Radical Lexicalization of Mildly Context-Sensitive Languages

Daniel Genkin
Radical Lexicalization of Mildly Context-Sensitive Languages

Research Thesis

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Daniel Genkin

Submitted to the Senate of the Technion — Israel Institute of Technology
Tevet 5771 Haifa December 2010
The research thesis was done under the supervision of Prof. Michael Kamin-
ski in the Computer Science Department.

The generous financial support of the Technion is gratefully acknowledged.
Contents

Abstract 1

Abbreviations and Notations 2

1 Introduction 3
   1.1 Pregroup grammars and formal languages 3
   1.2 The results of the thesis 5

2 Pregroups and pregroup grammars 7

3 Buffer augmented pregroup grammars 12
   3.1 Definition of buffer augmented pregroup grammars 13
   3.2 Multiple agreement 15
   3.3 Crossed dependencies 16
   3.4 Reduplication 17

4 Pumping lemma for (restricted) buffer augmented pregroup grammar languages 19

5 Complexity of buffer augmented pregroup grammar languages 23
   5.1 Proof of lemma 22 27
   5.2 Proof of lemma 23 29

6 Complexity of restricted buffer augmented pregroup grammar 30
   6.1 Solving system (6.5) in polynomial time 32
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Hierarchy of restricted buffer augmented pregroup grammar languages</td>
<td>38</td>
</tr>
<tr>
<td>7.1</td>
<td>Each $K$-restricted BAPGG language is also a $(K+1)$-restricted BAPGG one</td>
<td>39</td>
</tr>
<tr>
<td>7.2</td>
<td>$K$-restricted BAPGG languages are strictly included into $(K+1)$-restricted BAPGG languages</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>Buffer Augmented Pushdown Automata</td>
<td>42</td>
</tr>
<tr>
<td>8.1</td>
<td>Definition of buffer augmented pushdown automata</td>
<td>43</td>
</tr>
<tr>
<td>8.2</td>
<td>From buffer augmented pushdown automata to buffer augmented pregroup grammar</td>
<td>49</td>
</tr>
<tr>
<td>8.3</td>
<td>From buffer augmented pregroup grammar to buffer augmented pushdown automata</td>
<td>51</td>
</tr>
<tr>
<td>9</td>
<td>Concluding remarks</td>
<td>53</td>
</tr>
<tr>
<td>Appendix A.</td>
<td>Proof of Theorem 33</td>
<td>55</td>
</tr>
<tr>
<td>Appendix B.</td>
<td>Proof of Theorem 34</td>
<td>59</td>
</tr>
<tr>
<td>Appendix C.</td>
<td>Proof of Theorem 38</td>
<td>65</td>
</tr>
<tr>
<td>Appendix D.</td>
<td>Proof of Theorem 39</td>
<td>69</td>
</tr>
<tr>
<td>Abstract in Hebrew</td>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>
Abstract

A family of languages is called mildly context-sensitive if

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

In this thesis 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.

We also present an automaton counterpart of buffer augmented pregroup grammars, called buffer augmented pushdown automata and prove that a language is generated by a buffer augmented pregroup grammar if and only if it is accepted by a buffer augmented pushdown automaton.
Abbreviations and Notations

$TLG$ — type logical grammar
$PGG$ — pregroup grammar
$BAPGG$ — buffer augmented pregroup grammar
$|\delta|$ — the length of $\delta$
$\tau[x := \theta]$ — simultaneous substitution of all appearances of $x$ with $\theta$ in $\tau$
$\epsilon$ — the empty word
$BAPDA$ — buffer augmented push down automaton
$[x]$ — the collection of all types derivable from $x$ by (con)s only.
Chapter 1

Introduction

1.1 Pregroup grammars and formal languages

Since their introduction in [12], 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 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 [9], 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 variables) 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.

The basic ideas in this vein were presented by Ajdukiewicz [10] and Bar-Hillel [7] and in [8] it was proved that Ajdukiewicz’s model is context free.

In [11] Lambek introduced a syntactic calculus that formalized the function type constructors along with various rules for the combination of functions. This calculus is a forerunner of linear logic in that it is a substructural logic. Montague grammar [14, 15, 17] uses an ad hoc syntactic system for English that is based on the principles of categorial grammar. Although Montague’s work is sometimes regarded as syntactically uninteresting, it helped to bolster interest in categorial grammar by associating it with a highly successful formal treatment of natural language semantics.
Later on, in [16] a new model called type-logical grammar (TLG), was presented. In this model categories (called also types) are formulas in a suitable logic, a syntactic calculus. The base logic is the (associative) Lambek calculus $L$, that has two directed implications, and that is naturally interpreted in string models. Starting with [12], there emerged an alternative approach, by which categories are elements of a suitable algebra, known as a pregroup (see below), that generalizes groups in having directed adjoints, which are the algebraic counterpart to the directed implication in $L$.

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^n b^n c^n d^n : m, n \geq 1\}), and reduplication (\{ww : w \in \Sigma^+\}), but on the other hand have semi-linearity\footnote{A language $L$ is called semi-linear if $\{|w| : w \in L\}$ is a finite union of sets of integers of the form $l + im : i = 0, 1, \cdots, l, m \geq 0.$} [13] 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 [18], namely to mildly context-sensitive languages.

\subsection{1.2 The results of the thesis}

In this thesis, 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. In addition, we present a hierarchy inside the class of languages generated by our extension and we present an automaton counterpart of this extension.\footnote{The results in Chapters 3, 4, 5, 6, 7 and 9 where first presented in [4]}

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

Pregroups and pregroup grammars

In this chapter we recall the definition of pregroup grammars.

Definition 1 A pregroup is a tuple \( P = \langle G, \leq, \circ, \ell, r, 1 \rangle \), such that \( \langle G, \leq, \circ, 1 \rangle \) is a partially-ordered monoid\(^1\), i.e., satisfying

\[(\text{mon}) \quad \text{if } A \leq B, \text{ then } CA \leq CB \text{ and } AC \leq BC\]

and \( \ell, r \) are unary operations (left/right inverses/adjoints) satisfying

\[(\text{pre}) \quad A^\ell A \leq 1 \leq AA^r \text{ and } AA^r \leq 1 \leq A^r A.\]

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

\[1^\ell = 1^r = 1, \quad A^r A^\ell = A, \quad (AB)^\ell = B^\ell A^\ell, \quad (AB)^r = B^r A^r.\]

Also, \((\text{mon})\) together with \((\text{pre})\) yield

\[A \leq B \text{ if and only if } B^\ell \leq A^\ell \text{ if and only if } B^r \leq A^r. \quad (2.1)\]

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

Types\(^2\) are finite strings (products) of terms. The set of all categories generated by \( B \) is denoted by \( \kappa(B) \).

\(^1\)\(\circ\) is usually omitted.
\(^2\)They are also called categories.
Remark 2  By definition, \( \kappa(B) = (\tau(B))^* \).

Extend ‘\( \leq \)’ to \( \kappa(B) \) by setting it to the smallest quasi-partial-order\(^3\) satisfying

(\text{con}) \quad \gamma A^{(n)} B^{(n+1)} \delta \leq \gamma \delta \quad \text{(contraction)}

(\text{exp}) \quad \gamma \delta \leq \gamma A^{(n+1)} B^{(n)} \delta \quad \text{(expansion)}

and

(\text{ind}) \quad \gamma A^{(n)} \delta \leq \gamma B^{(n)} \delta \quad \text{if}\quad \begin{cases} A \leq B \text{ and } n \text{ is even}, & \text{or} \\ B \leq A \text{ and } n \text{ is odd} \end{cases} \quad \text{(induced steps)}.

The following two inequalities can be easily derived from (\text{con}), (\text{exp}), and (\text{ind}).

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

and

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

Obviously, (\text{con}) and (\text{exp}) are particular cases of (\text{gcon}) and (\text{gexp}), respectively. Conversely, (\text{gcon}) can be obtained as (\text{ind}) followed by (\text{con}), and (\text{gexp}) can be obtained as (\text{exp}) followed by (\text{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 (\text{gcon}), (\text{gexp}), or (\text{ind}).

**Proposition 3** ([12, Proposition 2]) If \( \alpha' \leq \alpha'' \) has a derivation of length \( m \), then there exist categories \( \beta \) and \( \gamma \) such that

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

\(^3\)That is, \( \leq \) is not necessarily antisymmetrical.
Corollary 4 If $\alpha' \leq \alpha''$, then there exist types $\gamma$, $\gamma'$, and $\gamma''$ such that

- $\alpha' \leq \gamma$ by (ind) only;
- $\gamma \leq \gamma'$ by (con) only;
- $\gamma' \leq \gamma''$ by (exp) only; and
- $\gamma'' \leq \alpha''$ by (ind) only.

Proof As we have already observed above, (gcon) and (gexp) can be decomposed to (ind) followed by (con) and (exp) followed by (ind), respectively. In addition we observe that (con) followed by (ind) can be replaced by the same (ind) followed by the same (con), and (ind) followed by (exp) can be replaced by the same (exp) followed by the same (ind). Thus, the corollary follows.

Corollary 5 If $\alpha' \leq \alpha''$ where $\alpha''$ 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.

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

and

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

Next we strengthen Corollary 4. For this we need the following notation. Let $\langle \mathcal{B}, \leq \rangle$ be a partially ordered set.

