Mildly Context-Sensitive Languages via Buffer Augmented Pregroup Grammars

Daniel Genkin, Nissim Francez*, and Michael Kaminski

Department of Computer Science
Technion– Israel Institute of Technology
Haifa 32000
Israel
{danielg3,francez,kaminski}@cs.technion.ac.il

Abstract. A family of languages is called *mildly context-sensitive* if
- it includes the family of all ϵ-free context-free languages;
- it contains the languages
  - $\{a^n b^n c^n : n \geq 1\}$ – *multiple agreement*,
  - $\{a^m b^n c^m d^n : m, n \geq 1\}$ – *crossed dependencies*, and
  - $\{ww : w \in \Sigma^+\}$ – *reduplication*;
- all its languages are *semi-linear*; and
- their membership problem is decidable in polynomial time.

In our paper we introduce a new model of computation called *buffer augmented pregroup grammars* that defines a family of mildly context-sensitive languages. This model of computation is an extension of Lambek pregroup grammars with a variable symbol – the (*buffer*) and is allowed to make an arbitrary substitution from the original pregroup to the variable. This increases the pregroup grammar generation power, but still retains the desired properties of mildly context-sensitive languages such as semi-linearity and polynomial parsing. We establish a strict hierarchy within the family of mildly context-sensitive languages defined by buffer augmented pregroup grammars. In this hierarchy, the hierarchy level of the family language depends on the allowed number of occurrences of the variable in lexical category assignments.

Keywords: Formal language theory, mildly context-sensitive languages, pregroup grammars.

1 Introduction

Since their introduction in [7], *pregroup grammars* have attracted a lot of attention, giving rise to a *radically lexicalized* theory of formal (and, of course, natural) languages. The theory of formal languages partly developed from an abstraction originating in the syntax of natural languages, namely *constituency* (known also as *phrase-structure*). By this abstraction, *rewrite-rules* formed the basis of formal grammar, culminating in their classification by the well-known

* In memory of Amir Pnueli, a teacher and a friend.

Z. Manna and D. Peled (Eds.): Pnueli Festschrift, LNCS 6200, pp. 144–166, 2010.

Chomsky hierarchy. To their success in computer science contributed the realization of their suitability for specifying the syntax of programming languages, after they were abandoned as a tool for natural language syntax specification. The theory matured even more when the grammar classification was complemented by the classification of various classes of automata corresponding to the various classes of the Chomsky hierarchy of grammar formalisms, see [6], a standard reference to the area.

This standard approach to formal languages has certain characteristics, challenged by modern *computational linguistics*, summarized below.

- Terminals are *atomic* structureless entities, that can only be compared for equality.
- Similarly, *non-terminals* (better called *categories*) are also atomic, structureless entities, representing sets of strings (of terminals).
- Language variation (over some fixed set of terminals) is determined by *grammar variation*, which was taken to mean *variation in the rewrite rules.*
- String *concatenation* is the *only* admissible syntactic operation.

Modern computational linguistics is the source of a different abstraction, based on a different view of language theory known as *radical lexicalism.* There are several radically-lexicalized linguistic theories for natural language (we omit references, as the focus here is on *formal* languages), having the following characteristics.

- Terminals are *informative* entities, that have their own properties, determined by a *lexicon*, mapping terminals to "pieces of information" about them. The lexicon is the "heart" of a grammar. Most often, those pieces of information are taken as (finite) sets of complex categories.
- Similarly, *categories* are also structured entities, representing sets of strings (of terminals).
- Language variation (over some fixed set of terminals) is determined by *lexicon variation.* There is a *universal grammar* (common to all languages) that extends the lexicon by attributing categories to strings too, controlling the combinatorics of strings based on their categories.

There are variants that admit other syntactic operations besides concatenation. We will assume here that concatenation is maintained as the only operation.

Buszkowski [2] establishes the weak generative equivalence between pregroup grammars and context-free grammars. On the other hand, motivated by the syntactic structure of natural languages, computational linguists became interested in a family of languages that became to be known as *mildly context-sensitive* languages, that on the one hand transcend context-free languages in containing *multiple agreement* ($\{a^n b^n c^n : n \geq 1\}$), *crossed dependencies* ($\{a^m b^n c^m d^n : m, n \geq 1\}$), and *reduplication* ($\{ww : w \in \Sigma^+\}$), but on the other hand have *semi-linearity* [8] and their membership problem is decidable in polynomial time (in the length of the input word). Several formalisms for grammar specification are known to converge to the same class [9], namely to mildly context-sensitive languages.

In this paper, we explore a mild extension of pregroup grammars, obtained by adding to the underlying (free) pregroup a new element – the buffer, that is a lower bound on some set of elements of the underlying free pregroup, cf. [1]. We establish the main properties of this class of languages, namely semi-linearity and polynomial parsability.

The paper is organized as follows. In Section 2 we review the standard definition of pregroups and grammars based on them. Then, in Section 3 we define buffer augmented pregroup grammars and show that they are powerful enough to generate the characteristic mildly context-sensitive languages. In Section 4 we prove the pumping lemma for the languages generated by buffer augmented pregroup grammars. Sections 5 and 6 deal with complexity issues of languages generated by buffer augmented pregroup grammars. In Section 7 we establish a strict hierarchy in the class of these languages and in Section 8 we extend our model of computation to a number of buffers. Finally, Section 9 contains some concluding remarks.

2 Pregroups and Pregroup Grammars

In this section we recall the definition of pregroup grammars.

Definition 1. *A* pregroup *is a tuple* $\mathcal{P} = \langle \boldsymbol{G}, \leq, \circ, \ell, r, 1\rangle$*, such that* $\langle \boldsymbol{G}, \leq, \circ, 1\rangle$ *is a* partially-ordered monoid,[1] *i.e., satisfying*

(mon) *if* $A \leq B$*, then* $CA \leq CB$ *and* $AC \leq BC$

and ℓ,r *are unary operations (left/right inverses/adjoints) satisfying*

(pre) $A^\ell A \leq 1 \leq AA^\ell$ *and* $AA^r \leq 1 \leq A^r A$*.*

The following equalities can be shown to hold in any pregroup.

$$1^\ell = 1^r = 1,\ A^{\ell r} = A^{r\ell} = A,\ (AB)^\ell = B^\ell A^\ell,\ (AB)^r = B^r A^r.$$

Also, **(mon)** together with **(pre)** yield

$$A \leq B \text{ if and only if } B^\ell \leq A^\ell \text{ if and only if } B^r \leq A^r. \tag{1}$$

Let $\langle \mathcal{B}, \leq\rangle$ be a (finite) partially ordered set. *Terms* are of the form $A^{(n)}$, where $A \in \mathcal{B}$ and n is an integer. The set of all terms generated by $\mathcal{B}$ is denoted by $\tau(\mathcal{B})$.

Categories[2] are finite strings (products) of terms. The set of all categories generated by $\mathcal{B}$ is denoted by $\kappa(\mathcal{B})$.

Remark 1. By definition, $\kappa(\mathcal{B}) = (\tau(\mathcal{B}))^*$.

Extend '$\leq$' to $\kappa(\mathcal{B})$ by setting it to the smallest quasi-partial-order[3] satisfying

[1] '$\circ$' is usually omitted.

[2] They are also called *types*.

[3] That is, $\leq$ is not necessarily antisymmetrical.

(con) $\gamma A^{(n)} A^{(n+1)} \delta \leq \gamma\delta$ (*contraction*)
(exp) $\gamma\delta \leq \gamma A^{(n+1)} A^{(n)} \delta$ (*expansion*)
and
(ind) $\gamma A^{(n)} \delta \leq \gamma B^{(n)} \delta$ if $\begin{cases} A \leq B \text{ and } n \text{ is even, or} \\ B \leq A \text{ and } n \text{ is odd} \end{cases}$ (*induced steps*) .

The following two inequalities can be easily derived from **(con)**, **(exp)**, and **(ind)**.

(gcon) $\gamma A^{(n)} B^{(n+1)} \delta \leq \gamma\delta$, if $\begin{cases} A \leq B \text{ and } n \text{ is even, or} \\ B \leq A \text{ and } n \text{ is odd} \end{cases}$ (*generalized contraction*)

and

(gexp) $\gamma\delta \leq \gamma A^{(n+1)} B^{(n)} \delta$, if $\begin{cases} A \leq B \text{ and } n \text{ is even, or} \\ B \leq A \text{ and } n \text{ is odd} \end{cases}$ (*generalized expansion*) .

Obviously, **(con)** and **(exp)** are particular cases of **(gcon)** and **(gexp)**, respectively. Conversely, **(gcon)** can be obtained as **(ind)** followed by **(con)**, and **(gexp)** can be obtained as **(exp)** followed by **(ind)**. Consequently, if $\alpha' \leq \alpha''$, then there exists a *derivation*

$$\alpha' = \gamma_0 \leq \gamma_1 \leq \cdots \leq \gamma_m = \alpha'', \quad m \geq 0$$

such that for each $i = 1, 2, \ldots, m$, $\gamma_{i-1} \leq \gamma_i$ is **(gcon)**, **(gexp)**, or **(ind)**.

