leqslant a^r a \tag{1}$$

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. This terminology is borrowed from category theory; adjoint functors display a similar behavior.

The following laws are valid in pregroups:

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

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

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

$$a \leqslant b \quad \text{iff} \quad b^l \leqslant a^l \quad \text{iff} \quad b^r \leqslant a^r. \tag{5}$$

We prove the first equation in (4). By (1), $(ab)^l ab \leqslant 1$, so $(ab)^l abb^l a^l \leqslant b^l a^l$. Then we use $1 \leqslant bb^l$, $1 \leqslant aa^l$, which yields $(ab)^l \leqslant b^l a^l$. To show $b^l a^l \leqslant (ab)^l$ begin with $1 \leqslant ab(ab)^l$. The remaining proofs are left to the reader.

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

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

for all elements a, b, c. Consequently, pregroups are a special class of residuated monoids, i.e. models of the associative Lambek calculus admitting sequents with empty antecedents, denoted by $\mathbf{L}^*$ [4,3].

A pregroup satisfying $a^l = a^r$ for all elements a is simply a partially ordered group, and a^l is the inverse of a. This equation holds in all commutative pregroups, so they are commutative partially ordered groups. So pregroups are a generalization of partially ordered groups. Pregroups are more suitable for linguistic purposes than groups for several reasons. For instance, modifiers, e.g. adverbs, prepositional phrases, are assigned types of the form XX^l and $X^r X$ in pregroups, which correspond to XX^{-1} and $X^{-1}X$ in groups. Since $XX^{-1} = X^{-1}X = 1$ in groups, then a phrase of the latter type(s) could always be inserted (removed) in (from) any place in the sentence – a linguistic nonsense.

Lambek [10] (also see [11,12]) proposes (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 [2,3]; linguists usually employ the system $\mathbf{L}$ of the Lambek calculus, which is complete with respect to residuated semigroups (it is weaker than $\mathbf{L}^*$).

The logic of pregroups is called Compact Bilinear Logic (**CBL**). It arises from Bilinear Logic (Noncommutative **MLL**) by collapsing 'times' and 'par'. **CBL** is stronger than $\mathbf{L}^*$; $(p/((q/q)/p))/p \leqslant p$ is valid in pregroups but not in residuated monoids [4], whence it is provable in **CBL**, but not in $\mathbf{L}^*$. By the same example, **CBL** is stronger than Bilinear Logic, since the latter is a conservative extension of $\mathbf{L}^*$.

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

$$\left(a^{(n)}\right)^l = a^{(n-1)}, \qquad \left(a^{(n)}\right)^r = a^{(n+1)}, \quad \text{for all } n \in \mathbb{Z}, \tag{7}$$

$$a^{(n)} a^{(n+1)} \leqslant 1 \leqslant a^{(n+1)} a^{(n)}, \quad \text{for all } n \in \mathbb{Z}, \tag{8}$$

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

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

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

where $\mathbb{Z}$ denotes the set of integers.

CBL can be formalized as follows (after [10], we admit a finite number of non-lexical assumptions $p \Rightarrow q$, which express some subtyping relations). Let $(P, \leqslant)$ 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 $p \leqslant q$, for even n, and $q \leqslant p$, for odd n,

called Contraction, Expansion, and Poset rules, respectively (the latter are called Induced Steps in Lambek [10]). 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 $\mu(p) \leqslant \mu(q)$ in M whenever $p \leqslant q$ in $(P, \leqslant)$. Clearly any assignment μ is uniquely extendable 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, $\mu(X) \leqslant \mu(Y)$ [4]. Our Theorem 1 yields a stronger completeness result.

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 $\pi_k \leqslant \pi$, 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 $s_i \leqslant s$, for $i = 1, 2$, one can assign type s (statement) to all sentences listed above. Let o, i, j be types of object, infinitive and infinitive of a complete verb phrase, respectively, with the new assumption $i \leqslant j$. Then 'I will find him' can be parsed as:

$$(\pi_1)(\pi^{(1)} s_1 j^{(-1)})(i\, o^{(-1)})(o) \quad\Rightarrow\quad s_1 \quad\Rightarrow\quad s.$$