- For a type $\alpha$ we denote by $t(\alpha)$ the set of all terms which occur in $\alpha$.
- For a term $t = A^{(n)}$ we denote its degree by $d(t)$, the absolute value of $n$:

$$d(\tau) = |n|.$$
We extend degrees to (non-empty) finite sets $C$ of types by

$$d(C) = \max \{ d(t) : t \in \bigcup_{\alpha \in C} t(\alpha) \}.$$ 

That is, $d(C)$ is the maximal degree of the terms which occur in the types in $C$.

**Corollary 6** ([6, Corollary 4]) If $\alpha' \leq \alpha''$, then there exist types $\gamma$, $\gamma'$, and $\gamma''$ such that

- $\alpha' \leq \gamma$ by (ind) only;
- $\gamma \leq \gamma'$ by (con) only;
- $\gamma' \leq \gamma''$ by (exp) only; and
- $\gamma'' \leq \alpha''$ by (ind) only; and
- for each term $t$ occurring in the above derivations, $d(t) \leq d(\{\alpha', \alpha''\})$.

Next we turn to type grammars based on quasi-pregroups.

**Definition 7** A pregroup grammar (PGG) is a tuple $G = \langle \Sigma, B, \leq, I, \Delta \rangle$, where

- $\Sigma$ is a finite set of terminals (the alphabet),
- $\langle B, \leq \rangle$ is a finite partially ordered set of atoms,
- $I$ is a finite-range mapping $I : \Sigma \to 2^{\tau(B)}$,\footnote{That is, $I(\sigma)$ is finite for all $\sigma \in \Sigma$.} and
- $\Delta \subset \tau(B)$ is a finite set of distinguished categories.\footnote{Cf. an equivalent definition in [2], where $\Delta$ consists of one term only.}

We extend $I$ to $\Sigma^+$ 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 8 ([6, Example 6]) Consider the PGG $G_a = \langle \Sigma, B, =, I, \Delta \rangle$, where

- $\Sigma = \{a, b\}$,
- $B = \{S, A\}$, 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

$$SA^\ell S^\ell SA^\ell S^\ell A A A$$

and cancellation is indicated by underline.

$$SA^\ell S^\ell SA^\ell S^\ell AAA \leq SA^\ell A^\ell A^\ell AAA \leq SA^\ell A^\ell AA \leq SA^\ell A \leq S.$$

Proposition 9 ([6, Proposition 7]) Let $G = \langle \Sigma, B, \leq, I, \Delta \rangle$ and $G' = \langle \Sigma, B, \leq, I', \Delta \rangle$ be pregroup grammars such that for each $\sigma \in \Sigma$ the following holds.

1. $I(\sigma) \subseteq I'(\sigma)$, and

2. for each type $\alpha' \in I'(\sigma)$ there is a type $\alpha \in I(\sigma)$ such that $\alpha \leq \alpha'$.

Then $L(G') = L(G)$.

Theorem 10 ([6, Theorem 2]) An $\epsilon$-free language $L$ is $L(G)$ for some pregroup grammar $G$ if and only if $L$ is context-free.
Chapter 3

Buffer augmented pregroup grammars

In this chapter we introduce buffer augmented pregroup grammars and present some of their basic properties. The idea behind this extension of pregroup grammars is the addition of a variable symbol - the buffer to the underlying free pregroup. The buffer is a lower bound on some set of elements of the underlying free pregroup. This buffer can store any element of the underlying (free) pregroup. Since the “data” stored in the buffer can be accessed in any position during a derivation of a word $w$, it can be used to compare and count different elements that appear in $w$. For example, in $L_{ma} = \{a^n b^n c^n : n \geq 0\}$ one can guess $n$ in unary and store inside the buffer. Afterwards, one can compare each subword $a^i$, $b^j$ and $c^k$ of an input word $w$ with the buffer. Since the data stored in the buffer is fixed throughout the derivation, it imposes $i = j = k = n$. Another use of the buffer can be seen in $L_{rd} = \{ww : w \in \{a,b\}^+\}$. In this case, given an input word $w$ it is possible to guess the first half of the word $w'$ and store it in the buffer. Afterwards, one can compare each half of $w$ with the buffer, which forces $w$ to be of the form $w = w'w' \in L_{rd}$. 

12
3.1 Definition of buffer augmented pregroup grammars

Definition 11 A buffer augmented pregroup grammar (BAPGG) is a tuple $G = \langle \Sigma, B, \leq, A, I, \Delta \rangle$, where the components of $G$ are as follows.

- $\Sigma$ is a finite set of terminals (the alphabet).
- $\langle B, \leq \rangle$ is a partially ordered finite set such that if $A \leq B$ or $B \leq A$ for some $A \in A$ and $B \in B$ then $A = B$.
- $A \subseteq B$ is the set of the buffer elements.
- $I$ is a mapping that assigns to each element of $\Sigma$ a finite set of categories from $\kappa(B \cup \{x\})$, 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) $\tau \in \kappa(B \setminus A)$,
  
  (ii) $\tau = \alpha A^{(\pm 1)} \beta$, where $A \in A$, $\alpha, \beta \in \kappa(B \setminus A)$, or
  
  (iii) $\tau = \alpha x \beta$, where $\alpha, \beta \in \kappa(B \setminus A)$.

  In addition,
  
  - for each $\tau = \alpha A^r \beta \in I(\sigma)$ there is $\tau' = \alpha A^r \beta' \in I(\sigma)$ such that $\beta' \alpha \leq 1$,\(^1\) and
  
  - for each $\tau = \alpha A^l \beta \in I(\sigma)$ there is $\tau' = \alpha' A^l \beta \in I(\sigma)$ 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),\(^2\) and we shall say that $\sigma$ is of type (i) or type (ii), respectively.

- $\Delta \subseteq \kappa(B \setminus A)$ is a finite set of distinguished categories.

The language generated by $G$ is defined by

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

\(^1\)This condition is needed for the proof of the pumping lemma for BAPGG languages, but is not needed for polynomial parsing of restricted BAPGG languages defined below.

\(^2\)This constraint is needed for polynomial parsing of restricted BAPGG languages, but is not needed for the proof of the pumping lemma for BAPGG languages.
where \( \tau[x := \theta] \) is the result of simultaneous substitution of \( \theta \) for \( x \) in \( \tau \).

We shall see in Chapter 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.

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

That is, \( K \) is the number of times the BAPGG “may consult” its buffer. It is shown in Chapter 6 that the membership problem for restricted BAPGG languages can be solved in polynomial time.\(^3\)

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

**Theorem 12** Buffer augmented pregroup grammars are at least as powerful as pregroup grammars.

**Proof** Let \( G = (\Sigma, B, \leq, I, \Delta) \) be a PGG. Then \( L(G) = L(G') \), where the BAPGG \( G' \) is defined by \( G' = (\Sigma, B, \leq, \emptyset, I, \Delta) \). 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 11.

The same construction, obviously, works for the restricted languages. \( \blacksquare \)

**Remark 13** 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 [6]) with the additional feature that, before reading \( x \), the automaton can pop a number of symbols from \( \{A^\ell : A \in A\} \) from the pushdown stack (using \( \epsilon \)-moves) and then push there a number of symbols from \( A \) (again using \( \epsilon \)-moves).

We show next that characteristic mildly context-sensitive languages are

\(^3\)In other words, the membership problem for BAPGG languages is fixed-parameter tractable.
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 [5].

3.2 Multiple agreement

Let \( L_{ma} = \{a^n b^n c^n : n \geq 1\} \) and let \( G_{ma} = (\Sigma, B, =, A, I, \Delta) \), where

- \( \Sigma = \{a, b, c\} \),
- \( B = \{A, P, T, U\} \),
- \( A = \{A\} \),
- \( I \) is defined by
  - \( I(a) = \{A^\ell, xT\} \),
  - \( I(b) = \{T^\nu A^\ell T, T^\nu xU\} \), and
  - \( I(c) = \{U^\nu A^\ell U, U^\nu xP\} \),

and

- \( \Delta = \{P\} \).

Then

\[ L(G_{ma}) = L_3(G_{ma}) = L_{ma}. \]

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

\[
A^\ell \quad A^\ell \quad \underbrace{a \quad a \quad b \quad b}_x \quad \underbrace{b \quad b \quad c \quad c \quad c}_x \quad \text{\ldots} \quad U^\nu A^\ell U \quad U^\nu A^\ell U \quad U^\nu xP \quad (3.1)
\]

\[ \text{Example 15 in the next section shows that } L_{ma} \text{ is not a 2-restricted BAPGG language.} \]

15
and, substituting $\theta = AA (\in A^+)$ for $x$, we obtain

$$A^\ell A^\ell \theta TT^T A^\ell TT^T A^\ell TT^T \theta UU^T A^\ell UU^T A^\ell UU^T \theta P \leq A^\ell A^\ell \theta A^\ell \theta A^\ell \theta P = A^\ell A^\ell AAA^\ell A^\ell AAA^\ell A^\ell AAP \leq A^\ell A^\ell AAA^\ell A^\ell AAA^\ell A^\ell AAP \leq P.$$

The lexical category assignment (3.1) naturally extends on all elements of $L_{ma}$, implying $L_{ma} \subseteq L(G_{ma})$.

For the proof of the converse inclusion $L(G_{ma}) \subseteq L_{ma}$, consider $w \in \Sigma^+$ such that for some $\tau \in I(w)$ there exists a substitution $\theta \in A^+$ for which $\tau[x := \theta] \leq P$. It follows from Corollary 5 and the definition of $I$ that after replacing all $A$’s in $\tau[x := \theta]$ with 1 we obtain

$$\frac{1}{a} \cdots \frac{1}{a} xT T^T \cdots T^T T^T U U^T U \cdots U^T U U^T \theta P \leq P$$

Therefore, $w = a^i b^j c^k$ and $\tau$ is of the form $\alpha xTT^\tau \beta xUU^\tau \gamma xP$, where $\alpha = (A^\ell)^{-1}$, $\beta = (T^\tau A^\ell T)^{-1}$, and $\gamma = (U^\tau A^\ell U)^{-1}$. Thus $\theta = A^{i-1} (= A^{j-1} = A^{k-1})$ and the desired inclusion follows.