Proposition 1. ([7, Proposition 2]) *If $\alpha' \leq \alpha''$ has a derivation of length m, then there exist categories β and γ such that*

- $\alpha' \leq \beta$ *by* (**gcon**) *only;*
- $\beta \leq \gamma$ *by* **(ind)** *only;*
- $\gamma \leq \alpha''$ *by* **(gexp)** *only; and*
- *the sum of the lengths of the above three derivations is at most m.*

Corollary 1. *If $\alpha' \leq \alpha''$ where α'' is a term, then, effectively, this can be established without expansions.*

Let $\alpha' \equiv \alpha''$ if and only if $\alpha' \leq \alpha''$ and $\alpha'' \leq \alpha'$. The equivalence-classes of '$\equiv$' form the *free pregroup generated by $\langle \mathcal{B}, \leq \rangle$*, where $1 = [\epsilon]_\equiv$, $[\alpha']_\equiv \circ [\alpha'']_\equiv = [\alpha'\alpha'']_\equiv$. Also, $[\alpha']_\equiv \leq [\alpha'']_\equiv$ if and only if $\alpha' \leq \alpha''$. The adjoints are defined as follows.

$$[A_1^{(n_1)} \cdots A_l^{(n_l)}]^\ell = [A_l^{(n_l-1)} \cdots A_1^{(n_1-1)}]$$

and

$$[A_1^{(n_1)} \cdots A_l^{(n_l)}]^r = [A_l^{(n_l+1)} \cdots A_1^{(n_1+1)}].$$

Definition 2. *A* pregroup grammar *(PGG) is a tuple $G = \langle \Sigma, \mathcal{B}, \leq, I, \Delta \rangle$, where*

- Σ *is a finite set of terminals (the* alphabet*),*
- $\langle \mathcal{B}, \leq \rangle$ *is a finite partially ordered set of* atoms,

– *I is a finite-range mapping* $I : \Sigma \to 2^{\kappa(\mathcal{B})}$,[4] *and*
– $\Delta \subset \tau(\mathcal{B})$ *is a finite set of* distinguished categories.[5]

We extend I to Σ^+ by

$$I(w\sigma) = \{\tau\tau' : \tau \in I(w) \text{ and } \tau' \in I(\sigma)\}$$

and define the language $L(G)$ generated by G by

$$L(G) = \{w : \text{there exist } \tau \in I(w) \text{ and } \delta \in \Delta \text{ such that } \tau \leq \delta\}.$$

Example 1. Consider the PGG $G_a = \langle \Sigma, \mathcal{B}, =, I, \Delta \rangle$, where

– $\Sigma = \{a, b\}$,
– $\mathcal{B} = \{S, A\}$, and
– I is defined by
  • $I(a) = \{SA^\ell S^\ell, SA^\ell\}$, and
  • $I(b) = \{A\}$,

  and
– $\Delta = \{S\}$.

It can be readily seen that

$$L(G_a) = \{a^n b^n : n \geq 1\}.$$

Below is a derivation for $a^3 b^3 \in L(G_a)$. The lexical category assignment chosen is

$$\overbrace{SA^\ell S^\ell}^{a}\ \overbrace{SA^\ell S^\ell}^{a}\ \overbrace{SA^\ell}^{a}\ \overbrace{A}^{b}\ \overbrace{A}^{b}\ \overbrace{A}^{b}$$

and cancellation is indicated by underline.

$$SA^\ell \underline{S^\ell S} A^\ell \underline{S^\ell S} A^\ell AAA \leq SA^\ell A^\ell \underline{A^\ell A} AA \leq SA^\ell \underline{A^\ell A} A \leq S\underline{A^\ell A} \leq S.$$

Theorem 1. ([2]) *An ϵ-free language L is $L(G)$ for some pregroup grammar G if and only if L is context-free.*

3 Buffer Augmented PGGs

In this section we introduce buffer augmented pregroup grammars and present some of their basic properties.

Definition 3. *A* buffer augmented *pregroup grammar (BAPGG) is a tuple* $G = \langle \Sigma, \mathcal{B}, \leq, \mathcal{B}', I, \Delta \rangle$, *where the components of G are as follows.*

– *Σ is a finite set of terminals (the* alphabet*).*
– *$\langle \mathcal{B}, \leq \rangle$ is a partially ordered finite set.*

[4] That is, $I(\sigma)$ is finite for all $\sigma \in \Sigma$.
[5] Cf. an equivalent definition in [2], where Δ consists of one term only.

– $\mathcal{B}' \subseteq \mathcal{B}$ *is the set of the* buffer *elements.*
– I *is a mapping that assigns to each element of* Σ *a finite set of categories from* $\kappa(\mathcal{B} \cup \{x\})$*, where* x *is a new (variable) symbol – the buffer, such that for all* $\sigma \in \Sigma$*, each* $\tau \in I(\sigma)$ *is of one of the following forms:*
  (*i*) $\tau \in \kappa(\mathcal{B} \setminus \mathcal{B}')$,
  (*ii*) $\tau = \alpha A^{(\pm 1)} \beta$*, where* $A \in \mathcal{B}'$*,* $\alpha, \beta \in \kappa(\mathcal{B} \setminus \mathcal{B}')$*, or*
  (*iii*) $\tau = \alpha x \beta$*, where* $\alpha, \beta \in \kappa(\mathcal{B} \setminus \mathcal{B}')$*.*

  In addition,
  - *for each* $\tau = \alpha A^{(\pm 1)} \beta \in I(\sigma)$ *there is* $\tau' = \alpha A^{(\pm 1)} \beta' \in I(\sigma)$ *such that* $\beta' \alpha \leq 1$ *or there is* $\tau' = \alpha' A^{(\pm 1)} \beta \in I(\sigma)$ *such that* $\beta \alpha' \leq 1$*,*[6] *and*
  - *if* $I(\sigma)$ *contains a category of the form* (*i*)*, then it contains no category of the form* (*ii*)*,*[7] *and we shall say that* σ *is of type* (*i*) *or type* (*ii*)*, respectively.*

– $\Delta \subset \kappa(\mathcal{B} \setminus \mathcal{B}')$ *is a finite set of* distinguished *categories.*

The language generated by G is defined by

$$L(G) = \{w : \text{there exist } \tau \in I(w),\ \theta \in \mathcal{B}'^{+}, \text{ and } \delta \in \Delta \text{ such that } \tau[x := \theta] \leq \delta\},$$

where $\tau[x := \theta]$ is the result of simultaneous substitution of θ for x in τ.

We shall see in Section 5 that the membership problem for BAPGG languages is NP-complete. Therefore, for each positive integer K we associate with G the K-*restricted* language $L_K(G)$ generated by G that is defined as follows.

$$L_K(G) = \{w : \text{there exists } \tau \in I(w) \text{ having at most } K \text{ occurrences of } x, \text{ and there exist } \theta \in \mathcal{B}'^{+} \text{ and } \delta \in \Delta \text{ such that } \tau[x := \theta] \leq \delta\}.$$

That is, K is the number of times the BAPGG "may consult" its buffer. It is shown in Section 6 that the membership problem for restricted BAPGG languages can be solved in polynomial time.[8]

In what follows we establish some basic properties of the class of (restricted) BAPGG languages.

Theorem 2. *Buffer augmented pregroup grammars are at least as powerful as pregroup grammars.*

Proof. Let $G = \langle \Sigma, \mathcal{B}, \leq, I, \Delta \rangle$ be a PGG. Then $L(G) = L(G')$, where the BAPGG G' is defined by $G' = \langle \Sigma, \mathcal{B}, \leq, \emptyset, I, \Delta \rangle$. The proof is immediate by using the same assignment for both grammars..

Note that G' is indeed a BAPGG whose lexical category assignment satisfies clause (i) of the definition of I in Definition 3.

The same construction, obviously, works for the restricted languages.

[6] This condition is needed for the proof of the pumping lemma for BAPGG languages, but is not needed for for polynomial parsing of restricted BAPGG languages defined below.

[7] This constraint is need for polynomial parsing of restricted BAPGG languages, but is not needed for the proof of the pumping lemma for BAPGG languages.

[8] In other words, the membership problem for BAPGG languages is *fixed-parameter tractable.*

Remark 2. In fact, 1-restricted BAPGG languages are context-free. The proof is based on building the corresponding pushdown automaton. The automaton construction is similar to that in [3] (see also [5]) with the additional feature that, before reading x, the automaton can pop a number of symbols from $\{A^\ell : A \in \mathcal{B}'\}$ from the pushdown stack (using ϵ-moves) and then push there a number of symbols from $\mathcal{B}'$ (again using ϵ-moves).

We show next that characteristic mildly context-sensitive languages are generated by buffer augmented pregroup grammars. The grammar constructions are based on "push information" technique, where new terms are "pushed" into a number of positions in a category to cancel some of its other terms. Because the information being "pushed" is the same in all positions in the category, it can be used to compare the number of occurrences of a term in different positions. This can be thought of as a counterpart of *commutations* introduced in [4].

Multiple Agreement

Let $L_{ma} = \{a^n b^n c^n : n \geq 1\}$ and let $G_{ma} = \langle \Sigma, \mathcal{B}, =, \mathcal{B}', I, \Delta \rangle$, where

- $\Sigma = \{a, b, c\}$,
- $\mathcal{B} = \{A, P, T, U\}$,
- $\mathcal{B}' = \{A\}$,
- I is defined by
  - $I(a) = \{A^\ell, xT\}$,
  - $I(b) = \{T^r A^\ell T, T^r xU\}$, and
  - $I(c) = \{U^r A^\ell U, U^r xP\}$,

  and
- $\Delta = \{P\}$.

Then

$$L(G_{ma}) = L_3(G_{ma}) = L_{ma}.^9$$

For example, $aaabbbccc \in L_{ma}$ can be derived as follows. The lexical category assignment is

$$\overbrace{A^\ell}^{a}\ \overbrace{A^\ell}^{a}\ \overbrace{xT}^{a}\ \overbrace{T^r A^\ell T}^{b}\ \overbrace{T^r A^\ell T}^{b}\ \overbrace{T^r xU}^{b}\ \overbrace{U^r A^\ell U}^{c}\ \overbrace{U^r A^\ell U}^{c}\ \overbrace{U^r xP}^{c} \tag{2}$$

and, substituting $\theta = AA\,(\in \mathcal{B}'^+)$ for x, we obtain

$$\begin{aligned} &A^\ell A^\ell \theta \underline{TT^r} A^\ell \underline{TT^r} A^\ell \underline{TT^r} \theta \underline{UU^r} A^\ell \underline{UU^r} A^\ell \underline{UU^r} \theta P \\ &\qquad \leq A^\ell A^\ell \theta A^\ell A^\ell \theta A^\ell A^\ell \theta P \\ &\qquad = A^\ell A^\ell AAA^\ell A^\ell AAA^\ell A^\ell AAP \\ &\qquad \leq \underline{A^\ell A^\ell AAA^\ell A^\ell AAA^\ell A^\ell AA}P \\ &\qquad \leq P. \end{aligned}$$

The lexical category assignment (2) naturally extends on all elements of L_{ma}, implying $L_{ma} \subseteq L(G_{ma})$.