These examples come from [10]. The book [12] offers much more examples and references to other works on pregroup grammars. Let us consider a sentence 'Alice likes everybody'. Assume n is a type of name. Then we can add an assumption $n \leqslant \pi_3$. 'Alice likes everybody' can be parsed as:

$$(n)(\pi_3^{(1)} s_1 j^{(-1)} i\, o^{(-1)})(o\, i^{(1)} i) \quad\Rightarrow\quad s_1 \quad\Rightarrow\quad s.$$

Let us assume p_i is a type of a participle. Then 'He was seen' can be parsed as:

$$(\pi_3)(\pi_3^{(1)} s_2\, o^{(-2)} p_2^{(-1)})(p_2\, o^{(-1)}) \quad\Rightarrow\quad s_2 \quad\Rightarrow\quad s.$$

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 $p \leqslant q$ 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 ϵ-free context-free grammars [4]; 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 transforms 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 [6].

Lambek [10] proves a normalization theorem for **CBL** (also called: Lambek Switching Lemma). One introduces new rules:

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

if either n is even and $p \leqslant q$, or n is odd and $q \leqslant p$. 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** [5]. Below we prove Theorem 2 which strengthens Lambek's result.

We show that this yields the polynomial time complexity of the provability problem for **CBL** [4,5]. For any type X, define X^l and X^r as follows:

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

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:

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

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 [14]). Furthermore, every pregroup grammar can be transformed into an equivalent context-free grammar in polynomial time [4,6].

Francez and Kaminski [7] show that pregroup grammars augmented with partial commutation can generate some non-context-free languages; this also holds for tupled pregroup grammars of Stabler [22] and product pregroup grammars [9].

We have formalized **CBL** with special assumptions. Assumptions $p \leqslant q$ 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 [13] 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 $R \vdash_{\mathbf{CBL}} t \Rightarrow u$, given R, t, u.

Since the formalism is not based on any 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 now mean 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 $R \vdash_{\mathbf{CBL}} X \Rightarrow Y$ and $R \vdash_{\mathbf{CBL}} Y \Rightarrow X$. One defines: $[X] \cdot [Y] = [XY]$, $[X]^{\alpha} = [X^{\alpha}]$, for $\alpha \in \{l, r\}$, $[X] \leqslant [Y]$ iff $R \vdash_{\mathbf{CBL}} X \Rightarrow Y$. For $\mu(p) = [p]$, $p \in P$, one proves: $X \Rightarrow Y$ is true in (M, μ) iff $R \vdash_{\mathbf{CBL}} X \Rightarrow Y$. □

Mater and Fix [13] 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 the decimal (or: binary, etc.) code; see [8,21]. 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: $a \leqslant a^{ll}$ iff a is surjective (i.e. $ax = b$ has a solution, for any b), and $a^{ll} \leqslant a$ iff a is injective (i.e. $ax = ay$ implies $x = y$) [4]. 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 [10,12] use at most three (usually, one or two) iterated left or right adjoints. Accordingly, binary encoding does not seem very useful for such applications.

It seems more natural to look at $p^{(n)}$ as an abbreviated notation for $p^{l\ldots l}$ or $p^{r\ldots r}$, 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). It confirms a final remark from [13]: "Though allowing general letter promotions make the word problem for pregroups NP-hard, there may be algorithms that work well in practice."

We also prove that pregroup grammars with letter promotions are weakly equivalent to (ϵ-free) context-free grammars, which strengthens a result of [4] for pregroup grammars. At the end, we obtain analogous results for more general letter promotions, admitting promotions $p^{(n)} \Rightarrow \epsilon$ and $\epsilon \Rightarrow p^{(n)}$; they come from the PhD thesis [15], and all other results are due to the first two authors.

Oehrle [17] and Moroz [16] 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 [20]). These algorithms can be adjusted for pregroup grammars with letter promotions [15], but we skip this matter here.

Although assumptions of a more general form can be useful in NLP (they can express restricted commutativity or contraction for certain special types), it is not easy to find linguistic applications of letter promotions (different from Lambek's poset arrows $p \Rightarrow q$), which postulate some natural subtyping properties. One of the reasons is that single adjoint types $p^{(n)}$, with $n \neq 0$, are never assigned to expressions of main syntactic categories, at least in the works of Lambek and his collaborators. Letter promotions may, however, be justified from a more theoretical perspective, involving cancellation grammars.