### 3.3 Crossed dependencies

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

- $\Sigma = \{a, b, c, d\}$,
- $B = \{A, B, P, T, U, V\}$,
- $A = \{B\}$,
- $I$ is defined by
  - $I(a) = \{A^\ell, A^\ell T\}$,
  - $I(b) = \{T^\tau B^\ell T, T^\tau xU\}$,
  - $I(c) = \{U^\tau AU, U^\tau AV\}$, and
  - $I(d) = \{V^\tau B^\ell V, V^\tau xP\}$,
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

\[ a \xrightarrow{A} T \xrightarrow{T} T \xrightarrow{b} T \xrightarrow{T} T \xrightarrow{x} U \xrightarrow{T} A \xrightarrow{U} U \xrightarrow{AV} \ell \xrightarrow{V} b \xrightarrow{V} V \xrightarrow{B} \ell \xrightarrow{V} V \xrightarrow{r} P. \]

and, substituting \( \theta = BB (\in A^+) \) for \( x \), we obtain

\[ A^\ell A^T T^r B^\ell T^r B^\ell T^r \theta U^r U^r A \xrightarrow{V} V^r B^\ell V^r B^\ell V^r B^\ell \theta P \]
\[ \leq A^\ell A^\ell B^\ell B^\ell \theta A A B^\ell B^\ell \theta P \]
\[ = A^\ell A^\ell A^\ell B^\ell B B A A B^\ell B^\ell B B P \]
\[ \leq A^\ell A^\ell A A P \]
\[ \leq P. \]

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.

### 3.4 Reduplication

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

- \( \Sigma = \{a, b\} \),
- \( B = \{A, B, C, D, P, T\} \),
- \( A = \{A, B\} \),
- \( I \) is defined by
  - \( I(a) = \{A^\ell, xC^\ell T, T^r A^\ell T, T^r A^\ell xCP\} \) and
  - \( I(b) = \{B^\ell, xD^\ell T, T^r B^\ell T, T^r B^\ell xDP\} \),
\[ \Delta = \{ P \}. \]

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

\[
\begin{array}{cccccccc}
  & & a & b & b & a & b & b \\
A & B & x & D & T & T & T & T & x & D & P
\end{array}
\]

and, substituting \( \theta = BA (\in A^+) \) for \( x \), we obtain

\[
A^\ell B^\ell \theta D^\ell T T T \ A^\ell T T T B^\ell T T T \theta D P \leq A^\ell B^\ell \theta D^\ell A^\ell B^\ell \theta D P
\]

\[
= A^\ell B^\ell B A D^\ell A^\ell B^\ell B A D P
\]

\[
\leq D^\ell D P
\]

\[
\leq P.
\]

We omit the proof of the equality \( L(G_{rd}) = L_{rd} \).
Chapter 4

Pumping lemma for (restricted) buffer augmented pregroup grammar languages

In this chapter we present a version of a pumping lemma for (restricted) BAPGG languages. The idea laying behind this version of the pumping lemma is that using the first two additional properties of the lexical category assignment we can replicate some symbols of the buffer and thus “pumping” more symbols inside the input word.

Theorem 14 For each BAPGG language \( L \) there exist a positive integer \( N \) such that every \( w \in L \), \( |w| \geq N \), can be partitioned as

\[
w = u_1v_1u_2v_2 \cdots u_mv_mv_{m+1}
\]

where

- \( m \geq 1 \),
- \( |v_1|, |v_2| \geq 1 \), if \( m \leq 2 \), and \( |v_1| = \cdots = |v_m| = 1 \), if \( m \geq 3 \), and
- for all \( i \geq 1 \)
  \[
u_1v_1^iu_2v_2^i \cdots u_mv_m^iu_{m+1} \in L. \]

\(^1\)Note the difference with the ordinary pumping lemma, where \( i \geq 0 \).
Proof Let $L = L(G)$ for a BAPGG $G = (\Sigma, B, \leq, A, I, \Delta)$ and let $G' = (\Sigma, B, \leq, \emptyset, I, \Delta)$. 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 $|w| \geq N$. If $w \in L(G')$, then the theorem follows from the ordinary pumping lemma for context-free languages.

Otherwise, i.e., $w \not\in G'$, every $\tau \in I(w)$ such that $\tau[x := \theta] \leq \delta$, for some $\theta \in A^+$ and some $\delta \in \Delta$, must contain an occurrence of $x$.

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

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 $u_1 v_1 u_2 v_2 \cdots u_m v_m u_{m+1}$ and the substitution $\theta$ for $x$ are as follows.

- The lexical category assignment to the elements of $\Sigma$ occurring in the $u_j$s is the original one.
- The $i$ copies of $v_j = \sigma_{k_j}$, $j = 1, \ldots, m$, are assigned

$$\underbrace{\sigma_{k_j} \cdots \sigma_{k_j}}_{\text{i-1 times}} \alpha_j t_j \beta_j \cdots \alpha_j t_j \beta_j$$

if $t_j = A^r$ and thus there is $\alpha_j t_j \beta_j' \in I(\sigma_{k_j})$ such that $\beta_j' \alpha \leq 1$, or are assigned

$$\underbrace{\sigma_{k_j} \cdots \sigma_{k_j}}_{\text{i-1 times}} \alpha_j t_j \beta_j \cdots \alpha_j t_j \beta_j$$

if $t_j = A^\ell$ and thus there is $\alpha_j' t_j \beta_j \in I(\sigma_{k_j})$ such that $\beta_j \alpha' \leq 1$. 

20
The substitution for $x$ is $A^i\theta'$.

Then, in the former case,

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

and, in the latter case,

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

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

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

Example 15 It immediately follows from Theorem 14 that the multiple agreement language $L_{ma}$ is not a 2-restricted BAPGG language.

Remark 16 Usually, a pumping lemma is used to show that a specific language does not belong to the languages under consideration. We refer the reader to Section 7.2 that demonstrates such use.

In particular, Theorem 14 implies that if a language $L$ is generated by a BAPGG, every $w$ in $L$ is built from some small set of basic building blocks concatenated together using Kleene star, much like sentences in natural languages that are constructed from nouns connected together by grammatical conjunctions and prepositions.

We shall prove next that restricted BAPGG languages are semi-linear the proof is based on Lemma 17 below.

Lemma 17 Given a BAPGG $G = \langle \Sigma, \mathcal{B}, \leq, \mathcal{A}, I, \Delta \rangle$ and an integer $K \geq 0$ there exists an integer $C$ and two finite sets $S_G, T_G \subseteq \mathbb{N}$ such that

- For every $w \in L_K(G)$, if $|w| \geq C$, then there exists $s \in S_G$ and $t \in T_G$ such that

$$|w| = sn + t$$
for some \( n \geq 0 \).

- For every \( s \in S_G \), \( t \in T_G \) and an integer \( n \geq 0 \) such that \( sn + t \geq C \) there exists \( w \in L_K(G) \) such that

\[
|w| = sn + t
\]

**Proof** Let \( N \) be the constant from Theorem 14 and let \( w \in L_K(G) \), \( |w| \geq N \). There exists \( \tau \in I(w) \) with at most \( k \leq K \) appearances of \( x \) and there exists \( \theta \in A^+ \) such that

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

for some \( \delta \in \Delta \). Thus, there exists \( i \geq 0 \) and \( r \leq k - 1 \) such that

\[
|w| = ki + r
\]

Let \( S_G \) be the set of possible values for \( k \) and let \( T_G \) be all possible values of \( r \). Since \( 1 \leq k \leq K \) and \( 0 \leq r \leq K - 1 \), both \( S_G \) and \( T_G \) are finite. Thus, the first part of the theorem easily follows.

Let

\[
d = \max_{s \in S_G, t \in T_G} \min_{n \in \mathbb{N}} \{ sn + t : \text{there exists } w \in L_K(G) \text{ such that } |w| = sn + t \}
\]

we let \( C = \max\{d, N\} \).

For the proof of the second part of the theorem let \( s \in S_G \) and \( t \in T_G \). By the definition of \( S_G \) and \( T_G \) there exists \( w \in L_K(G) \), \( |w| \geq C \) such that

\[
|w| = sn + t \quad (4.1)
\]

for some \( n \). Assume \( w \) is of minimal length satisfying (4.1). Let, \( n' \geq 0 \) be any integer such that \( sn' + t \geq C \). Then by the choice of \( C \), \( sn' + t \geq sn + t \) and by Theorem 14 there exists \( w' \), \( |w'| = sn' + t \) such that \( w' \in L_K(G) \). That is, the proof of the second part of the lemma is complete.

**Corollary 18** Restricted BAPGG languages are semi-linear.
Chapter 5

Complexity of buffer augmented pregroup grammar languages

In this chapter we show that the membership problem for BAPGG languages is NP-complete. The main idea behind the proof is a reduction from the 3−SAT problem. Namely we define BAPGG $G_{3SAT}$ and a translation $[\phi]$ of a 3-CNF formula $\phi$ such that $\phi$ is satisfiable if and only if $[\phi]$ belongs to $L(G_{3SAT})$.

Theorem 19 The membership problem for BAPGG languages is in NP.