---

[9] Example 2 in the next section shows that L_{ma} is not a 2-restricted BAPGG language.

For the proof of the converse inclusion $L(G_{ma}) \subseteq L_{ma}$, let $w \in \Sigma^+$ be such that for some $\tau \in I(w)$[10] there exists a substitution $\theta \in \mathcal{B}'^+$ for which $\tau[x := \theta] \leq P$. It follows from the definition of I (and Corollary 1, of course) that $w = a^i b^j c^k$ and τ is of the form $\alpha x T T^r \beta x U U^r \beta x P$, where $\alpha = (A^\ell)^{i-1}$, $\beta = (T^r A^\ell T)^{j-1}$, and $\beta = (U^r A^\ell U)^{k-1}$. Thus $\theta = A^{i-1}$ $(= A^{j-1} = A^{k-1})$ and the desired inclusion follows.

Crossed Dependencies

Let $L_{cd} = \{a^n b^m c^n d^m : m, n \geq 1\}$ and let $G_{cd} = \langle \Sigma, \mathcal{B}, =, \mathcal{B}', I, \Delta \rangle$, where

- $\Sigma = \{a, b, c, d\}$,
- $\mathcal{B} = \{A, B, P, T, U, V\}$,
- $\mathcal{B}' = \{B\}$,
- I is defined by
  - $I(a) = \{A^\ell, A^\ell T\}$,
  - $I(b) = \{T^r B^\ell T, T^r x U\}$,
  - $I(c) = \{U^r A U, U^r A V\}$, and
  - $I(d) = \{V^r B^\ell V, V^r x P\}$,

  and
- $\Delta = \{P\}$.

Then

$$L(G_{cd}) = L_2(G_{cd}) = L_{cd}.$$

For example, $aabbbccddd \in L_{cd}$ can be derived as follows. The lexical category assignment is

$$\overbrace{A^\ell}^{a}\ \overbrace{A^\ell T}^{a}\ \overbrace{T^r B^\ell T}^{b}\ \overbrace{T^r B^\ell T}^{b}\ \overbrace{T^r x U}^{b}\ \overbrace{U^r A U}^{c}\ \overbrace{U A V^\ell}^{c}\ \overbrace{V^r B^\ell V}^{d}\ \overbrace{V B^\ell V}^{d}\ \overbrace{V^r x P}^{d}$$

and, substituting $\theta = BB$ $(\in \mathcal{B}'^+)$ for x, we obtain

$$\begin{aligned} A^\ell A^\ell \underline{TT^r} B^\ell \underline{TT^r} B^\ell \underline{TT^r} \theta \underline{UU^r} A \underline{UU^r} A \underline{VV^r} B^\ell \underline{VV^r} B^\ell \underline{VV^r} \theta P & \\ \leq A^\ell A^\ell B^\ell B^\ell \theta A A B^\ell B^\ell \theta P & \\ = A^\ell A^\ell \underline{B^\ell B^\ell B B} A A \underline{B^\ell B^\ell B B} P & \\ \leq \underline{A^\ell A^\ell A A} P & \\ \leq P. & \end{aligned}$$

The proof of the equality $L(G_{cd}) = L_{cd}$ is similar to that of the equality $L(G_{ma}) = L_{ma}$ and is omitted.

Reduplication

Let $\Sigma = \{a, b\}$, $L_{rd} = \{ww : w \in \Sigma^+\}$, and let $G_{rd} = \langle \Sigma, \mathcal{B}, =, \mathcal{B}', I, \Delta \rangle$, where

- $\Sigma = \{a, b\}$,
- $\mathcal{B} = \{A, B, P, T\}$,
- $\mathcal{B}' = \{A, B\}$,
- I is defined by

[10] Of course, by $I(\sigma_1 \cdots \sigma_n)$ we mean $\{\tau_1 \cdots \tau_n : \tau_i \in I(\sigma_i),\ i = 1, \ldots, n\}$.

  - $I(a) = \{A^\ell, A^\ell xT, T^r A^\ell T, T^r A^\ell xP\}$ and
  - $I(b) = \{B^\ell, B^\ell xT, T^r B^\ell T, T^r B^\ell xP\}$,

  and
- $\Delta = \{P\}$.

Then

$$L(G_{rd}) = L_2(G_{rd}) = L_{rd}.$$

For example, $abbbabb \in L_{rd}$ can be derived as follows. The lexical category assignment is

$$\overbrace{A^\ell}^{a}\ \overbrace{B^\ell}^{b}\ \overbrace{B^\ell xT}^{b}\,\overbrace{T^r A^\ell T}^{a}\,\overbrace{T^r B^\ell T}^{b}\,\overbrace{T^r B^\ell xP}^{b}$$

and, substituting $\theta = BBA\,(\in \mathcal{B}'^{+})$ for x, we obtain

$$\begin{aligned} A^\ell B^\ell B^\ell \theta \underline{TT^r} A^\ell \underline{TT^r} B^\ell \underline{TT^r} B^\ell \theta P &\le A^\ell B^\ell B^\ell \theta A^\ell B^\ell B^\ell \theta P \\ &= \underline{A^\ell B^\ell B^\ell BBAA^\ell B^\ell B^\ell BBA}P \\ &\le P. \end{aligned}$$

We omit the proof of the equality $L(G_{rd}) = L_{rd}$.

4 Pumping Lemma for (Restricted) BAPGG Languages

In this section we present the following version of pumping lemma for (restricted) BAPGG languages.

Theorem 3. *For each BAPGG language L there exist a positive integer N such that every $w \in L$, $|w| \ge N$, can be partitioned as $w = u_1 v_1 u_2 v_2 \cdots u_m v_m u_{m+1}$, where*

- *$m \ge 1$,*
- *$|v_1|, |v_2| \ge 1$, if $m \le 2$, and $|v_1| = \cdots = |v_m| = 1$, if $m \ge 3$, and*
- *for all $i \ge 1$*
$$u_1 v_1^i u_2 v_2^i \cdots u_m v_m^i u_{m+1} \in L.^{11}$$

Proof. Let $L = L(G)$ for a BAPGG $G = \langle \Sigma, \mathcal{B}, \le, \mathcal{B}', I, \Delta \rangle$ and let $G' = \langle \Sigma, \mathcal{B}, \le, \emptyset, I, \Delta \rangle$. Then $L(G')$ is a context-free language, because, in this case, we may restrict I to the categories of the form (i), only. We choose N to be a pumping lemma constant for $L(G')$.

Let $w = \sigma_1 \cdots \sigma_n \in L$ be such that $|w| \ge N$. If $w \in L(G')$, then the theorem follows from the ordinary pumping lemma for context-free languages.

Otherwise, i.e., $w \notin G'$, every $\tau \in I(w)$ such that $\tau[x := \theta] \le \delta$, for some $\theta \in \mathcal{B}'^{+}$ and some $\delta \in \Delta$, must contain an occurrence of x.

Given such τ, let m be the number of occurrences of x in it and let $\theta = A\theta'$, where $A \in \mathcal{B}'$ and $\theta' \in \mathcal{B}'^{*}$. Since $\tau[x := \theta] \le \delta$ and $\delta \in \kappa(\mathcal{B} \setminus \mathcal{B}')$, the first occurrence of A in the jth θ in $\tau[x := A\theta']$, $j = 1, \ldots, m$, (from left to right) is cancelled by $A^{(\pm 1)}$ that comes from some $\alpha_j t_j \beta_j \in I(\sigma_{k_j})$ of type (ii), where $t_j = A^{(\pm 1)}$.

[11] Note the difference with the ordinary pumping lemma, where $i \ge 0$.

We let

- $v_j = \sigma_{k_j}$, $j = 1, \ldots, m$,
- $u_1 = \sigma_1 \cdots \sigma_{k_1 - 1}$,
- $u_j = \sigma_{k_{j-1}+1} \cdots \sigma_{k_j - 1}$, $j = 2, \ldots, m$, and
- $u_{m+1} = \sigma_{k_m+1} \cdots \sigma_n$,

and the lexical category assignment to the symbols in $u_1 v_1^i u_2 v_2^i \cdots u_m v_m^i u_{m+1}$ and the substitution θ for x are as follows.

- The lexical category assignment to the elements of Σ occurring in the u_js is the original one.
- The i copies of $v_j = \sigma_{k_j}$, $j = 1, \ldots, m$, are assigned
$$\underbrace{\overbrace{\alpha_j t_j \beta'_j}^{\sigma_{k_j}} \cdots \overbrace{\alpha_j t_j \beta'_j}^{\sigma_{k_j}}}_{i-1 \text{ times}} \overbrace{\alpha_j t_j \beta_j}^{\sigma_{k_j}},$$
  if there is $\alpha_j t_j \beta'_j \in I(\sigma_{k_j})$ such that $\beta'_j \alpha \leq 1$, or are assigned
$$\overbrace{\alpha_j t_j \beta_j}^{\sigma_{k_j}} \underbrace{\overbrace{\alpha'_j t_j \beta_j}^{\sigma_{k_j}} \cdots \overbrace{\alpha'_j t_j \beta_j}^{\sigma_{k_j}}}_{i-1 \text{ times}},$$
  if there is $\alpha'_j t_j \beta_j \in I(\sigma_{k_j})$ such that $\beta_j \alpha' \leq 1$.
- The substitution for x is $A^i \theta'$.