By *a cancellation grammar* we mean a tuple $G = (\Sigma, V, X, R, I)$ such that Σ is a finite (terminal) alphabet, V is a finite (auxiliary) alphabet, disjoint with Σ, $X \in V^*$, R is a finite set of cancellation rules of the form $A, B \Rightarrow \epsilon$, $A \Rightarrow B$, where

$A, B \in V$, and I is a mapping which assigns a finite set of strings on V to any element of Σ. We say that G assigns a string $Y \in V^*$ to a string $a_1 \ldots a_n$ on Σ ($a_i \in \Sigma$), if there exist strings $Y_i \in I(a_i)$, $1 \leqslant i \leqslant n$, such that the string $Y_1 \cdots Y_n$ reduces to Y by a finite number of applications of rules from R. The language of G, denoted $L(G)$, is the set of all strings $Z \in \Sigma^+$ which are assigned X by G (thus X plays the part of the principal type).

Clearly the pregroup grammars of Lambek [10] are a special kind of cancellation grammars. A pregroup grammar $G = (\Sigma, P, I, s, R)$ is equivalent to a cancellation grammar $G' = (\Sigma, T(G), s, R', I)$ such that $T(G)$ is the set of all terms appearing in I, and R' consists of all generalized contractions $tu \Rightarrow \epsilon$ and induced steps $t \Rightarrow u$, for $t, u \in T(G)$. Also pregroup grammars with letter promotions can be represented in this form; see Section 4. On the other hand, an arbitrary cancellation grammar can be naturally simulated by a pregroup grammar with letter promotions $p \Rightarrow q^{(-1)}$, for any cancellation rule $p, q \Rightarrow \epsilon$, and letter promotions $p \Rightarrow q$, identical to the corresponding rules from R.

For instance, the simple cancellation grammar with $\Sigma = \{a, b\}$, $V = \{A, B\}$, $X = \epsilon$ (recall that X is the principal type), $R = \{A, B \Rightarrow \epsilon\}$, $I(a) = \{A\}$, $I(b) = \{B\}$ generates the Dyck language on $\{a, b\}$. The corresponding pregroup grammar admits one letter promotion $A \Rightarrow B^{(-1)}$. Lambek [12] uses marked types $[i$ which may be contracted by right adjoints i^r but not left adjoints i^l. Constraints of this kind can be formalized, using letter promotions; treat both $[i$ and $i^r]$ as new atoms and admit the promotion $[i, i^r] \Rightarrow \epsilon$, equivalent to the letter promotion $[i \Rightarrow (i^r])^{(-1)}$. More generally, if one wants to contract an atom p with an atom q on the right (resp. on the left), then one may stipulate $p, q \Rightarrow \epsilon$ (resp. $q, p \Rightarrow \epsilon$) as a new promotion, equivalent to a letter promotion; a single atom p can be contracted with several q_i. It seems that the framework of cancellation grammars, corresponding to pregroup grammars with letter promotions, may be more flexible in NLP than the original framework of pregroup grammars.

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 $k \geqslant 0$, $t_0 = t$, $t_k = u$, and $t_{i-1} \Rightarrow_R t_i$, for all $i = 1, \ldots, k$. Hence $\Rightarrow_R^*$ is the reflexive and transitive closure of $\Rightarrow_R$.

We 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 [13], 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 the forthcoming 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 $l \leqslant k$. 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 $l = k$. We proceed with case analysis.

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 the pair $p^{(n+1)}, q^{(m)}$. 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 $X, Y_1, \ldots, Y_l$ can be transformed into a shorter derivation: drop the first application of (PRO-R), derive $Y_1, \ldots, Y_{i-1}$ 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 $Y_{i+1}, \ldots, Y_l$. 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 $X_0, Y_1, \ldots, Y_l$ can be transformed into a shorter derivation: drop the first application of (PRO-R), derive $Y_1, \ldots, Y_{i-1}$ 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 with $Y_{i+1}, \ldots, Y_l$. 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 $G(R)$. Let $P(R)$ denote the set of atoms occurring in promotions from R. The vertices of $G(R)$ are exactly the 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 $G(R)$ (see Fig. 1 for an example).

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 $(v_0, k_1, v_1), \ldots, (v_{r-1}, k_r, v_r)$ 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.