Proof Let $G = \langle \Sigma, B, \leq, A, I, \Delta \rangle$ be a BAPGG and let

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

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

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

Since $\theta$ is in $A^{+}$, no term occurring in a copy of it can be canceled by a term occurring in another copy. Therefore,

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

23
That is, an appropriate lexical category assignment $\tau \in I(w)$, the substitution $\theta$, and an appropriate $\delta \in \Delta$ can be “guessed” in an $O(|w|)$ time.

\textbf{Theorem 20} The membership problem for BAPGG languages is NP-hard.

The proof of Theorem 20 is by a polynomial reduction from the 3-SAT problem. Namely, we shall construct a BAPGG $G_{3\text{-SAT}}$ 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 $\varphi$ is in $L(G_{3\text{-SAT}})$ if and only if $\varphi$ is satisfiable.

The language of $G_{3\text{-SAT}}$ 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 $c = L_i \lor L_j \lor L_k$ we associate its type $t(c) = t_m t'_m t''_m$, $m = 0, \ldots, 7$, that is defined as follows.

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

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

$$[c_q] = \#^{l_i-1} b r^{n-i-1} b r^{n-j-1} b r^{n-k-1} b s t_m t'_m t''_m t_m @,$$

where $t(c_q) = t_m t'_m t''_m$.  

24
Remark 21 In the above encoding, $b$ is the “buffer symbol” to be substituted with the content of the buffer; the pairs of words $(l_{i-1}, r_{n-i})$, $(l_{n-j}, r_{n-j})$, and $(l_{k-1}, r_{n-k})$ indicate the literal variable (whose truth assignment will be cut from the “truth assignment word” $v_1 \cdots v_n \in \{\bot, \top\}$ provided by the buffer); and the type $t(c_q) = t_m t_m' t_m''$ determines the type of the literals in the clause $c_q$. The delimiters $\#$, $\$, @, and $\%$ are needed for a technical (cancellation) purpose that will become clear in the sequel.

Now, the encoding $[\phi]$ of $\phi$ over $\Sigma$ is $[\phi] = [c_1] \cdots [c_p]$.

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

$$B = \{A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, S, \bot, \top\}.$$ Intuitively, $A_m$, $m = 0, \ldots, 7$, correspond to truth assignments as follows.

- $A_0 \leftrightarrow (\bot, \bot, \bot)$.
- $A_1 \leftrightarrow (\bot, \bot, \top)$.
- $A_2 \leftrightarrow (\bot, \top, \bot)$.
- $A_3 \leftrightarrow (\bot, \top, \top)$.
- $A_4 \leftrightarrow (\top, \bot, \bot)$.
- $A_5 \leftrightarrow (\top, \bot, \top)$.
- $A_6 \leftrightarrow (\top, \top, \bot)$.
- $A_7 \leftrightarrow (\top, \top, \top)$.

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, $A = \{\bot, \top\}$ and $I$ is defined as follows.

- $I(b) = \{x\}$.
- $I(l) = \{\bot^\ell, \top^\ell\}$.
- $I(r) = \{\bot^r, \top^r\}$.
- $I(#) = \{S\}$.
\[ I(\$) = \{A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7\}. \]

\[ I(@) = \{A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7\}. \] That is, the lexical category assignment to \$ is supposed to be canceled by the lexical category assignment to @, see Remark 21.

\[ I(\#) = \{S^r\}. \] That is, the lexical category assignment to \# is supposed to be canceled by the lexical category assignment to \%, see Remark 21.

\[ I(t_0) = \{A_1 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t'_0) = \{A_1 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t''_0) = \{A_1 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t_1) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t'_1) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t''_1) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t_2) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t'_2) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t''_2) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t_3) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t'_3) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t''_3) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t_4) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t'_4) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t''_4) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t_5) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t'_5) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]

\[ I(t''_5) = \{A_0 \perp A_1^r, A_2 \perp A_2^r, A_3 \perp A_3^r, A_4 \perp A_4^r, A_5 \perp A_5^r, A_6 \perp A_6^r, A_7 \perp A_7^r\}. \]
• \( I(t_6) = \{ A_0 \perp^* A_0^\ell, A_1 \perp^* A_1^\ell, A_2 \perp^* A_2^\ell, A_3 \triangledown^* A_3^\ell, A_4 \perp^* A_4^\ell, A_5 \triangledown^* A_5^\ell, A_7 \triangledown^* A_7^\ell \} \).

• \( I(t_6') = \{ A_0 \perp^* A_0^\ell, A_1 \perp^* A_1^\ell, A_2 \perp^* A_2^\ell, A_3 \triangledown^* A_3^\ell, A_4 \perp^* A_4^\ell, A_5 \perp^* A_5^\ell, A_7 \triangledown^* A_7^\ell \} \).

• \( I(t_6'') = \{ A_0 \perp^* A_0^\ell, A_1 \perp^* A_1^\ell, A_2 \perp^* A_2^\ell, A_3 \perp^* A_3^\ell, A_4 \perp^* A_4^\ell, A_5 \perp^* A_5^\ell, A_6 \perp^* A_6^\ell \} \).

• \( I(t_7) = \{ A_0 \perp^* A_0^\ell, A_1 \perp^* A_1^\ell, A_2 \perp^* A_2^\ell, A_3 \perp^* A_3^\ell, A_4 \perp^* A_4^\ell, A_5 \perp^* A_5^\ell, A_6 \perp^* A_6^\ell \} \).

• \( I(t_7') = \{ A_0 \perp^* A_0^\ell, A_1 \perp^* A_1^\ell, A_2 \perp^* A_2^\ell, A_3 \perp^* A_3^\ell, A_4 \perp^* A_4^\ell, A_5 \perp^* A_5^\ell, A_6 \perp^* A_6^\ell \} \).

• \( I(t_7'') = \{ A_0 \perp^* A_0^\ell, A_1 \perp^* A_1^\ell, A_2 \perp^* A_2^\ell, A_3 \perp^* A_3^\ell, A_4 \perp^* A_4^\ell, A_5 \perp^* A_5^\ell, A_6 \perp^* A_6^\ell \} \).

To complete the proof of Theorem 19 we need to prove the following to lemmas

**Lemma 22** \( L_{3-SAT} \subseteq L(G_{3-SAT}) \)

**Lemma 23** \( L(G_{3-SAT}) \cap \{ \varphi : \varphi \in 3-CNF \} \subseteq L_{3-SAT} \)

### 5.1 Proof of lemma 22

The proof of the inclusion \( L_{3-SAT} \subseteq L(G_{3-SAT}) \) is based on Lemmas 24 and 25 below.

**Lemma 24** Let \( c = L_i \vee L_j \vee L_k \) be a 3-CNF clause of type \( m, m = 0, \ldots, 7 \), and let \( v_i, v_j, v_k = \perp, \top \) be such that \((v_i, v_j, v_k)\) satisfies \( c \). 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 \( A v_i^r A^\ell \in I(t_m), A v_j^r A^\ell \in I(t'_m), \) 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 \((v_i, v_j, v_k) = (\top, \top, \perp)\), then \( A = A_6 \).

**Lemma 25** Let \( c = L_i \vee L_j \vee L_k \) be a 3-CNF clause and let \( V = (v_1, \ldots, v_n) \in \{ \perp, \top \}^n \) be an “assignment vector” such that \( c |_V = \top \). Then there exists \( \tau \in I(\{ c \}) \) such that

\[
\tau[x := v_1 \cdots v_n] \leq 1.
\]
\textbf{Proof} It follows from $c|_V = \top$ that
\begin{equation}
    c(v_i, v_j, v_k) = \top. \tag{5.1}
\end{equation}

Let $c$ be of type $m$. Then,
\begin{align*}
[c] = \# l^{-1} b r^{-i} p^{-1} b r^{-j} t_{m'} t_{m^*} @ \% = \\
\# l \cdots b r \cdots r l \cdots b r \cdots r l \cdots b r \cdots r \quad \$t_{m'} t_{m^*} @ \%,
\end{align*}
and desired lexical category assignment $\tau \in I([c])$ is defined by
\begin{align*}
\tau = S & \quad l \quad l \quad l \quad l \quad b \quad b \quad r \quad r \\
& \quad v_{i-1}^l \quad \ldots \quad v_1^l \quad x \quad v_n^r \quad \ldots \quad v_{n-i}^r \\
& \quad v_{j-1}^l \quad \ldots \quad v_1^l \quad b \quad b \quad r \quad r \\
& \quad v_{n-j}^r \quad \ldots \quad x \quad v_n^r \quad \ldots \quad v_{n-j}^r \\
& \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\
& \quad l \quad l \quad l \quad l \quad b \quad b \quad r \quad r \\
& \quad v_{k-1}^l \quad \ldots \quad v_1^l \quad x \quad v_n^r \quad \ldots \quad v_{n-k}^r \\
& \quad S \quad v_m'' \quad v_m' \quad t_{m} \quad t_{m^*} \quad @ \% \quad A \quad A' \quad S^r,
\end{align*}
where $A$ is provided by (5.1) and Lemma 24. Therefore,
\begin{align*}
\tau[x := v_1 \cdots v_n] &= S \quad v_{i-1}^l \cdots v_1^l v_1 \cdots v_n v_n^r \cdots v_{n-i}^r \\
& \quad v_{j-1}^l \cdots v_1^l v_1 \cdots v_n v_n^r \cdots v_{n-j}^r \\
& \quad v_{k-1}^l \cdots v_1^l v_1 \cdots v_n v_n^r \cdots v_{n-k}^r A A k A' A' A A' \ S^r \\
& \leq S \quad v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 S^r \\
& \leq 1. \tag{5.2}
\end{align*}

Now, let $\varphi = \bigwedge_{q=1}^p c_q$ and let $V = (v_1, \ldots, v_n) \in \{\bot, \top\}^n$ be an “assignment vector” such that $c|_V = \top$. By Lemma 25, for every $q = 1, \ldots, p$,
\begin{equation}
    [c_q][x := v_1 \cdots v_n] \leq 1.
\end{equation}

Therefore,
\begin{align*}
[\varphi][x := v_1 \cdots v_n] &= ([c_1] \cdots [c_p])[x := v_1 \cdots v_n] \\
&= ([c_1][x := v_1 \cdots v_n]) \cdots ([c_p][x := v_1 \cdots v_n]) \\
&\leq 1
\end{align*}
and the desired inclusion follows.
5.2 Proof of lemma 23

For the proof of the inclusion

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

we shall need the following lemma.