Then, in the former case,

$$\cdots (\alpha_j t_j \beta'_j)^{i-1} \alpha_j t_j \beta_j \cdots \leq \cdots \alpha_j t_j^i \beta_j \cdots ,$$

and, in the latter case,

$$\cdots \alpha_j t_j \beta_j (\alpha'_j t_j \beta_j)^{i-1} \cdots \leq \cdots \alpha_j t_j^i \beta_j \cdots .$$

That is, t_j^i cancels A^i in the substitution $A^i \theta'$, whereas all other cancellations are as in τ. Therefore,

$$u_1 v_1^i u_2 v_2^i \cdots u_m v_m^i u_{m+1} \in L.$$

Example 2. It immediately follows from Theorem 3 that the multiple agreement language L_{ma} is not a 2-restricted BAPGG language.

We conclude this section with the following corollary to Theorem 3.

Corollary 2. *BAPGG languages are semi-linear.*

5 Complexity of BAPGG Languages

In this section we show that the membership problem for BAPGG languages is NP-complete.

Theorem 4. *The membership problem for BAPGG languages is in NP.*

Proof. Let $G = \langle \Sigma, \mathcal{B}, \leq, \mathcal{B}', I, \Delta \rangle$ be a BAPGG and let

$$M = \max\{|I(\sigma)| : \sigma \in \Sigma\}.$$

Let $w \in L(G)$ and let $\tau \in I(w)$, $\theta \in \mathcal{B}'^{+}$, and $\delta \in \Delta$ be such that

$$\tau[x := \theta] \leq \delta.$$

Since θ is in $\mathcal{B}'^{+}$, no term occurring in a copy of it can be cancelled by a term occurring in another copy. Therefore,

$$|\theta| \leq M|w| + \max\{|\delta| : \delta \in \Delta\}.$$

That is, an appropriate lexical category assignment $\tau \in I(w)$, the substitution θ, and an appropriate $\delta \in \Delta$ can be "guessed" in an $O(|w|)$ time.

Theorem 5. *The membership problem for BAPGG languages is NP-hard.*

The proof of Theorem 5 is by a polynomial reduction from the 3-SAT problem. Namely, we shall construct a BAPGG G_{3-SAT} and define a polynomial time encoding of 3-CNF formulas (i.e., conjunctions of disjunctions of three literals) such that the encoding $[\varphi]$ of a 3-CNF formula φ is in $L(G_{3-SAT})$ if and only if φ is satisfiable.

The language of G_{3-SAT} is over the alphabet

$$\Sigma = \{b, l, r, \#, \$, @, \%, \} \cup \{t_m, t'_m, t''_m\}_{m=0,\ldots,7}$$

and 3-CNF formulas are encoded as follows.

Let x_i, x_j, and x_k be pairwise distinct variables and let $L_i \in \{x_i, \overline{x_i}\}$, $L_j \in \{x_j, \overline{x_j}\}$, and $L_k \in \{x_k, \overline{x_k}\}$ be literals. With the clause $\boldsymbol{c} = L_i \vee L_j \vee L_k$ we associate its *type* $t(\boldsymbol{c}) = t_m t'_m t''_m$, $m = 0, \ldots, 7$, that is defined as follows.

- $t(x_i \vee x_j \vee x_k) = t_0 t'_0 t''_0$.
- $t(x_i \vee x_j \vee \overline{x_k}) = t_1 t'_1 t''_1$.
- $t(x_i \vee \overline{x_j} \vee x_k) = t_2 t'_2 t''_2$.
- $t(x_i \vee \overline{x_j} \vee \overline{x_k}) = t_3 t'_3 t''_3$.
- $t(\overline{x_i} \vee x_j \vee x_k) = t_4 t'_4 t''_4$.
- $t(\overline{x_i} \vee x_j \vee \overline{x_k}) = t_5 t'_5 t''_5$.
- $t(\overline{x_i} \vee \overline{x_j} \vee x_k) = t_6 t'_6 t''_6$.
- $t(\overline{x_i} \vee \overline{x_j} \vee \overline{x_k}) = t_7 t'_7 t''_7$.

Let $\varphi = \bigwedge_{q=1}^{p} \boldsymbol{c}_q$ and let $x_1, \ldots, x_n$ be all variables that occur in ϕ. Then the encoding $[\boldsymbol{c}_q]$ of clause $\boldsymbol{c}_q = L_i \vee L_j \vee L_k$ that occurs in φ is

$$[\boldsymbol{c}_q] = \# l^{i-1} b r^{n-i} l^{j-1} b r^{n-j} l^{k-1} b r^{n-k} \$ t''_m t'_m t_m @ \%,$$

where $t(\boldsymbol{c}_q) = t_m t'_m t''_m$.

Remark 3. In the above encoding, b is the "buffer symbol" to be substituted with the content of the buffer; the pairs of words (l^{i-1}, r^{n-i}), (l^{j-1}, r^{n-j}), and (l^{k-1}, r^{n-k}) indicate the literal variable (whose truth assignment will be cut from the "truth assignment word $v_1 \cdots v_n \in \{\bot, \top\}^n$ provided by the buffer); and the type $t(\boldsymbol{c}_q) = t_m t'_m t''_m$ determines the type of the literals in the clause $\boldsymbol{c}_q$. The delimiters #, \$, @, and % are needed for a technical (cancellation) purpose that will become clear in the sequel.

Now, the encoding $[\varphi]$ of φ over Σ is

$$[\phi] = [\boldsymbol{c}_1] \cdots [\boldsymbol{c}_p].$$

Let $L_{3-SAT} = \{[\phi] : \phi \in 3-SAT\}$ and let $G_{3-SAT} = \langle \Sigma, \mathcal{B}, =, \mathcal{B}', I, \{1\} \rangle$, where the components of G_{3-SAT} are defined below.

$$\mathcal{B} = \{A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, S, \bot, \top\}.$$

Intuitively, A_ms, $m = 0, \ldots, 7$, correspond to truth assignments as follows.

$$\begin{aligned} A_0 &\leftrightarrow (\bot, \bot, \bot). \\ A_1 &\leftrightarrow (\bot, \bot, \top). \\ A_2 &\leftrightarrow (\bot, \top, \bot). \\ A_3 &\leftrightarrow (\bot, \top, \top). \\ A_4 &\leftrightarrow (\top, \bot, \bot). \\ A_5 &\leftrightarrow (\top, \bot, \top). \\ A_6 &\leftrightarrow (\top, \top, \bot). \\ A_7 &\leftrightarrow (\top, \top, \top). \end{aligned}$$

In particular, A_m corresponds to the only truth assignment that does not satisfy a clause of type $t_m t'_m t''_m$, $m = 0, \ldots, 7$.

Next, $\mathcal{B}' = \{\bot, \top\}$ and I is defined as follows.

- $I(b) = \{x\}$.
- $I(l) = \{\bot^\ell, \top^\ell\}$.
- $I(r) = \{\bot^r, \top^r\}$.
- $I(\#) = \{S\}$.
- $I(\$) = \{A_0^\ell, A_1^\ell, A_2^\ell, A_3^\ell, A_4^\ell, A_5^\ell, A_6^\ell, A_7^\ell\}$.
- $I(@) = \{A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7\}$. That is, the lexical category assignment to \$ is supposed to be canceled by the lexical category assignment to @, see Remark 3.
- $I(\%) = \{S^r\}$. That is, the lexical category assignment to # is supposed to be canceled by the lexical category assignment to %, see Remark 3.
- $I(t_0) = \{A_1 \bot^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \bot^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t'_0) = \{A_1 \bot^r A_1^\ell, A_2 \top^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \bot^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t''_0) = \{A_1 \top^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \bot^r A_3^\ell, A_4 \top^r A_4^\ell A_5 \top^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_1) = \{A_0 \bot^r A_0^\ell, A_2 \bot^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \bot^r A_6^\ell, A_7 \top^r A_7^\ell\}$.

- $I(t_1') = \{A_0 \bot^r A_0^\ell, A_2 \top^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \bot^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_1'') = \{A_0 \bot^r A_0^\ell, A_2 \bot^r A_2^\ell, A_3 \bot^r A_3^\ell, A_4 \top^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_2) = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \bot^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_2') = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \bot^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_2'') = \{A_0 \bot^r A_0^\ell, A_1 \top^r A_1^\ell, A_3 \bot^r A_3^\ell, A_4 \top^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_3) = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \bot^r A_2^\ell, A_4 \bot^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \bot^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_3') = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \top^r A_2^\ell, A_4 \bot^r A_4^\ell, A_5 \bot^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_3'') = \{A_0 \bot^r A_0^\ell, A_1 \top^r A_1^\ell, A_2 \bot^r A_2^\ell, A_4 \top^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_4) = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \top^r A_3^\ell, A_5 \top^r A_5^\ell, A_6 \bot^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_4') = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \top^r A_2^\ell, A_3 \top^r A_3^\ell, A_5 \bot^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_4'') = \{A_0 \bot^r A_0^\ell, A_1 \top^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \bot^r A_3^\ell, A_5 \top^r A_5^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_5) = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_6 \bot^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_5') = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \top^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_5'') = \{A_0 \bot^r A_0^\ell, A_1 \top^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \bot^r A_3^\ell, A_4 \top^r A_4^\ell, A_6 \top^r A_6^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_6) = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \top^r A_5^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_6') = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \top^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \bot^r A_5^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_6'') = \{A_0 \bot^r A_0^\ell, A_1 \top^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \bot^r A_3^\ell, A_4 \top^r A_4^\ell, A_5 \top^r A_5^\ell, A_7 \top^r A_7^\ell\}$.
- $I(t_7) = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \bot^r A_6^\ell\}$.
- $I(t_7') = \{A_0 \bot^r A_0^\ell, A_1 \bot^r A_1^\ell, A_2 \top^r A_2^\ell, A_3 \top^r A_3^\ell, A_4 \bot^r A_4^\ell, A_5 \bot^r A_5^\ell, A_6 \top^r A_6^\ell\}$.
- $I(t_7'') = \{A_0 \bot^r A_0^\ell, A_1 \top^r A_1^\ell, A_2 \bot^r A_2^\ell, A_3 \bot^r A_3^\ell, A_4 \top^r A_4^\ell, A_5 \top^r A_5^\ell, A_6 \top^r A_6^\ell\}$.