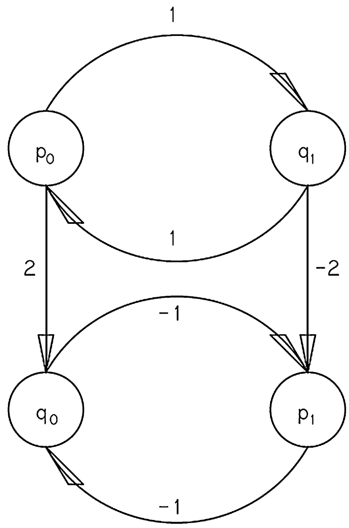

Fig. 1. $G(R)$ for $R = \{p \Rightarrow q^{(1)},\ p^{(1)} \Rightarrow q,\ p \Rightarrow q^{(2)}\}$.

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 $s \in P(R)$ 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. Polynomial complexity

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. Recall that 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 nondeterministic 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,

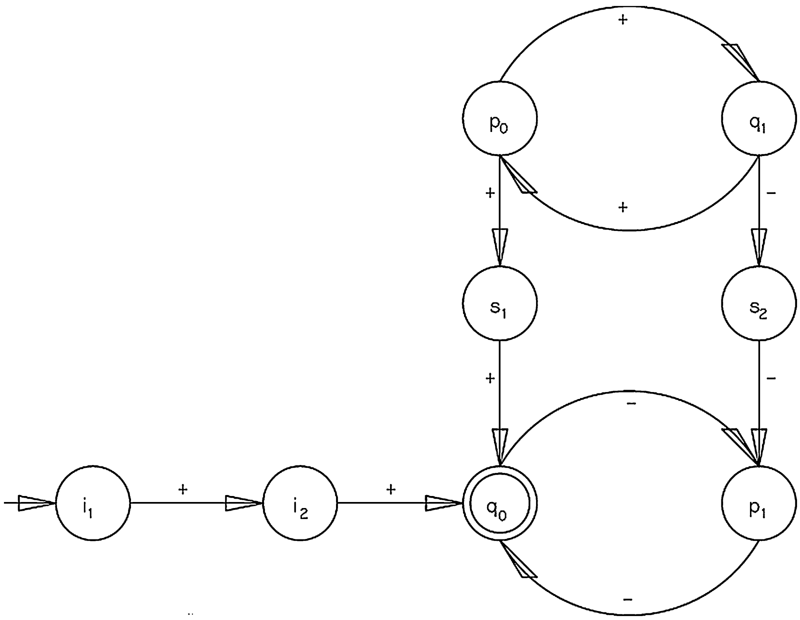

Fig. 2. $M(G(R))$ for $G(R)$ from Fig. 1 and the problem $q^{(2)} \Rightarrow_R^* q$ ($v = w = q_0$, $k = -2$).

$n > 0$, 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 (see Fig. 2 for an example). 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 Chomsky Normal Form (i.e. all rules are of the form $A \mapsto BC$ or $A \mapsto a$) in constant time. The CFG just obtained is modified to a CFG for $L(M(G)) \cap L$ in the standard way. The new variables (nonterminals) are of the form (q, A, q'), where q, q' are arbitrary states of $M(G)$, and A is a variable of the CFG in Chomsky Normal Form. 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 derivable. 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 paragraph above, this procedure is polynomial.

Theorem 5. *Pregroup grammars with letter promotions are equivalent to ϵ-free context-free grammars.*

Proof. It was shown in [4] that any ϵ-free context-free language can be generated by a pregroup grammar on a trivial poset $(P, =)$. This yields one half of the theorem.

For a pregroup grammar $G = (\Sigma, P, I, s, R)$, one constructs a CFG G' (in an extended sense: the terminal alphabet need not be disjoint with the nonterminal one) 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 $R \vdash_{\mathbf{CBL}} t \Rightarrow u$,
(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 [6] 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 [17] or Moroz [16]; see [15].

5. Letter promotions with unit

The above results can be extended to letter promotions with unit $p^{(n)} \Rightarrow \epsilon$ and $\epsilon \Rightarrow p^{(n)}$. Since in pregroups $a^{(n)} \leqslant 1$ is equivalent to $a \leqslant 1$, if n is even, and to $1 \leqslant a$, if n is odd, and similarly for $1 \leqslant a^{(n)}$, it suffices to consider promotions of the form $p \Rightarrow \epsilon$ and $\epsilon \Rightarrow p$. In this section, the set R can contain letter promotions $p^{(m)} \Rightarrow q^{(n)}$ and promotions $p \Rightarrow \epsilon$, $\epsilon \Rightarrow q$ such that m, n are integers and p, q are atoms. The set $P(R)$ is defined as above.