**Lemma 26** Let \( c \) be a 3-CNF clause, \( \tau \in I([c]) \), and let \( V = (v_1, \ldots, v_n) \in \{ \bot, \top \}^n \) be such that

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

then \( c[V] = \top \).

**Proof** The lemma follows immediately from (5.2) and the definition of lexical category assignment \( I \) to \( t_m, t'_m \), and \( t''_m \), \( m = 0, \ldots, 7 \).

Now let \( \varphi = \bigwedge_{q=1}^p c_q \) be a 3-CNF formula and let

\[ [\phi] = \underbrace{\#w_1\% \cdots \#w_p\%}_{c_1 \cdots c_p} \]

where \( [c_q] = \#w_q\% \), \( q = 1, \ldots, p \).

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

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

For the proof of the inclusion (5.3) we have to show that \( V \) satisfies \( \phi \). 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 \( \tau'_q \)s in \( I(w_q) \), \( q = 1, \ldots, p \). Then the \( q \)th \( S^r \) (from left to right) must be canceled from the left by the \( q \)th \( S \), \( q = 1, \ldots, p \). 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 \( \tau_q \in I([c_q]) \), by Lemma 26, \( V \) satisfies all clauses of \( \varphi \) and the proof is complete.
Chapter 6

Complexity of restricted buffer augmented pregroup grammar

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

The main idea behind our solution is to reduce the following problem: given a restricted BAPGG $G = \langle \Sigma, \mathcal{B}, \preceq, \mathcal{A}, I, \Delta \rangle$, a word $w \in \Sigma^+$ and an integer $K \geq 0$, find $\tau \in I(w)$ with at most $K$ appearances of $x$ and $\theta \in \mathcal{A}^+$ such that $\tau[x := \theta] \leq \delta$ for some $\delta \in \Delta$, to the problem of solving a system of inequations in one variable over pregroups. In the description below we show how this system can be built in deterministic polynomial time (the polynomial degree is a function of $K$) and in section (6.1) we show that such a system can also be solved in polynomial time. Thus, we obtain a deterministic polynomial parsing algorithm for restricted buffer augmented pregroup grammars (the polynomial degree is a function of $K$).

**Theorem 27** The membership problem for restricted BAPGG languages is in $P$.

The proof of Theorem 27 is based on a sequence of reductions described below.

Let $G = \langle \Sigma, \mathcal{B}, \preceq, \mathcal{A}, I, \Delta \rangle$ 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 $\theta \in A^+$ and $\delta \in \Delta$ such that

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

Therefore, there exist positive integers $i', i, i''$ such that exactly one of the following cases holds:

1. $\tau_{i'} = \alpha_{i'} A_{i'}^x \beta_{i'}$ and $\tau_{i''} = \alpha_{i''} A_{i''}^\nu \beta_{i''}$ are categories of the form (ii), $\tau_i = \alpha_i x \beta_i$ is a category of the form (iii), and

$$(A_{i'}^x \beta_{i'} \tau_{i'+1} \cdots \tau_{i-1} \alpha_i x \beta_i \tau_{i+1} \cdots \tau_{i-1} \alpha_{i''} A_{i''}^\nu)[x := \theta] \leq 1; \tag{6.1}$$

2. $\tau_{i'} = \alpha_{i'} A_{i'}^x \beta_{i'}$ is a category of the form (ii), $i'' = i$, $\tau_i = \alpha_i x \beta_i$ is a category of the form (iii), and

$$(A_{i'}^x \beta_{i'} \tau_{i'+1} \cdots \tau_{i-1} \alpha_i x)[x := \theta] \leq 1; \tag{6.2}$$

3. $\tau_{i''} = \alpha_{i''} A_{i''}^\nu \beta_{i''}$ is a category of the form (ii), $i' = i$, $\tau_i = \alpha_i x \beta_i$ is a category of the form (iii), and

$$(x \beta_i \tau_{i''+1} \cdots \tau_{i''-1} \alpha_i A_{i''}^\nu)[x := \theta] \leq 1; \tag{6.3}$$

In addition, $\tau_1 \cdots \tau_{i'-1} \tau_{i'+1} \cdots \tau_n$ has at most $k - 1$ occurrences of $x$ and

$$\tau_1 \cdots \tau_{i'-1} \alpha_{i'} \beta_{i'} \tau_{i'+1} \cdots \tau_n \leq \delta. \tag{6.4}$$

Let $\sigma_{i'}$ and $\sigma_{i''}$ be disjoint copies of $\sigma_{i'}$ and $\sigma_{i''}$ receptively. Consider the PGG $\hat{G} = \langle \Sigma \cup \{\sigma_{i'}, \sigma_{i''}\}, B, \leq, \hat{I}, \delta \rangle$, where $\hat{I}$ is defined as follows.

$$\hat{I}(\sigma) = \begin{cases} I(\sigma), & \text{if } \sigma \in \Sigma \\ \alpha_{i'}, & \text{if } \sigma = \sigma_{i'}' \\ \beta_{i''}, & \text{if } \sigma = \sigma_{i''}'. \end{cases}$$

Then there is a assignment for $\sigma_i$,

$$i \in \{1, \ldots, i' - 1\} \cup \{i'' + 1, \ldots, n\}$$

---

That is, $A_{i'}^j$ and $A_{i''}^\nu j = 1, \ldots, k$, cancel the rightmost and the leftmost symbols of $\theta$, respectively.
containing at most \( k - 1 \) occurrences of \( x \) and satisfying (6.4) if and only if (6.1) and

\[
\hat{w} = \sigma_1 \cdots \sigma_{i-1} \sigma_i' \sigma_{i+1} \cdots \sigma_n \in L_{k-1}(\hat{G}),
\]

using \( \theta \) as the assignment for \( x \). After applying the reduction described above \( k \leq K \) times each time using the word \( \hat{w} \) and the grammar \( \hat{G} \) obtained in a previous step as the input for the next one we obtain a word \( \tilde{w} \), a grammar \( \bar{G} \) and a system

\[
\begin{align*}
(A^1_{i_1} & \beta_1 \cdot \tau_1 \tau_2 \cdots \tau_{i_1-1} \alpha_i x \beta_1 \cdot \tau_{i_2+1} \cdots \tau_i \cdots \tau_{i_{i_1-1}} A^1_{i_1})[x := \theta] \leq 1 \\
(A^2_{i_2} & \beta_2 \cdot \tau_2 \tau_3 \cdots \tau_{i_2-1} \alpha_i x \beta_2 \cdot \tau_{i_3+1} \cdots \tau_i \cdots \tau_{i_{i_2-1}} A^2_{i_2})[x := \theta] \leq 1 \\
& \cdots \\
(A^k_{i_k} & \beta_k \cdot \tau_{i_k+1} \cdots \tau_i \cdots \tau_{i_{i_k-1}} \alpha_i A^k_{i_k})[x := \theta] \leq 1
\end{align*}
\]

(6.5)

Such that \( w \in L_K(\hat{G}) \) if and only if there exists \( x \in A^+ \) satisfying (6.5) and

\[
\tilde{w} \in L_0(\bar{G})
\]

Obviously, the number of possible systems using (6.1), (6.2) and (6.3) is bounded by a polynomial in \( n \) and thus the process of computing \( \tilde{w} \) and \( \bar{G} \) is also a polynomial in \( n \) (whose degree is a function of \( K \)).

By Theorem 10 \( L_0(\bar{G}) \) is context free. Therefore containment (6.6) can be tested in polynomial time.

### 6.1 Solving system (6.5) in polynomial time

Let \( j = 1, \ldots, k \) and let \( m_j = i''_j - i_j \) and let \( n_j = i_j - i'_j \) We shall use the following renaming of the elements of \( \Sigma \) occurring in \( w = \sigma_1 \cdots \sigma_n \):

\[
\sigma_{j,1} = \sigma_{i'_j+i} \text{ if } i < i_j - i'_j
\]

and

\[
\sigma_{j,2} = \sigma_{i_j+i} \text{ if } i < i''_j - i_j
\]
In this notation, (6.5) has the following form

\[
\begin{align*}
(A_i^\ell \beta_{i_1}^\ell \tau_{1,1,1} \cdots \tau_{1,n_1-1,1} \alpha_{i_1} x \beta_{i_1}^r \tau_{1,1,1,2} \cdots \tau_{1,m_1-1,2} \alpha_{i_1}^r A_{i_1}^r)[x := \theta] & \leq 1 \\
(A_i^\ell \beta_{i_2}^\ell \tau_{2,1,1} \cdots \tau_{2,n_2-1,1} \alpha_{i_2} x \beta_{i_2}^r \tau_{2,1,2} \cdots \tau_{2,m_2-1,2} \alpha_{i_2}^r A_{i_2}^r)[x := \theta] & \leq 1 \\
& \vdots \\
(A_i^\ell \beta_{i_k}^\ell \tau_{k,1,1} \cdots \tau_{k,n_k-1,1} \alpha_{i_k} x \beta_{i_k}^r \tau_{k,1,2} \cdots \tau_{k,m_k-1,2} \alpha_{i_k}^r A_{i_k}^r)[x := \theta] & \leq 1
\end{align*}
\]

(6.7)

Next we observe that, by the definition of assignment \( I \), for each solution \( \theta \) of (6.7) and each inequation \( j, j = 1, \ldots, k \), the following holds:

Each type of the from (ii) to the left of \( x \) is of the from \( \alpha A_1^\ell \beta \) and each type of the from (ii) to the right of \( x \) is of the from \( \alpha A_1^r \beta \).