Proof of the Inclusion $L_{3-SAT} \subseteq L(G_{3-SAT})$

The proof of the inclusion $L_{3-SAT} \subseteq L(G_{3-SAT})$ is based on Lemmas 1 and 2 below.

Lemma 1. *Let $\boldsymbol{c} = L_i \vee L_j \vee L_k$ be a 3-CNF clause of type m, $m = 0, \ldots, 7$, and let $v_i, v_j, v_k = \bot, \top$ be such that (v_i, v_j, v_k) satisfies $\boldsymbol{c}$. Then there exists*

$$A \in \{A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7\} \setminus \{A_m\}$$

such that $Av_i^r A^\ell \in I(t_m)$, $A{v_j}^r A^\ell \in I(t_m')$, and $A{v_k}^r A^\ell \in I((t_m'')$.

Proof. The lemma follows immediately from the definition of $I(t_m)$, $I(t_m')$, and $I(t_m'')$. For example, if $(v, v_j, v_k) = (\top, \top, \bot)$, then $A = A_6$.

Lemma 2. *Let $\boldsymbol{c} = L_i \vee L_j \vee L_k$ be a 3-CNF clause and let $V = (v_1, \cdots, v_n) \in \{\bot, \top\}^n$ be an "assignment vector" such that $\boldsymbol{c}|_V = \top$. Then there exists $\tau \in I([\boldsymbol{c}])$ such that*

$$\tau[x := v_1 \cdots v_n] \leq 1.$$

Proof. It follows from $\boldsymbol{c}|_V = \top$ that

$$\boldsymbol{c}(v_i, v_j, v_k) = \top. \tag{3}$$

Let $\boldsymbol{c}$ be of type m. Then,

$$[\boldsymbol{c}] = \#l^{i-1}br^{n-i}l^{j-1}br^{n-j}l^{k-1}br^{n-k}\$t_m''t_m't_m@\% =$$

$$\#\underbrace{l\cdots l}_{i-1}b\underbrace{r\cdots r}_{n-i}\ \underbrace{l\cdots l}_{j-1}b\underbrace{r\cdots r}_{n-j}\ \underbrace{l\cdots l}_{k-1}b\underbrace{r\cdots r}_{n-k}\ \$t_m''t_m't_m@\%,$$

and desired lexical category assingment $\tau \in I([\boldsymbol{c}])$ is defined by

$$\tau = \overbrace{S}^{\#}\,\overbrace{v_{i-1}^{\ell}}^{l}\cdots\overbrace{v_1^{\ell}}^{l}\,\overbrace{x}^{b}\,\overbrace{v_n^{r}}^{r}\cdots\overbrace{v_{n-i}^{r}}^{r}$$
$$\overbrace{v_{j-1}^{\ell}}^{l}\cdots\overbrace{v_1^{\ell}}^{l}\,\overbrace{x}^{b}\,\overbrace{v_n^{r}}^{r}\cdots\overbrace{v_{n-j}^{r}}^{r}$$
$$\overbrace{v_{k-1}^{\ell}}^{l}\cdots\overbrace{v_1^{\ell}}^{l}\,\overbrace{x}^{b}\,\overbrace{v_n^{r}}^{r}\cdots\overbrace{v_{n-k}^{r}}^{r}\,\overbrace{A^{\ell}}^{\$}\,\overbrace{Av_k^rA^{\ell}}^{t''_m}\,\overbrace{Av_j^rA^{\ell}}^{t'_m}\,\overbrace{Av_i^rA^{\ell}}^{t_m}\,\overbrace{A}^{@}\,\overbrace{S^r}^{\%},$$

where A is provided by (3) and Lemma 1. Therefore,

$$\begin{aligned}\tau[x := v_1\cdots v_n] &= S v_{i-1}^{\ell}\cdots v_1^{\ell}v_1\cdots v_nv_n^r\cdots v_{n-i}^r\\ &\quad v_{j-1}^{\ell}\cdots v_1^{\ell}v_1\cdots v_nv_n^r\cdots v_{n-j}^r\\ &\quad v_{k-1}^{\ell}\cdots v_1^{\ell}v_1\cdots v_nv_n^r\cdots v_{n-k}^rA^{\ell}Av_k^rA^{\ell}Av_j^rA^{\ell}Av_i^rA^{\ell}AS^r \\ &\le S v_iv_jv_k\ v_k^rv_j^rv_i^rS^r\\ &\le 1.\end{aligned} \tag{4}$$

Now, let $\varphi = \bigwedge_{q=1}^{p} \boldsymbol{c}_q$ and let $V = (v_1, \cdots, v_n) \in \{\bot, \top\}^n$ be an "assignment vector" such that $\boldsymbol{c}|_V = \top$. By Lemma 2, for every $q = 1, \ldots, p$,

$$[\boldsymbol{c}_q][x := v_1\cdots v_n] \le 1.$$

Therefore,

$$\begin{aligned}[\phi][x := v_1\cdots v_n] &= ([\boldsymbol{c}_1]\cdots[\boldsymbol{c}_p])[x := v_1\cdots v_n]\\ &= ([\boldsymbol{c}_1][x := v_1\cdots v_n])\cdots([\boldsymbol{c}_p][x := v_1\cdots v_n])\\ &\le 1\end{aligned}$$

and the desired inclusion follows.

Proof of the Inclusion $L(G_{3-SAT}) \cap \{[\varphi] : \varphi \in$ 3-CNF$\} \subseteq L_{3-SAT}$

For the proof of the inclusion

$$L(G_{3-SAT}) \cap \{[\varphi] : \varphi \in \text{3-CNF}\} \subseteq L_{3-SAT} \tag{5}$$

we shall need the following lemma.

Lemma 3. *Let* $\mathbf{c}$ *be a 3-CNF clause,* $\tau \in I([\mathbf{c}])$*, and let* $V = (v_1, \ldots, v_n) \in \{\bot, \top\}^n$ *be such that*

$$\tau[x := v_1\cdots v_n] \le 1.$$

then $\mathbf{c}|_V = \top$*.*

Proof. The lemma follows immediately from (4) and the definition of lexical category assignment I to t_m, t'_m, and t''_m, $m = 0, \ldots, 7$.

Now let $\varphi = \bigwedge_{q=1}^{p} \boldsymbol{c}_q$ be a 3-CNF formula and let

$$[\phi] = \underbrace{\#w_1\%}_{\boldsymbol{c}_1} \cdots \underbrace{\#w_p\%}_{\boldsymbol{c}_p},$$

where $[\boldsymbol{c}_q] = \#w_q\%$, $q = 1, \ldots, p$.

Let $V = (v_1, \ldots, v_n) \in \{\bot, \top\}^n$ and let $\boldsymbol{c} \in I([\phi])$ be such that

$$\boldsymbol{c}[x := v_1 \cdots v_n] \leq 1.$$

For the proof of the inclusion (5) we have to show that V satisfies ϕ.

We have

$$\tau[x := v_1 \cdots v_n] \;=\; S\tau'_1[x := v_1 \cdots v_n]S^r \;\cdots\; S\tau'_p[x := v_1 \cdots v_n]S^r$$

for appropriate τ'_qs in $I(w_q)$, $q = 1, \ldots, p$. Then the qth S^r (from left to right) must be canceled from the left by the qth S, $q = 1, \ldots, p$. Therefore, $\tau'_q[x := v_1 \cdots v_n] \leq 1$, implying

$$\tau_q[x := v_1 \cdots v_n] = S\tau'_q[x := v_1 \cdots v_n]S^r \leq 1,$$

$q = 1, \ldots, p$. Since $\tau_q \in I([\boldsymbol{c}_q])$, by Lemma 3, V satisfies all clauses of φ and the proof is complete.

6 Complexity of Restricted BAPGG Languages

In this section we show that the membership problem for restricted BAPGG languages can be decided in polynomial time.

Theorem 6. *The membership problem for restricted BAPGG languages is in P.*

The proof of Theorem 6 is based on a sequence of reductions described below.