The complete logic contains two new rewriting rules:

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

Perhaps it would be more natural to write 1 instead of ϵ, but it would require new rules for the constant 1. In this paper we do not include 1 in the language of **CBL**. The extended system of **CBL** with rules (CON), (EXP), (PRO), (PRO-C) and (PRO-E) is denoted by **CBL***. $R \vdash_{\mathbf{CBL}^*} X \Rightarrow Y$ means that $X \Rightarrow Y$ is derivable in **CBL*** from the set of assumptions R.

Theorem 1 can easily be generalized for **CBL***. Theorem 2 needs additional notions.

We assume that ϵ is also a term (other terms are of the form $p^{(n)}$). We write $t \Rightarrow_R u$, if $t \Rightarrow u$ is a single instance of (PRO), (PRO-C) or (PRO-E) (with X, Y empty). $\Rightarrow_R^*$ denotes the reflexive and transitive closure of the relation $\Rightarrow_R$.

We introduce some derivable rules of **CBL***.

(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 $p^{(n)} \Rightarrow_R^* q^{(m)}$,
(PRO*-R) $X, t, Y \Rightarrow X, u, Y$, if $t \Rightarrow_R^* u$, $t \neq \epsilon$, $u \neq \epsilon$,
(PRO*-C-R) $X, t, Y \Rightarrow X, Y$, if $t \Rightarrow_R^* \epsilon$, $t \neq \epsilon$,
(PRO*-E-R) $X, Y \Rightarrow X, u, Y$, if $\epsilon \Rightarrow_R^* u$, $u \neq \epsilon$.

Theorem 6. *If $R \vdash_{\mathbf{CBL}^*} X \Rightarrow Y$, then there exist Z, U such that $X \Rightarrow Z$ is provable by a finite number of instances of (GCON*-R) and (PRO*-C-R), $Z \Rightarrow U$ is provable by a finite number of instances of (PRO*-R), and $U \Rightarrow Y$ is provable by a finite number of instances of (GEXP*-R) and (PRO*-E-R).*

We omit the proof, which can be found in [15]. Its construction is similar to the proof of Theorem 2; the only difference is that one must consider more cases and use new generalized contractions (PRO*-C-R) and new generalized expansions (PRO*-E-R).

Corollary 2. *If $R \vdash_{\mathbf{CBL}^*} X \Rightarrow t$, where t is a term, then X can be reduced to t by a finite number of instances of (GCON*-R), (PRO*-C-R), (PRO*-R), and (PRO*-E-R), where the latter rules are applied at the end (if anywhere).*

Lemma 4. *$R \vdash_{\mathbf{CBL}^*} t \Rightarrow u$ if, and only if, $t \Rightarrow^*_R u$.*

Proof. The 'if' part is obvious. We prove the 'only if' part. Assume $R \vdash_{\mathbf{CBL}^*} t \Rightarrow u$. There exists a derivation of $t \Rightarrow u$ of the form from Corollary 2. Since (GCON*-C-R) cannot be applied to a single term, then it is not applied at all. Consequently, this derivation applies at most (PRO), (PRO-C) and (PRO-E) (with the empty context), which yields $t \Rightarrow^*_R u$. □

We will prove that LPPP, admitting promotions with unit, is solvable in polynomial time.

For a finite set R of letter promotions, which possibly contains promotions with unit, by R' we denote the set of all letter promotions (without unit) from R. The graph $G(R')$ is defined as in Section 3. $G(R)$ is $G(R')$ enriched with all vertices p_0, p_1, for $p \in P(R) - P(R')$, treated as isolated vertices.

The following modification of Theorem 3 is true: for any $p, q \in P(R)$ and integers m, n, $p^{(m)} \Rightarrow^*_{R'} q^{(n)}$ if, and only if, there exists a route in $G(R)$ from $p_{\pi(m)}$ to $q_{\pi(n)}$ of weight $n - m$. The 'only if' part follows from Theorem 3, since $G(R')$ is a subgraph of $G(R)$, and there exists the trivial route of weight 0 from p_i to p_i in $G(R)$, for $p \in P(R) - P(R')$. We prove the 'if' part. Assume that there exists a route from $p_{\pi(m)}$ to $q_{\pi(n)}$ of weight $n - m$ in $G(R)$. If the route is trivial, then $m = n$ and $p = q$, so $p^{(m)} \Rightarrow^*_{R'} q^{(n)}$. If the route is nontrivial, then it must be a route in $G(R')$, hence $p^{(m)} \Rightarrow^*_{R'} q^{(n)}$, by Theorem 3.