For our next observation we shall need the following notation. For a type \( \tau = \alpha A_1^\ell \beta \) of the form (ii) we denote by \( \tau' \) the type \( \tau = \alpha A_1^r \beta \) and for a type \( \tau = \alpha A_1^r \beta \) of the form (ii) or for a type \( \tau \) of the form (i), \( \tau' \) is \( \tau \) itself. Then, (6.7) if and only if

\[
\begin{align*}
\beta_{i_1} \tau_{1,1,2} \cdots \tau_{1,n_1-1,2} \alpha_{i_1}^r A_{i_1}^r \beta_{i_1}^r \tau_{1,1,1} \cdots \tau_{1,m_1-1,1} \alpha_{i_1} \theta & \leq 1 \\
\beta_{i_2} \tau_{2,1,2} \cdots \tau_{2,n_2-1,2} \alpha_{i_2}^r A_{i_2}^r \beta_{i_2}^r \tau_{2,1,1} \cdots \tau_{2,m_2-1,1} \alpha_{i_2} \theta & \leq 1 \\
& \vdots \\
\beta_{i_k} \tau_{k,1,2} \cdots \tau_{k,n_k-1,2} \alpha_{i_k}^r A_{i_k}^r \beta_{i_k}^r \tau_{k,1,1} \cdots \tau_{k,m_k-1,1} \alpha_{i_k} \theta & \leq 1
\end{align*}
\]

(6.8)

That is, the left-hand side of the \( j \)th inequation in (6.8) can be thought of as a assignment \( \bar{I} \) over the alphabet

\[\Sigma = \Sigma \cup \bar{\Sigma} \cup \Sigma' \cup \Sigma''\]

where \( \Sigma, \Sigma', \Sigma'' \), and \( \bar{I} \) are defined as follows:

\[\bar{\Sigma} = \{ \sigma : \sigma \in \Sigma \text{ is of type (ii)} \}\]

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

\[\Sigma' = \{ \sigma_{j'} : \sigma = \sigma_{j'}, j = 1, \cdots, k \}\]
is a disjoint copy of \( \{ \sigma_{i_1}', \ldots, \sigma_{i_k}' \} \), that is \( \sigma_{i_1}' = \sigma_{i_2}' \) if and only if \( i_1 = i_2 \),

\[
\Sigma'' = \{ \sigma_{i_j}' : \sigma = \sigma_{i_j}', \ j = 1, \ldots, k \}
\]
is a disjoint copy of \( \{ \sigma_{i_1}'', \ldots, \sigma_{i_k}''' \} \), that is \( \sigma_{i_1}'' = \sigma_{i_2}'' \) if and only if \( i_1'' = i_2'' \).

where the type of the elements of \( \Sigma \cup \Sigma' \cup \Sigma'' \) is \((ii)\), and

\[
T(\sigma) = \begin{cases}
I(\sigma), & \text{if } \sigma \in \Sigma \\
\{ \tau : \tau \in I(\sigma) \}, & \text{if } \bar{\sigma} \in \Sigma' \\
A_{i_j}^\ell \beta_{i_j}', & \text{if } \sigma = \sigma_{i_j}' \in \Sigma' \\
\alpha_{i_j} A_{i_j}^\ell, & \text{if } \sigma = \sigma_{i_j}'' \in \Sigma''
\end{cases}
\]

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

We have to find assignment \( \tau_{j,i} \in \overline{T}(\sigma_{j,i}) \), and \( \tau_{j,i,h} \in \overline{T}(\sigma_{j,i,h}) \) such that

\[
\begin{align*}
\beta_{i_1} \tau_{1,1,2} \cdots \tau_{1,m_1-1,2} \tau_{i_1} \tau_{1,1,1} \cdots \tau_{1,n_1-1,1} & \alpha_{i_1} \theta \leq 1 \\
\beta_{i_2} \tau_{2,1,2} \cdots \tau_{2,m_2-1,2} \tau_{i_2} \tau_{2,1,1} \cdots \tau_{2,n_2-1,1} & \alpha_{i_2} \theta \leq 1 \\
& \vdots \\
\beta_{i_k} \tau_{k,1,2} \cdots \tau_{k,m_k-1,2} \tau_{i_k} \tau_{k,1,1} \cdots \tau_{k,n_k-1,1} & \alpha_{i_k} \theta \leq 1
\end{align*}
\]  

(6.9)

Since \( \theta \in A^+ \), the symbols occurring in \( \theta \) can be canceled only by the pregroup elements of the form \( A^\ell \), and such elements can occur only in assignments to the elements of \( \Sigma \) of type \((ii)\). Therefore, the number of the elements of \( \Sigma \) of type \((ii)\) occurring in inequation in (6.8) is the same, say \( t = |\theta| \). We rewrite system (6.9) in the following way

\[
\begin{align*}
\beta_{i_1} \lambda_{1,1} \mu_{1,1} \cdots \lambda_{1,t} \mu_{1,t} \lambda_{1,t+1} & \alpha_{i_1} \theta \leq 1 \\
\beta_{i_2} \lambda_{2,1} \mu_{2,1} \cdots \lambda_{2,t} \mu_{2,t} \lambda_{2,t+1} & \alpha_{i_2} \theta \leq 1 \\
& \vdots \\
\beta_{i_k} \lambda_{k,1} \mu_{k,1} \cdots \lambda_{k,t} \mu_{k,t} \lambda_{k,t+1} & \alpha_{i_k} \theta \leq 1
\end{align*}
\]  

(6.10)

Where

- \( \mu_{j,i} \) is the \( i \)th occurrence (from the left) of elements from \( \Sigma \) where
\( \bar{I}(\mu_{j,i}) \) is of type (ii) or of the form \( A_{j}^{\ell_{i}} \beta_{j}^{\epsilon_{i}} \), \( A_{j}^{\ell_{i}} \beta_{j}^{\epsilon_{i}} \) in 
[\sigma_{j,1,2} \cdot \cdot \cdot \sigma_{j,m_{j}-1,2} \sigma_{j,m_{j}}^{\prime} \sigma_{j,1,1}^{\prime} \cdot \cdot \cdot \sigma_{j,n_{j}-1,1}^{\prime}]

- \( \lambda_{j,i} \) are all the elements of \( \Sigma \) between \( \mu_{j,i-1} \) and \( \mu_{j,i} \)

Let \( \theta = A_{1} \cdot \cdot \cdot A_{t} \). Then \( \mu_{j,1} \) is of the form \( \alpha_{j,1}^{r} A_{1} \alpha_{j,1} \), where \( \alpha_{j,1} \in \kappa(B \setminus A) \) and there exists \( \gamma_{j,1} \in \bar{I}(\lambda_{j,1}) \) such that 
\[
\begin{align*}
\beta_{1} \gamma_{1,1} \alpha_{1,1}^{r} & \leq 1 \\
\beta_{2} \gamma_{2,1} \alpha_{2,1}^{r} & \leq 1 \\
& \vdots \\
\beta_{k} \gamma_{k,1} \alpha_{k,1}^{r} & \leq 1
\end{align*}
\]
(6.11)
i.e., the “up to \( A_{1} \)” prefix in the lefthand side of each inequation reduces to 1; and there exists \( \gamma_{j,i} \in \bar{I}(\lambda_{j,i}) \) such that 
\[
\begin{align*}
\alpha_{1,i} \gamma_{1,i} \alpha_{1,i+1}^{r} & \leq 1 \\
\alpha_{2,i} \gamma_{2,i} \alpha_{2,i+1}^{r} & \leq 1 \\
& \vdots \\
\alpha_{k,i} \gamma_{k,i} \alpha_{k,i+1}^{r} & \leq 1
\end{align*}
\]
(6.12)
i.e., the subword between \( A_{i} \) and \( A_{i+1} \) in the lefthand side of each inequation reduces to 1, \( i = 1, \ldots, t-1 \); and there exists \( \gamma_{j,t+1} \in \bar{I}(\lambda_{j,t+1}) \) such that 
\[
\begin{align*}
\alpha_{1,t} \gamma_{1,t+1} \alpha_{1} & \leq 1 \\
\alpha_{2,t} \gamma_{2,t+1} \alpha_{2} & \leq 1 \\
& \vdots \\
\alpha_{k,t} \gamma_{k,t+1} \alpha_{k} & \leq 1
\end{align*}
\]
(6.13)
i.e., the “from \( A_{t} \)” suffix in the lefthand side of each inequation reduces to 1.

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

**Definition 28** Let \( \tau_{i} = \alpha_{i} A_{i}^{\ell_{i}} \beta_{i} \), \( \alpha_{i}, \beta_{i} \in \kappa(B \setminus A) \) and \( A_{i} \in A, i = 1, \ldots, k \).