Let $G = \langle \Sigma, \mathcal{B}, \leq, \mathcal{B}', I, \Delta \rangle$ be a BAPGG, $K \geq 1$, and let $w = \sigma_1 \cdots \sigma_n \in \Sigma$. By definition, $w \in L_K(G)$ if and only if there exist $\tau_i \in I(\sigma_i)$, $i = 1, \ldots, n$, such that $\tau_1 \cdots \tau_n$ has at most K occurrences of x and there exist $\theta \in \mathcal{B}'^+$ and $\delta \in \Delta$ such that

$$(\tau_1 \cdots \tau_n)[x := \theta] \leq \delta.$$

Therefore, there exist positive integers $i'_1, \ldots, i'_k$, $i_1, \ldots, i_k$, and $i''_1, \ldots, i''_k$, where $k \leq K$, such that

$$1 \leq i'_1 \leq i_1 \leq i''_1 < \cdots < i'_j \leq i_j \leq i''_j < \cdots < i_k \leq i''_k \leq n; \tag{6}$$

for each $j = 1, \ldots, k$, $\tau_{i'_j} = \alpha_{i'_j} A^{\ell}_{i'_j} \beta_{i'_j}$ and $\tau_{i''_j} = \alpha_{i''_j} A^{r}_{i''_j} \beta_{i''_j}$ are categories of the form (ii), and $\tau_{i_j} = \alpha_{i_j} x \beta_{i_j}$ is a category of the form (iii);

$$(A^{\ell}_{i'_j} \beta_{i'_j} \tau_{i'_j+1} \cdots \tau_{i_j-1} \alpha_{i_j} x \beta_{i_j} \tau_{i_j+1} \cdots \tau_{i''_j-1} \alpha_{i''_j} A^{r}_{i''_j})[x := \theta] \leq 1;\text{[12]} \tag{7}$$

and

[12] That is, $A^{\ell}_{i'_j}$ and $A^{r}_{i''_j}$ $j = 1, \ldots, k$, cancel the rightmost and the leftmost symbols of θ, respectively.

$$\tau_1 \cdots \tau_{i'_1-1}\alpha_{i'_1}\beta_{i''_1}\tau_{i''_1+1} \cdots \tau_{i'_j-1}\alpha_{i'_j}\beta_{i''_j}\tau_{i''_j+1} \cdots \tau_{i'_k-1}\alpha_{i'_k}\beta_{i''_k}\tau_{i''_k+1} \cdots \tau_n \le \delta.^{13} \quad (8)$$

Thus, for all $k = 1, \ldots, K$, all sets of positive integers

$$\{i'_1, \ldots, i'_k\} \cup \{i_1, \ldots, i_k\} \cup \{i''_1, \ldots, i''_k\}$$

satisfying (6),[14] all assignments $\tau_{i'_j} = \alpha_{i'_j} A^{\ell}_{i'_j} \beta_{i'_j}$ and $\tau_{i''_j} = \alpha_{i''_j} A^{r}_{i''_j} \beta_{i''_j}$ of the from (ii), all assignments $\tau_{i_j} = \alpha^{r}_{i_j} x \beta_{i_j}$ of the from (iii), and all $\delta \in \Delta$ we shall look for $\theta \in \mathcal{B}'^{+}$ and categories $\tau_i \in I(\sigma_i)$,

$$i \in \{1, \ldots, n\} \setminus (\{i'_1, \ldots, i'_k\} \cup \{i_1, \ldots, i_k\} \cup \{i''_1, \ldots, i''_k\}),$$

such that (8) and

$$\begin{cases} (A^{\ell}_{i'_1}\beta_{i'_1}\tau_{i'_1+1} \cdots \tau_{i_1-1}\alpha_{i_1}x\beta_{i_1}\tau_{i_1+1} \cdots \tau_{i''_1-1}\alpha_{i''_1}A^{r}_{i''_1})[x := \theta] & \le 1 \\ (A^{\ell}_{i'_2}\beta_{i'_2}\tau_{i'_2+1} \cdots \tau_{i_2-1}\alpha_{i_2}x\beta_{i_2}\tau_{i_2+1} \cdots \tau_{i''_2-1}\alpha_{i''_2}A^{r}_{i''_2})[x := \theta] & \le 1 \\ & \vdots \\ (A^{\ell}_{i'_k}\beta_{i'_k}\tau_{i'_k+1} \cdots \tau_{i_k-1}\alpha_{i_k}x\beta_{i_k}\tau_{i_k+1} \cdots \tau_{i''_k-1}\alpha_{i''_k}A^{r}_{i''_k})[x := \theta] & \le 1 \end{cases} . \quad (9)$$

That is, (9) consists of k inequations (7): one inequation for each $j = 1, \ldots, k$.

Obviously, the number of such possible pairs (8) and (9) is bounded by a polynomial in n (whose degree is a function of K).

First we shall show that (8) can be solved in polynomial time. Let

$$\widehat{\Sigma} = \{\widehat{\sigma_i} : i \in \{i'_1, \ldots, i'_k\} \cup \{i''_1, \ldots, i''_k\}\}$$

be a disjoint copy of

$$\{\sigma_{i'_1}, \ldots, \sigma_{i'_k}\} \cup \{\sigma_{i''_1}, \ldots, \sigma_{i''_k}\}.$$

Consider the PGG $\widehat{G} = \langle \Sigma \cup \widehat{\Sigma}, \mathcal{B}, \le, \widehat{I}, \delta \rangle$, where $\widehat{I}$ is defined as follows.

$$\widehat{I}(\sigma) = \begin{cases} I(\sigma), \text{ if } \sigma \in \Sigma \\ \alpha_{i'_j}, \ \text{ if } \sigma = \widehat{\sigma_{i'_j}}, \ j = 1, \ldots, k \\ \beta_{i''_j}, \ \text{ if } \sigma = \widehat{\sigma_{i''_j}}, \ j = 1, \ldots, k \end{cases} .$$

Then there is a lexical category assignment for σ_i,

$$i \in \{1, \ldots, i'_1 - 1\} \cup \bigcup_{j=1}^{k-1} \{i''_j + 1, \ldots, i'_{j+1} - 1\} \cup \{i''_k + 1, \ldots, n\}$$

satisfying (8) if and only if

$$\sigma_1 \cdots \sigma_{i'_1-1}\widehat{\sigma_{i'_1}}\,\widehat{\sigma_{i''_1}}\sigma_{i''_1+1} \cdots \sigma_{i'_k-1}\widehat{\sigma_{i'_k}}\,\widehat{\sigma_{i''_k}} \cdots \sigma_n \in L(\widehat{G}).$$

[13] Note that all τ_is occurring in (8) are of the form (i).

[14] Actually, in (9) we assume that all inequalities in (6) are strict. It will be clear in the sequel how to treat the case of non-strict inequalities.

The latter membership can be tested in polynomial time, because, by Theorem 1, $L(\widehat{G})$ is context-free.

We shall show next how to solve (9) in polynomial time. First we observe that, by the definition of assignment I, for each solution of (9) and each inequation j, $j = 1, \ldots, k$, the following holds.

Each category of the from (ii) to the left of x is of the from $\alpha A^{\ell}\beta$ and each category of the from (ii) to the right of x is of the from $\alpha A^{r}\beta$.

For our next observation we shall need the following notation. For a category $\tau = \alpha A^{r}\beta$ of the form (ii) we denote by $\widetilde{\tau}$ the category $\tau = \alpha A^{\ell}\beta$ and for a category τ of the form (i), $\widetilde{\tau}$ is τ itself. Then, (9) if and only if

$$\begin{cases} \beta_{i_1}\widetilde{\tau_{i_1+1}}\cdots\widetilde{\tau_{i''_1-1}}\alpha_{i''_1}A^{\ell}_{i''_1}A^{\ell}_{i'_1}\beta_{i'_1}\tau_{i'_1+1}\cdots\tau_{i_1-1}\alpha_{i_1}\theta \ \leq 1 \\ \beta_{i_2}\widetilde{\tau_{i_2+1}}\cdots\widetilde{\tau_{i''_2-1}}\alpha_{i''_2}A^{\ell}_{i''_2}A^{\ell}_{i'_2}\beta_{i'_2}\tau_{i'_2+1}\cdots\tau_{i_2-1}\alpha_{i_2}\theta \ \leq 1 \\ \vdots \\ \beta_{i_k}\widetilde{\tau_{i_k+1}}\cdots\widetilde{\tau_{i''_k-1}}\alpha_{i''_k}A^{\ell}_{i''_k}A^{\ell}_{i'_k}\beta_{i'_k}\tau_{i'_k+1}\cdots\tau_{i_k-1}\alpha_{i_k}\theta \leq 1 \end{cases} . \qquad (10)$$

That is, the left-hand side of the jth inequation in (10) can be thought of as a lexical assignment $\overline{I}$ over the alphabet

$$\overline{\Sigma} = \Sigma \cup \widetilde{\Sigma} \cup \Sigma' \cup \Sigma'',$$

where $\widetilde{\Sigma}$, Σ', Σ'', and $\overline{I}$ are defined as follows:

$$\widetilde{\Sigma} = \{\widetilde{\sigma} : \sigma \in \Sigma \text{ is of type } (ii)\}$$

is a disjoint copy of the subset of Σ consisting of all elements of Σ of type (ii),

$$\Sigma' = \{\sigma' : \sigma = \sigma_{i'_1}, \ldots, \sigma_{i'_k}\}$$

is a disjoint copy of $\{\sigma_{i'_1}, \ldots, \sigma_{i'_k}\}$,

$$\Sigma'' = \{\sigma'' : \sigma = \sigma_{i''_1}, \ldots, \sigma_{i''_k}\}$$

is a disjoint copy of $\{\sigma_{i''_1}, \ldots, \sigma_{i''_k}\}$, and

$$\overline{I}(\sigma) = \begin{cases} I(\sigma), & \text{if } \sigma \in \Sigma \\ \{\widetilde{\tau} : \tau \in I(\sigma)\}, & \text{if } \sigma \in \widetilde{\Sigma} \\ A^{\ell}_{i'_j}\beta_{i'_j}, & \text{if } \sigma = \sigma_{i'_j} \in \Sigma' \\ \alpha_{i''_j}A^{\ell}_{i''_j}, & \text{if } \sigma = \sigma_{i''_j} \in \Sigma'' \end{cases} .$$

The type of the elements of $\widetilde{\Sigma} \cup \Sigma' \cup \Sigma''$ is (ii).

Thus, we have reduced the membership problem for $L(G)$ to solving polynomially many systems of inequations described below.