Lemma 5. *Let $p, q \in P(R)$. Then, $p^{(m)} \Rightarrow^*_R \epsilon$ if, and only if, one of the following conditions holds:*

(i) *there exist a vertex q_0 of $G(R)$ such that $(q \Rightarrow \epsilon) \in R$ and a route in $G(R)$ from $p_{\pi(m)}$ to q_0 whose weight is even, if m is even, and odd, if m is odd,*
(ii) *there exist a vertex q_1 of $G(R)$ such that $(1 \Rightarrow q) \in R$ and a route in $G(R)$ from $p_{\pi(m)}$ to q_1 whose weight is odd, if m is even, and even, if m is odd.*

Proof. We prove the 'if' part. Assume (i). Then, for some $q \in P(R)$, $(q \Rightarrow \epsilon) \in R$, and there exists a route from $p_{\pi(m)}$ to q_0 of some weight k in $G(R)$, and k is even if, and only if, m is even. So $k + m$ is even, hence $\pi(k+m) = 0$. By Theorem 3 (in the modified form), $p^{(m)} \Rightarrow^*_{R'} q^{(k+m)}$. We get $q^{(k+m)} \Rightarrow_R \epsilon$, by (PRO-C), which yields $p^{(m)} \Rightarrow^*_R \epsilon$. Assume (ii). Then, for some $q \in P(R)$, $(\epsilon \Rightarrow q) \in R$, and there exists a route from $p_{\pi(m)}$ to q_1 of some weight k in $G(R)$, and k is even if, and only if, m is odd. So $k + m$ is odd, hence $\pi(k+m) = 1$. By Theorem 3 (in the modified form), $p^{(m)} \Rightarrow^*_{R'} q^{(k+m)}$. We get $q^{(k+m)} \Rightarrow_R \epsilon$, by (PRO-C), which yields $p^{(m)} \Rightarrow^*_R \epsilon$.

We prove the 'only if' part. Assume $p^{(m)} \Rightarrow^*_R \epsilon$. Clearly we may suppose that only the last term of the derivation equals ϵ. If the derivation has only one step, then $p^{(m)} \Rightarrow_R \epsilon$ is an instance of (PRO-C). Either m is even and $(p \Rightarrow \epsilon) \in R$, or m is odd and $(\epsilon \Rightarrow p) \in R$. The first case yields (i) with $p = q$ and the trivial route from p_0 to p_0, and the second case yields (ii) with $p = q$ and the trivial route from p_1 to p_1. Assume now that the derivation has at least two steps. Then, it consists of some (PRO)-steps (based on promotions from R') and one (PRO-C)-step at the end. Consequently, there exists a term $q^{(n)}$ such that $p^{(m)} \Rightarrow^*_{R'} q^{(n)}$ and $q^{(n)} \Rightarrow_R \epsilon$. By Theorem 3 (modified), there exists a route in $G(R)$ from $p_{\pi(m)}$ to $q_{\pi(n)}$ of weight $n - m$ in $G(R)$. If n is even, then $(q \Rightarrow \epsilon) \in R$, and $n - m$ is even if, and only if, m is even; this yields (i). If n is odd, then $(\epsilon \Rightarrow q) \in R$, and $n - m$ is even if, and only if, m is odd; this yields (ii). □

Lemma 6. *Let $p, q \in P(R)$. Then, $\epsilon \Rightarrow^*_R q^{(n)}$ if, and only if, one of the following conditions holds:*

(i) *there exist a vertex p_0 of $G(R)$ such that $(\epsilon \Rightarrow p) \in R$ and a route in $G(R)$ from p_0 to $q_{\pi(n)}$ whose weight is even, if n is even, and odd, if n is odd,*
(ii) *there exist a vertex p_1 of $G(R)$ such that $(p \Rightarrow \epsilon) \in R$ and a route in $G(R)$ from p_1 to $q_{\pi(n)}$ whose length is even, if n is odd, and odd, if n is even.*

We skip the proof, similar to the proof of Lemma 5. The next theorem provides a complete list of cases for $R \vdash_{\mathbf{CBL}^*} t \Rightarrow u$.