We say that the set of types \( \{\tau_{j}\}_{j=1, \ldots, k} \) is **consistent**, if 
\[
A_{1} = \cdots = A_{k}.
\]

35
Consider the directed graph $G$ whose set of vertices consists of 0, $t+1$, and all $(k+1)$-tuples of the form $(p, \tau_1, \ldots, \tau_k)$, $p = 1, \ldots, t$, where $\tau_j \in \overline{I(\mu_j,p)}$ and the set of types $\{\tau_j\}_{j=1}^k$ is consistent. Note that the number of vertices of $G$ is linear in $n$.

The edges of $G$ are as follows.

- There is no edge from 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 $\gamma_{j,1} \in \overline{I(\lambda_j,1)}$, $j = 1, \ldots, k$ such that
  \[
  \begin{aligned}
  \beta_{1,1} \gamma_{1,1} \alpha_1 &\leq 1 \\
  \beta_{1,2} \gamma_{2,1} \alpha_2 &\leq 1 \\
  \vdots \\
  \beta_{k,1} \gamma_{k,1} \alpha_k &\leq 1
  \end{aligned}
  \]  
  cf. (6.14).
- There is an edge from vertex $v = (p', \alpha_1' A_{p'} \beta_1', \ldots, \alpha_k' A_{p'} \beta_k')$ to vertex $u = (p'', \alpha_1'' A_{p''} \beta_1'', \ldots, \alpha_k'' A_{p''} \beta_k'')$ if and only if $p'' = p' + 1$ and there exist $\gamma_{j,p''} \in \overline{I(\lambda_j,p'')}$, $j = 1, \ldots, k$ such that
  \[
  \begin{aligned}
  \beta_{1,p''} \gamma_{1,p''} \alpha_1'' &\leq 1 \\
  \beta_{2,p''} \gamma_{2,p''} \alpha_2'' &\leq 1 \\
  \vdots \\
  \beta_{k,p''} \gamma_{k,p''} \alpha_k'' &\leq 1
  \end{aligned}
  \]  
  cf. (6.15).
- There is an edge from vertex $(p, \alpha_1 A_p \beta_1, \ldots, \alpha_k A_p \beta_k)$ to vertex $t+1$ if and only if $p = m$ and there exist $\gamma_{j,p+1} \in \overline{I(\lambda_j,p+1)}$, $j = 1, \ldots, k$ such that
  \[
  \begin{aligned}
  \beta_{1} \gamma_{1,p+1} \alpha_{i_1} &\leq 1 \\
  \beta_{2} \gamma_{2,p+1} \alpha_{i_2} &\leq 1 \\
  \vdots \\
  \beta_{k} \gamma_{k,p+1} \alpha_{i_k} &\leq 1
  \end{aligned}
  \]  
  cf. (6.16).
Similarly to the case of (6.6), finding appropriate assignments, which satisfy the above (independent) inequations (6.14),(6.15), and (6.16) reduces to the membership problems in corresponding context-free languages. Therefore, \( \mathcal{G} \) can be constructed in polynomial time.

Since (6.5) if and only if there is a path from 0 to \( m + 1 \) in \( \mathcal{G} \), the proof of Theorem 27 is complete.
Chapter 7

Hierarchy of restricted buffer augmented pregroup grammar languages

In this chapter 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. This result is not trivial since given a restricted $BAPGG G = \langle \Sigma, B, \leq, \mathcal{A}, I, \Delta \rangle$, if a word $w \in L_K(G)$ for some $K \geq 0$ then, there exists $\tau \in I(w)$ with at most $K$ appearances of $x$ and $\theta \in \mathcal{A}^+$ such that $\tau[x := \theta] \leq \delta$ for some $\delta \in \Delta$. Since, $K$ is the upper bound on the number of type $(iii)$ categories that appear in $\tau$, if we increase this bound to $K + 1$ we might accept words that cannot be accepted using only $K$ type $(iii)$ categories. In this chapter, we show that is possible to create from any BAPGG $G$ another BAPGG $G'$ such that for any $K \geq 0$, $L_K(G) = L(G') = L_{K+1}(G')$ thus eliminating $K$ as a bound on the number of type $(iii)$ categories allowed in $\tau$.

In addition, we show that for any given $K \geq 0$ there exists a language whose words cannot be accepted using only $K$ type $(iii)$ categories and, therefore, there exists a strict hierarchy inside the family of languages that are generated by restricted BAPGGs.
7.1 Each $K$-restricted BAPGG language is also a $(K + 1)$-restricted BAPGG one

Let $G = (\Sigma, B, \leq^\prime, A, I, \Delta)$ 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 $G' = (\Sigma, B \cup \{Z\}, \leq', A, I', \{\delta Z^k : k = 0, \ldots, K\})$, where $Z \notin B$ and $\leq' = \leq \cup (Z, Z)$ and, for $\sigma \in \Sigma$, the lexical category assignment $I'(\sigma)$ is defined as follows.

- If $\sigma$ 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 $\sigma$ 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 $\tau \in I'(w)$, $\theta \in A^+$, and $k = 0, \ldots, K$ be such that

\[ \tau[x := \theta] \leq \delta Z^k. \tag{7.1} \]

Then, $\tau$ has exactly $k$ occurrences of $x$ and, substituting 1 for $Z$ in (7.1) 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 \( \tau_{ij} \in I(\sigma_{ij}) \), \( j = 1, \ldots, k \), 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, \( w \in L(G') \).

7.2 \( K \)-restricted BAPGG languages are strictly included into \((K+1)\)-restricted BAPGG languages

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 \( G_{e,3} = \{(a, b), \{B, S, T\}, =, \{B\}, I, \{T^3\}\} \), where \( I \) is defined by

- \( I(a) = \{S\} \) and
- \( I(b) = \{S^r B^k S, S^r x T\} \).

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

and, substituting \( \theta = BB \in A^+ \) for \( x \), we obtain (by (con)s)

\[
SS^r B^l SS^r B^l SS^r BBT SS^r B^l SS^r B^l SS^r B^l SS^r BBT SS^r B^l SS^r B^l SS^r BBT \leq TTT.
\]
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.
Chapter 8

Buffer Augmented Pushdown Automata

In this section, we present a new model of computation, called buffer augmented pushdown automaton, which is an extension of a pushdown automaton. The main idea behind this extension is the addition of an extra data structure, called the buffer, whose contents are “guessed” nondeterministically by the automaton as it starts the computation on an input word and remain fixed during the computation. The role of the buffer is to simulate the buffer element of the pregroup, i.e., providing the ability to “pop” from the top of the stack a portion matching some prefix of the buffer and “push” the remaining suffix of the buffer onto the stack - we call this operation buffer reset. Since the buffer’s contents remains fixed and accessible throughout the computation it is possible to compare (via the buffer) between data that appeared at the top of the stack in different stages of the computation and, thus, partially eliminating the restrictions imposed by the stack as the only data structure. This extension is reflected by additional possibilities of the transition relation involving the buffer and the stack. For example, let \( L = \{a^n b^n c^n : n \geq 0\} \) one can “guess” \( n \) in unary basis and store it inside the buffer. Next for every subword \( a^i, b^j \) and \( c^k \) of an input word \( w \) (we assume \( w \) is of the form \( a^+ b^+ c^+ \) otherwise it is rejected using the states), one can push in unary \( i-1, j-1 \) or \( k-1 \), respectively, inside the stack and reset the buffer when the automaton reads the last \( a, b \) or \( c \) respectively. Since the buffer was fixed at the beginning of the computation, the buffer would “pop”
out exactly \( n - 1 \) elements each time, thus, allowing a comparison between the number of \( a \)'s, \( b \)'s and \( c \)'s inside \( w \). Another example that demonstrates the power of the buffer is \( L = \{ww : w \in \{a, b\}^+\} \). Here one can “guess” \( w^R \) and store it in the buffer. Next, the automaton starts reading the input, while pushing it into the stack. When the automaton reaches the middle of the input it pops the buffer out of the stack thus ensuring that the first half of the input word matches \( w \). Next, the automaton repeats the above process for the second half of the input word ensuring that it also matches \( w \). Since the buffer is fixed, the content of the buffer is the same in both “pop”s. Thus, the first half of the input word matches the second one.

### 8.1 Definition of buffer augmented pushdown automata

A buffer-augmented pushdown automaton (BAPDA) is a tuple

\[
M = \langle Q, \Sigma, \Gamma, q_0, \gamma_0, \Xi, F, S, \mu \rangle
\]

where

- \( Q \) is a finite set of states,
- \( \Sigma \) is the input alphabet,
- \( \Gamma \) is the stack alphabet,
- \( q_0 \in Q \) is the initial state,
- \( \gamma_0 \in (\Gamma \setminus \Xi)^* \) is the initial stack content,
- \( \Xi \subseteq \Gamma \) is the buffer alphabet,\(^1\)
- \( F \subseteq Q \) is the set of accepting states,
- \( S \subseteq (\Gamma \setminus \Xi)^* \) is a set of accepting stack contents, and
- \( \mu = \bigcup_{i=1}^{6} \mu_i \) is the (finite) transition relation, where \( \mu_i, \, i = 1, 2, \ldots, 6 \)

are as follows.

\(^1\)Actually, the stack alphabet is \( \Gamma \cup \Xi^c \).
1. \( \mu_1 \subseteq (Q \times \Sigma \times (\Gamma \setminus \Xi)^*) \times (Q \times (\Gamma \setminus \Xi)^*) \).