Let $j = 1, \ldots, k$ and let $m_j = i''_j - i'_j$. We shall use the following renaming of the elements of Σ occurring in $w = \sigma_1 \cdots \sigma_n$:

$$\sigma_{j,i} = \begin{cases} \sigma_{i_j+i}, & \text{if } 1 \leq i < i''_j - i_j \text{ and } \sigma_{i_j+i} \text{ is of type } (i) \\ \widetilde{\sigma_{i_j+i}}, & \text{if } 1 \leq i < i''_j - i_j \text{ and } \sigma_{i_j+i} \text{ is of type } (ii) \\ \sigma_{i''_j}, & \text{if } i = i''_j - i_j \\ \sigma_{i'_j}, & \text{if } i = i''_j - i_j + 1 \\ \sigma_{i'_j+i-(i''_j-i_j+1)}, & \text{if } i''_j - i_j + 1 < i \leq m_j \end{cases} .$$

Then, we have to find assignment $\tau_{j,i} \in \overline{I}(\sigma_{j,i})$, $j = 1, \ldots, k$ and $i = 1, \ldots, m_j$, and $\theta \in \mathcal{B}'^{+}$ such that

$$\begin{cases} \beta_{i_1}\tau_{1,1}\cdots\tau_{1,m_1}\alpha_{i_1}\theta \leq 1 \\ \beta_{i_2}\tau_{2,1}\cdots\tau_{2,m_2}\alpha_{i_2}\theta \leq 1 \\ \qquad\vdots \\ \beta_{i_k}\tau_{k,1}\cdots\tau_{k,m_k}\alpha_{i_k}\theta \leq 1 \end{cases}. \tag{11}$$

That is, the systems of inequations (10) and (11) are related as follows:

$$\tau_{j,1} = \widetilde{\tau_{i_j+1}}, \ldots, \tau_{j,i''_j - i_j} = \alpha_{i''_j} A^{\ell}_{i''_j}, \tau_{j,i''_j - i_j + 1} = A^{\ell}_{i'_j}\beta_{i'_j} \ldots, \tau_{j,m_j} = \tau_{i_j - 1},$$

$j = 1, \ldots, k$. In particular, m_j is the number of elements of Σ corresponding to the jth inequation in (10).

Since $\theta \in \mathcal{B}'^{+}$, the symbols occurring in θ can be cancelled only by the pregroup elements of the form A^{ℓ}, and such elements can occur only in assignments to the elements of Σ of type (ii). Therefore, the number of the elements of Σ of type (ii) occurring in each $\sigma_{j,1}\cdots\sigma_{j,m_j}$, $j = 1, \ldots, k$, is the same, say $m = |\theta|$.

Let $\theta = A_1 \cdots A_m$ and let $\sigma_{j,l_{j,i}}$ be the ith occurrence (from the left) of the elements of Σ of type (ii) in $\sigma_{j,1}\cdots\sigma_{j,m_j}$, $j = 1, \ldots, k$ and $i = 1, \ldots, m$. Then $\tau_{j,l_{j,i}}$ is of the form $\alpha_{j,l_{j,i}} A^{\ell}_i \beta_{j,l_{j,i}}$, where $\alpha_{j,l_{j,i}}, \beta_{j,l_{j,i}} \in \kappa(\mathcal{B} \setminus \mathcal{B}')$ and

$$\begin{cases} \beta_{i_1}\tau_{1,1}\cdots\tau_{1,l_{1,1}-1}\alpha_{1,l_{1,1}} \leq 1 \\ \beta_{i_2}\tau_{2,1}\cdots\tau_{2,l_{2,1}-1}\alpha_{2,l_{2,1}} \leq 1 \\ \qquad\vdots \\ \beta_{i_k}\tau_{k,1}\cdots\tau_{k,l_{k,1}-1}\alpha_{k,l_{k,1}} \leq 1 \end{cases}, \tag{12}$$

i.e., the "before A_1" prefix in the left-hand side of each inequation reduces to 1;

$$\begin{cases} \beta_{1,l_{1,p}}\tau_{1,l_{1,p}+1}\cdots\tau_{1,l_{1,p+1}-1}\alpha_{1,l_{1,p+1}} \leq 1 \\ \beta_{2,l_{2,p}}\tau_{2,l_{2,p}+1}\cdots\tau_{2,l_{2,p+1}-1}\alpha_{2,l_{2,p+1}} \leq 1 \\ \qquad\vdots \\ \beta_{k,l_{k,p}}\tau_{k,l_{k,p}+1}\cdots\tau_{k,l_{k,p+1}-1}\alpha_{k,l_{k,p+1}} \leq 1 \end{cases}, \tag{13}$$

i.e., the subword between A_p and A_{p+1} in the left-hand side of each inequation reduces to 1, $p = 1, \ldots, m-1$; and

$$\begin{cases} \beta_{1,l_{1,m}}\tau_{1,l_{1,m}+1}\cdots\tau_{1,m_1}\alpha_{i_1} \leq 1 \\ \beta_{2,l_{2,m}}\tau_{2,l_{2,m}+1}\cdots\tau_{2,m_2}\alpha_{i_2} \leq 1 \\ \qquad\vdots \\ \beta_{k,l_{k,m}}\tau_{k,l_{k,m}+1}\cdots\tau_{k,m_k}\alpha_{i_k} \leq 1 \end{cases}, \tag{14}$$

i.e., the "after A_m" suffix in the left-hand side of each inequation reduces to 1.

To proceed from this point we shall need the following definition.

Definition 4. *Let* $\tau_i = \alpha_i A^{\ell}_i \beta_i$, $\alpha_i, \beta_i \in \kappa(\mathcal{B} \setminus \mathcal{B}')$ *and* $A_i \in \mathcal{B}'$, $i = 1, \ldots, k$. *We say that the set of categories* $\{\tau_j\}_{j=1,\ldots,k}$ *is* consistent*, if*

$$A_1 = \cdots = A_k.$$

Consider the directed graph $\mathcal{G}$ whose set of vertices consists of 0, $m+1$, and all $(k+1)$tuples of the form $(p, \tau_1, \ldots, \tau_k)$, $p = 1, \ldots, m$, where $\tau_j \in \overline{I}(\sigma_{j,l_{j,p}})$ and the set of categories $\{\tau_j\}_{j=1,\ldots,k}$ is consistent. Note that the number of vertices of $\mathcal{G}$ is linear in n.

The edges of $\mathcal{G}$ are as follows.

- There is no edge form vertex 0 to vertex $m+1$.
- There is an edge from vertex 0 to vertex

$$(p, \alpha_1 A_p \beta_1, \ldots, \alpha_k A_p \beta_k)$$

  if and only if $p = 1$ and there exist $\tau_{j,i} \in \overline{I}(\sigma_{j,i})$, $j = 1, \ldots, k$ and $i = 1, \ldots, l_{j,1} - 1$, such that

$$\begin{cases} \beta_{i_1} \tau_{1,1} \cdots \tau_{1,l_{1,1}-1} \alpha_1 & \leq 1 \\ \beta_{i_2} \tau_{2,1} \cdots \tau_{2,l_{2,1}-1} \alpha_2 & \leq 1 \\ & \vdots \\ \beta_{i_k} \tau_{k,1} \cdots \tau_{k,l_{k,1}-1} \alpha_k & \leq 1 \end{cases}, \tag{15}$$

  cf. (12).
- There is an edge from vertex

$$(p', \alpha'_1 A_{p'} \beta'_1, \ldots, \alpha'_k A_{p'} \beta'_k)$$

  to vertex

$$(p'', \alpha''_1 A_{p''} \beta''_1, \ldots, \alpha''_k A_{p''} \beta''_k)$$

  if and only if $p'' = p' + 1$ and there exist $\tau_{j,i} \in \overline{I}(\sigma_{j,i})$, $j = 1, \ldots, k$ and $i = l_{j,p'} + 1, \ldots, l_{j,p''} - 1$, such that

$$\begin{cases} \beta'_1 \tau_{1,l_{1,p'}+1} \cdots \tau_{1,l_{1,p''}-1} \alpha''_1 & \leq 1 \\ \beta'_2 \tau_{2,l_{2,p'}+1} \cdots \tau_{2,l_{2,p''}-1} \alpha''_2 & \leq 1 \\ & \vdots \\ \beta'_k \tau_{k,l_{k,p'}+1} \cdots \tau_{k,l_{k,p''}-1} \alpha''_k & \leq 1 \end{cases}, \tag{16}$$

  cf. (13).
- There is an edge from vertex

$$(p, \alpha_1 A_p \beta_1, \ldots, \alpha_k A_p \beta_k)$$

  to vertex $m+1$ if and only if $p = m$ and there exist $\tau_{j,i} \in \overline{I}(\sigma_{j,i})$, $j = 1, \ldots, k$ and $i = l_{j,m} + 1, \ldots, m_j$, such that

$$\begin{cases} \beta_1 \tau_{1,l_{1,m}+1} \cdots \tau_{1,m_1} \alpha_{i_1} & \leq 1 \\ \beta_2 \tau_{2,l_{2,m}+1} \cdots \tau_{2,m_2} \alpha_{i_2} & \leq 1 \\ & \vdots \\ \beta_k \tau_{k,l_{k,m}+1} \cdots \tau_{k,m_k} \alpha_{i_k} & \leq 1 \end{cases}, \tag{17}$$

  cf. (14).

Similarly to the case of (8), finding appropriate assignments, which satisfy the above (independent) inequations (15), (16), and (17) reduces to the membership problems in corresponding context-free languages. Therefore, $\mathcal{G}$ can be constructed in polynomial time.

Since (9) if and only if there is a path from 0 to $m+1$ in $\mathcal{G}$, the proof of Theorem 6 is complete.

7 Hierarchy of Restricted BAPGG Languages

In this section we show that the class of K-restricted BAPGG languages is a proper subclass of the $(K+1)$-restricted languages. We shall show first that each K-restricted BAPGG language is also a ($(K+1)$-restricted) BAPGG one.

Let $G = \langle \Sigma, \mathcal{B}, \leq, \mathcal{B}', I, \Delta \rangle$ be a BAPGG. Since, obviously, the class of (restricted) BAPGG languages is closed under union, we may assume that $\Delta = \{\delta\}$ consists of one category only. Consider the BAPGG $G' = \langle \Sigma, \mathcal{B} \cup \{Z\}, \leq', \mathcal{B}', I', \{\delta Z^k : k = 0, \ldots, K\} \rangle$, where $Z \notin \mathcal{B}$ and $\leq' = \leq \cup (Z, Z)$ and, for $\sigma \in \Sigma$, the lexical category assignment $I'(\sigma)$ is defined as follows.