Theorem 7. *For any terms t, u, $t \Rightarrow^*_R u$ holds if, and only if, one of the following conditions is true:*

(i) $t = u$,
(ii) $t = p^{(m)}$, $u = \epsilon$, *and* (i) *or* (ii) *of Lemma* 5 *holds,*
(iii) $t = \epsilon$, $u = q^{(n)}$, *and* (i) *or* (ii) *of Lemma* 6 *holds,*
(iv) $t = p^{(m)}$, $u = q^{(n)}$, *and both* (i) *or* (ii) *of Lemma* 5 *holds and* (i) *or* (ii) *of Lemma* 6 *holds* (*hence $t \Rightarrow^*_R \epsilon$ and $\epsilon \Rightarrow^*_R u$*),
(v) $t = p^{(m)}$, $u = q^{(n)}$, *and there exists a route in $G(R')$ from $p_{\pi(m)}$ to $q_{\pi(n)}$ of weight $n - m$.*

Proof. This theorem easily follows from Theorem 3 and Lemmas 5 and 6. The 'if' part is obvious. The 'only if' part is also easy. Let $t \neq u$. Assume $t \Rightarrow^*_R u$. The derivation has at least one step, hence t, u are empty or of the form $p^{(m)}$, for $p \in P(R)$. If one of them is empty, then we get clauses (ii) or (iii). Assume that $t = p^{(m)}$, $u = q^{(n)}$. If $t \Rightarrow^*_R \epsilon$ and $\epsilon \Rightarrow^*_R u$, then (iv) must be true; otherwise (v) must be true. □

It remains to show that each of the above conditions can be checked in time polynomial in size of R, t, u (remind that $p^{(m)}$ is of size $|m| + 1$). For (i) it is obvious, and (v) can be handled as in Section 4. For (ii), (iii), (iv), it suffices to show that conditions (i), (ii) from Lemmas 5 and 6 can be checked in polynomial time.

For the graph $G = G(R)$, we construct a nondeterministic FSA $N(G)$ which is similar to $M(G)$ from Section 4 except that we neither introduce auxiliary states $i_1, \ldots, i_k$, nor fix initial and final states (so we only construct the transition graph of an FSA). A simple automaton E with alphabet $\{0, 1\}$ and two states e, o (e being the initial state) reaches state e on any input of even length and state o on any input of odd length. We consider the product automaton $N(G) \times E$; note that in this case states are pairs (s, i) such that s is a state of $N(G)$ and $i \in \{e, o\}$.

The condition (i) of Lemma 5 holds if, and only if, there exists a promotion $(q \Rightarrow \epsilon) \in R$ such that $L(M_q^0) \neq \emptyset$, where M_q^0 is the automaton $N(G) \times E$ with the initial state $(p_{\pi(m)}, e)$ and the final state equal to (q_0, e), if m is even, or equal to (q_0, o), if m is odd.

The condition (ii) of Lemma 5 holds if, and only if, there exists a promotion $(\epsilon \Rightarrow q) \in R$ such that $L(M_q^1) \neq \emptyset$, where M_q^1 is the automaton $N(G) \times E$ with the initial state $(p_{\pi(m)}, e)$ and the final state equal to (q_1, o), if m is even, and equal to (q_1, e), if m is odd.

Obviously, the latter conditions can be checked in polynomial time. Conditions (i) and (ii) of Lemma 6 can be handled similarly; we omit details. This yields the following result.

Theorem 8. *LPPP admitting promotions with unit is solvable in polynomial time.*

As in Section 4 we show that the word problem for pregroups with letter promotions with unit is solvable in polynomial time. Also, given a pregroup grammar $G = (\Sigma, P, I, s, R)$ such that R is a finite set of letter promotions possibly with unit, and the underlying logic is **CBL***, one can compute in polynomial time all generalized contractions $R \vdash_{\mathbf{CBL}^*} t, u \Rightarrow \epsilon$ as well as all instances of $t \Rightarrow_R^* u$, $t \Rightarrow_R^* \epsilon$ and $\epsilon \Rightarrow_R^* u$, for $t, u \in T(G)$. This yields a polynomial construction of a CFG G' such that $L(G') = \{X \in (T(G))^+ : R \vdash_{\mathbf{CBL}^*} X \Rightarrow s\}$. The production rules of G' are (P1)–(P3) with **CBL*** instead of **CBL** and:

(P4) $t \mapsto 1$, if $\epsilon \Rightarrow_R^* t$,
(P5) $1 \mapsto t$, if $t \Rightarrow_R^* \epsilon$.