- If σ is of type (i) or of type (ii), then
$$I'(\sigma) = \{(Z^r)^k \tau Z^k : \tau \in I(\sigma) \text{ and } k = 0, \ldots, K\}.$$
- If σ is of type (iii), then
$$I'(\sigma) = \{(Z^r)^k \tau Z^{k+1} : \tau \in I(\sigma) \text{ and } k = 0, \ldots, K-1\}.$$

We contend that $L(G') = L_K(G)$. We start with the proof of the inclusion $L(G') \subseteq L_K(G)$. Let $w \in L(G')$ and let $\tau \in I'(w)$, $\theta \in \mathcal{B}'^{+}$, and $k = 0, \ldots, K$ be such that
$$\tau[x := \theta] \leq \delta Z^k. \tag{18}$$
Then, τ has exactly k occurrences of x and, substituting 1 for Z in (18) we obtain
$$(\tau[Z := 1])[x := \theta] \leq \delta,$$
i.e., $w \in L_K(G)$.

Conversely, let $w = \sigma_1 \cdots \sigma_n \in L_K(G)$, $\tau_i \in I(\sigma_i)$, $i = 1, \ldots, n$, be such that $\tau_1 \cdots \tau_n$ has k occurrences of x, $k = 0, \ldots, K$, and
$$(\tau_1 \cdots \tau_n)[x := \theta] \leq \delta,$$
and let
$$0 = i_0 < i_1 < \cdots < i_j < \cdots < i_k < i_{k+1} = n+1$$
be such that that $\tau_{i_j} \in I(\sigma_{i_j})$, $j = 1, \ldots, k$, is of the form (iii). Then, for $\tau'_i \in I'(\sigma_i)$ defined by
$$\tau'_i = \begin{cases} (Z^r)^j \tau_i Z^{j+1}, & \text{if } i = i_j,\ j = 1, \ldots, k \\ (Z^r)^j \tau_i Z^j & \text{if } i_{j-1} < i < i_j,\ j = 1, \ldots, k+1 \end{cases},$$
we have
$$(\tau'_1 \cdots \tau'_n)[x := \theta] \leq \delta Z^k.$$
That is, $w \in L(G')$.

For the proof of the strict inclusion of the hierarchy levels consider the languages

$$L_{e,K} = \{(ab^n)^K : n = 1, 2, \ldots\},$$

where K is a positive integer. It can be readily seen that, for all positive integers K, $L_{e,K}$ is a K-restricted BAPGG language. For example,

$$L_{e,3} = L_3(G_{e,3}) = L(G_{e,3})$$

for the BAPGG $G_{e,3} = \langle \{a, b\}, \{B, S, T\}, =, \{B\}, I, \{T^3\}\rangle$, where I is defined by

- $I(a) = \{S\}$ and
- $I(b) = \{S^r B^\ell S, S^r xT\}$.

In particular, $abbbabbbabbb \in L_{e,3}$ can be derived as follows. The lexical category assignment is

$$\overbrace{S}^{a}\;\overbrace{S^r B^\ell S}^{b}\;\overbrace{S^r B^\ell S}^{b}\;\overbrace{S^r xT}^{b}\;\overbrace{S}^{a}\;\overbrace{S^r B^\ell S}^{b}\;\overbrace{S^r B^\ell S}^{b}\;\overbrace{S^r xT}^{b}\;\overbrace{S}^{a}\;\overbrace{S^r B^\ell S}^{b}\;\overbrace{S^r B^\ell S}^{b}\;\overbrace{S^r xT}^{b}$$

and, substituting $\theta = BB \, (\in \mathcal{B}'^{+})$ for x, we obtain (by **(con)**s)

$$SS^r B^\ell SS^r B^\ell SS^r BBTSS^r B^\ell SS^r B^\ell SS^r BBTSS^r B^\ell SS^r B^\ell SS^r BBT \leq TTT.$$

The definition of the BAPGG $G_{e,3}$ and the lexical category assignment above naturally extend to all positive integers K and all elements of $L_{e,K}$, implying $L_{e,K} \subseteq L(G_{e,K})$. The proof of the converse inclusion is equally easy and is omitted.

It easily follows from the pumping lemma for restricted BAPGG languages that $L_{e,K+1}$ is not a K-restricted BAPGG languages. Thus, the "K–hierarchy" of restricted BAPGG languages is strict.

8 An Extension of BAPGGs

It can be readily seen that all results of this paper also hold for "multi-buffer augmented" pregroup grammars defined below and their corresponding K-restricted languages.

Definition 5. (Cf. Definition 3.) *A q-buffer augmented pregroup grammar (q-BAPGG) is a tuple $G = \langle \Sigma, \mathcal{B}, \leq, \mathcal{B}', \mathcal{V}, I, \Delta\rangle$, where the components of G are as follows.*

- *Σ is a finite set of terminals (the* alphabet*).*
- *$\langle \mathcal{B}, \leq\rangle$ is a partially ordered finite set.*
- *$\mathcal{B}' \subseteq \mathcal{B}$ is the set of the* buffer *elements.*
- *$\mathcal{V} = \{x_1, \ldots, x_q\}$ is a set of* v*ariables (buffers) disjoint from $\kappa(\mathcal{B})$.*
- *I is a mapping that assigns to each element of Σ a finite set of categories from $\kappa(\mathcal{B} \cup \mathcal{V})$ such that for all $\sigma \in \Sigma$, each $\tau \in I(\sigma)$ is of one of the following forms:*

(i) $\tau \in \kappa(\mathcal{B} \setminus \mathcal{B}')$,
(ii) $\tau = \alpha A^{(\pm 1)}\beta$, *where* $A \in \mathcal{B}'$, $\alpha, \beta \in \kappa(\mathcal{B} \setminus \mathcal{B}')$, *or*
(iii) $\tau = \alpha x_i \beta$, *where* $\alpha, \beta \in \kappa(\mathcal{B} \setminus \mathcal{B}')$ *and* $i = 1, \ldots, q$.

In addition,
- *for each* $\tau = \alpha A^{(\pm 1)}\beta \in I(\sigma)$ *there is* $\tau' = \alpha A^{(\pm 1)}\beta' \in I(\sigma)$ *such that* $\beta'\alpha \leq 1$ *or there is* $\tau' = \alpha' A^{(\pm 1)}\beta \in I(\sigma)$ *such that* $\beta\alpha' \leq 1$, *and*
- *if* $I(\sigma)$ *contains a category of the form* (i), *then it contains no category of the form* (ii).

– $\Delta \subset \kappa(\mathcal{B} \setminus \mathcal{B}')$ *is a finite set of* distinguished *categories.*

The language generated by G is defined by

$$L(G) = \{w : \text{there exist } \tau \in I(w),\ \theta_i \in \mathcal{B}'^{+},\ i = 1, \ldots, q, \text{ and } \delta \in \Delta \\ \text{such that } \tau[x_i := \theta_i, i = 1, \ldots, m] \leq \delta\},$$

where $\tau[x_i := \theta_i, i = 1, \ldots, q]$ is the result of simultaneous substitution of θ_i for x_i, $i = 1, \ldots, q$, in τ.

We end this section with the question whether the class of q-BAPGG languages is a proper subclass of the $(q + 1)$-BAPGG ones.

9 Concluding Remarks

In our paper we argued that pregroup based grammars are a very convenient tool for describing mildly context-sensitive languages, introduced a new model of such grammars called (restricted) buffer augmented pregroup grammars, and established some of their basic properties. These grammars have a natural automaton counterpart, called (restricted) *buffer augmented pushdown automata* which are pushdown automata augmented with only once written additional memory – the buffer. The class of languages accepted by (restricted) *buffer augmented pushdown automata* coincide with the class of (mildly context-sensitive) languages generated by (restricted) buffer augmented pregroup grammars. This automaton model of computation will be published elsewhere.

References

1. Aizikowitz, T., Francez, N., Genkin, D., Kaminski, M.: Extending free pregroups with lower bounds. Studia Logica (to appear)
2. Buszkowski, W.: Lambek grammars based on pregroups. In: Groote, P.D., Morrill, G., Retorè, C. (eds.) LACL 2001. LNCS (LNAI), vol. 2099, pp. 95–109. Springer, Heidelberg (2001)
3. Buszkowski, W., Moroz, K.: Pregroup grammars and context-free grammars. In: Pre-proceedings of the international workshop on non-classical formal languages in linguistics, Tarragona, Spain, Research Group on Mathematical Linguistics, Universitat Rovira i Virgili (2007)
4. Francez, N., Kaminski, M.: Commutation-extended pregroup grammars and mildly context-sensitive languages. Studia Logica 87, 295–321 (2007)

5. Francez, N., Kaminski, M.: Commutation-augmented pregroup grammars and push-down automaton with cancellation. Information and Computation 206, 1018–1032 (2008)
6. Hopcroft, J., Ullman, J.: Introduction to automata theory, languages, and computation. Addison-Wesley, Reading (1979)
7. Lambek, J.: Type grammars revisited. In: Lecomte, A., Perrier, G., Lamarche, F. (eds.) LACL 1997. LNCS (LNAI), vol. 1582, pp. 1–27. Springer, Heidelberg (1999)
8. Michaelis, J., Kracht, M.: Semilinearity as a syntactic invariant. In: Retoré, C. (ed.) LACL 1996. LNCS (LNAI), vol. 1328, pp. 329–345. Springer, Heidelberg (1997)
9. Vijay-Shanker, K., Weir, D.: The equivalence of four extensions of context-free grammars. Mathematical Systems Theory 27(6), 511–546 (1994)