Consequently, $L(G)$ is context-free, since context-free languages are closed under homomorphisms and inverse homomorphisms (see Section 4). This yields the equivalence of pregroup grammars with letter promotions (possibly with unit) and ϵ-free CFGs.

References

[1] D. Béchet, A. Foret, Fully lexicalized pregroup grammars, in: D. Leivant, P. de Queiroz (Eds.), Logic, Language, Information and Computation, in: Lecture Notes in Comput. Sci., vol. 4576, Springer, 2007, pp. 12–25.
[2] J. van Benthem, Language in Action. Categories, Lambdas and Dynamic Logic, Stud. Logic Found. Math., North-Holland, Amsterdam, 1991.
[3] W. Buszkowski, Mathematical linguistics and proof theory, in: J. van Benthem, A. ter Meulen (Eds.), Handbook of Logic and Language, Elsevier Science B.V., 1997, pp. 683–736.
[4] W. Buszkowski, Lambek grammars based on pregroups, in: P. de Groote, G. Morrill, C. Retoré (Eds.), Logical Aspects of Computational Linguistics, in: Lecture Notes in Artificial Intelligence, vol. 2099, Springer, 2001, pp. 95–109.
[5] W. Buszkowski, Sequent systems for compact bilinear logic, MLQ Math. Log. Q. 49 (2003) 467–474.
[6] W. Buszkowski, K. Moroz, Pregroup grammars and context-free grammars, in: C. Casadio, J. Lambek (Eds.), Computational Algebraic Approaches to Natural Language, Polimetrica, 2008, pp. 1–21.
[7] N. Francez, M. Kaminski, Commutation-augmented pregroup grammars and mildly context-sensitive languages, Studia Logica 87 (2007) 297–321.
[8] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison–Wesley, Reading, 1979.
[9] T. Kusalik, Product pregroups as an alternative to inflectors, in: C. Casadio, J. Lambek (Eds.), Computational Algebraic Approaches to Natural Language, Polimetrica, 2008, pp. 173–189.
[10] J. Lambek, Type grammars revisited, in: A. Lecomte, F. Lamarche, G. Perrier (Eds.), Logical Aspects of Computational Linguistics, in: Lecture Notes in Artificial Intelligence, vol. 1582, Springer, 1999, pp. 1–27.
[11] J. Lambek, Type grammars as pregroups, Grammars 4 (2001) 21–39.
[12] J. Lambek, From Word to Sentence: A Computational Algebraic Approach to Grammar, Polimetrica, 2008.
[13] A.H. Mater, J.D. Fix, Finite presentations of pregroups and the identity problem, in: Proc. of Formal Grammar – Mathematics of Language, CSLI Publications, 2005, pp. 63–72.
[14] M. Moortgat, R.T. Oehrle, Pregroups and type-logical grammar: Searching for convergence, in: C. Casadio, P.J. Scott, R.A. Seely (Eds.), Language and Grammar. Studies in Mathematical Linguistics and Natural Language, in: CSLI Lecture Notes, vol. 168, CSLI Publications, 2005, pp. 141–160.
[15] K. Moroz, Algorithmic questions for pregroup grammars, PhD thesis, Adam Mickiewicz University, Poznań, 2010.
[16] K. Moroz, A Savateev-style parsing algorithm for pregroup grammars, in: Lecture Notes in Artificial Intelligence, vol. 5591, Springer, 2011, pp. 133–149.
[17] R.T. Oehrle, A parsing algorithm for pregroup grammars, in: Categorial Grammars: An Efficient Tool for Natural Language Processing, University of Montpellier, 2004, pp. 59–75.
[18] M. Pentus, Lambek grammars are context-free, Proc. 8th IEEE Symp. Logic in Computer Sci. (1993) 429–433.
[19] M. Pentus, Lambek calculus is NP-complete, Theoret. Comput. Sci. 357 (2006) 186–201.
[20] Y. Savateev, Unidirectional Lambek grammars in polynomial time, Theory Comput. Syst. 46 (2010) 662–672.
[21] M. Sipser, Introduction to the Theory of Computation, Thomson Course Technology, Boston, 2006.
[22] E.P. Stabler, Tupled pregroup grammars, in: C. Casadio, J. Lambek (Eds.), Computational Algebraic Approaches to Natural Language, Polimetrica, 2008, pp. 23–52.