These are ordinary \((\Gamma \setminus \Xi)\)-transitions which correspond to BAPGG category assignments of type \((i)\). In what follows, for

\[ ((q, \sigma, \delta'), (p, \delta'')) \in \mu_1 \]

we shall also write

\[ (p, \delta'') \in \mu_1(q, \sigma, \delta'). \]

2. \( \mu_2 \subseteq (Q \times \Sigma \times (\Gamma \setminus \Xi)^*) \times (Q \times (\Gamma \setminus \Xi)^* \Xi (\Gamma \setminus \Xi)^*) \).

These are \(\Xi\)-push transitions which correspond to BAPGG category assignments of type \((ii)\). In what follows, for

\[ ((q, \sigma, \delta), (p, \delta'A^\ell \delta'')) \in \mu_2 \]

we shall also write

\[ (p, \delta'A^\ell \delta'') \in \mu_2(q, \sigma, \delta). \]

3. \( \mu_3 \subseteq (Q \times \Sigma \times (\Gamma \setminus \Xi)^* \Xi (\Gamma \setminus \Xi)^*) \times (Q \times (\Gamma \setminus \Xi)^*). \)

These are \(\Xi\)-pop transitions which correspond to BAPGG category assignments of type \((ii)\). In what follows, for

\[ ((q, \sigma, \delta'A\delta''), (p, \delta)) \in \mu_3 \]

we shall also write

\[ (p, \delta) \in \mu_3(q, \sigma, \delta'A\delta''). \]

4. \( \mu_4 \subseteq (Q \times \Sigma \times (\Gamma \setminus \Xi)^*) \times (Q \times (\Gamma \setminus \Xi)^+ \{x\} (\Gamma \setminus \Xi)^*) \)

These are right \(\Xi\)-reset transitions which correspond to BAPGG category assignments of type \((iii)\), where \(x\) corresponds to the buffer variable. In what follows, for

\[ ((q, \sigma, \delta), (p, \delta'x\delta'')) \in \mu_4 \]
we shall also write
$$(p, \delta' x' \delta'') \in \mu_4(q, \sigma, \delta).$$

5. \[\mu_5 \subseteq (Q \times \Sigma \times (\Gamma \setminus \Xi)^+ \{x^\ell\} (\Gamma \setminus \Xi)^* \times (Q \times (\Gamma \setminus \Xi)^*)].\]

These are left \(\Xi\)-reset transitions which correspond to BAPGG category assignments of type \((iii)\). In what follows, for

$$( (q, \sigma, \delta' x' \delta''), (p, \delta) ) \in \mu_5$$

we shall also write

$$(p, \delta) \in \mu_5(q, \sigma, \delta' x' \delta'').$$

6. \[\mu_6 \subseteq (Q \times \Sigma \{x'^\ell\} (\Gamma \setminus \Xi)^* \times (Q \times \{x''\} (\Gamma \setminus \Xi)^*)].\]

These are \(\Xi\)-reset transitions which correspond to BAPGG category assignments of type \((iii)\), and the buffer content \(x\) is \(x'x''\).

In what follows, for

$$( (q, \sigma, x'^\ell \delta'), (p, x'' \delta'') ) \in \mu_6$$

we shall also write

$$(p, x'' \delta'') \in \mu_6(q, \sigma, x'^\ell \delta').$$

• In addition:

- For every \((p, \delta' \xi \delta'') \in \mu_2(q, \sigma, \delta)\) there exists

  $$(p, \xi \delta'') \in \mu_2(p, \sigma, \delta)$$

- For every \((p, \delta) \in \mu_3(q, \sigma, \delta' \xi \delta'')\) there exists

  $$(p, \delta) \in \mu_3(q, \sigma, \xi \delta'')$$

- If for an input letter \(\sigma\) there is a transition of the form \((p, \delta'') \in \mu_1(q, \sigma, \delta')\), then \(M\) cannot have transitions of the form \((p, \delta' \xi \delta'') \in \mu_2(q, \sigma, \delta)\) or \((p, \delta) \in \mu_3(q, \sigma, \delta' \xi \delta'')\)

These conditions correspond to the additional restrictions imposed on the lexicon \(I\) in Definition 11.
**Definition 29** A $\theta$-configuration, $\theta \in \Xi^*$, of a BAPDA $M = \langle Q, \Sigma, \Gamma, q_0, \gamma_0, \Xi, F, S, \mu \rangle$ is a tuple $[q, w, \gamma, \theta]$ where

- $q \in Q$ is the current state of the automaton,
- $w \in \Sigma^*$ is the input part that has not been read yet,
- $\gamma \in \Gamma^*$ is the stack content, and
- $\theta \in \Xi^*$ is the content of the buffer.

The *initial $\theta$-configuration* of $M = \langle Q, \Sigma, \Gamma, q_0, \gamma_0, \Xi, F, S, \mu \rangle$ on an input $w$ is $[q_0, w, \gamma_0, \theta]$.

In order to define the transition relation we use the notation $\gamma' = \gamma\alpha$, $\gamma'' = \gamma\beta$, etc. for a decomposition of $\gamma'$ and $\gamma''$ in the indicated form, with appropriate restrictions on the sub-alphabet from which the components are taken.

**Definition 30** For a BAPDA automaton $M = \langle Q, \Sigma, \Gamma, q_0, \gamma_0, \Xi, F, S, \mu \rangle$ we define the transition relation $\vdash_M$ as follows. Let $\theta = A_1 \cdots A_n \in \Xi^*$. Then

$[q, w', \gamma', \theta] \vdash_M [p, w'', \gamma'', \theta]$ if and only if $w' = \sigma w''$, for some $\alpha, \beta, \gamma \in \Gamma^*$, $\gamma' = \gamma\alpha$, $\gamma'' = \gamma\beta$, and the following hold:

- $((q, \sigma, \alpha), (p, \beta)) \in \mu_1 \cup \mu_2 \cup \mu_3$, or
- for some $\delta' \in \Gamma^+$ and $\delta'' \in \Gamma^*$ such that $(p, \delta'x\delta'') \in \mu_4(q, \sigma, \alpha)$, $\beta = \delta'\theta\delta''$, or
- for some $\delta' \in \Gamma^+$ and $\delta'' \in \Gamma^*$ such that $(p, \alpha) \in \mu_5(q, \sigma, \delta'x\delta'')$, $\alpha = \delta'\theta\delta''$, or
- for some $\delta', \delta'' \in \Gamma^*$ such that $(p, x''\delta'') \in \mu_6(q, \sigma, x'\delta')$ and for some $t = 0, 1, \ldots, m$, $\alpha = A_1 \cdots A_t\delta'$ and $\beta = A_{t+1} \cdots A_m\delta''$.

As usual, we denote the reflexive transitive closure of $\vdash_M$ by $\vdash^*_M$.

---

2The contents of the buffer remains intact during configuration changes.
**Definition 31** Let \( M = (Q, \Sigma, \Gamma, q_0, \Xi, F, S, \mu) \) be a BAPDA and let \( K \) be a non-negative integer. We define

\[
L_{K, \theta}(M) = \{ w : [q_0, w, \gamma_0, \theta] \vdash^* M [q, \epsilon, \gamma, \theta, k] \text{ with at most } K \text{ reset transitions}^3, q \in F \text{ and } \gamma \in S \}
\]

\[
L_{K}(M) = \bigcup_{\theta \in \Xi^+} L_{K, \theta}(M)
\]

\[
L_{\theta}(M) = \bigcup_{K} L_{K, \theta}(M)
\]

and

\[
L(M) = \bigcup_{\theta \in \Xi^+} L_{\theta}(M) = \bigcup_{K} L_{K}(M) = \bigcup_{K \geq 0} L_{K, \theta}(M)
\]

**Example 32** Let \( L_{2k} = \{(a^n b c d^m)^k : n, k \geq 1\} \). Consider the following BAPDA \( M = (\{q_0\}, \{a, b, c, d\}, \{\bot, A, U, V\}, q_0, \bot, \{\{q_0\}\}, \{\bot\}, \mu) \), where \( \mu \) is defined as follows: (defined by the order of application)

- \( \mu_2(q_0, a, \bot) = \{(q_0, \bot AU)\} \)
- \( \mu_2(q_0, a, U) = \{(q_0, AU)\} \)
- \( \mu_5(q_0, b, \bot x^f U) = \{(q_0, V)\} \)
- \( \mu_4(q_0, c, V) = \{(q_0, \bot x)\} \)
- \( \mu_3(q_0, d, A) = \{(q_0, \epsilon)\} \)

Let \( \theta = AAAAA \) we now demonstrate a \( \theta \)-run \( C_1, \ldots, C_{20} \) of \( M \) on

\[
w = aaaa b c dddd aaaa b c dddd \in L_2
\]

the initial configuration of \( M \) is

\[
C_0 = [q_0, aaaa b c dddd aaaa b c dddd, \bot, AAAAA]
\]

After reading the first \( a \) on the input, \( M \) pushes an \( AS \) in the stack using the transition \((q_0, \bot AU) \in \mu_2(q_0, a, \bot) \). Thus \( M \)'s next configuration is

\[
C_1 = [q_0, aaaa b c dddd aaaa b c dddd, \bot A^f U, AAAAA]
\]

---

3That is, transitions from \( \mu_4 \cup \mu_5 \cup \mu_6 \).
Next, when $M$ encounters the second $a$ in the input it pops $U$ from the stack then pushes another $A$ into the stack followed by $U$ using the transition $(q_0, AU) \in \mu_2(q_0, a, U)$. Thus,

$$C_2 = [q_0, aabcdddd aaaaabcdddd, \perp A^\ell A^\ell U, AAAA]$$

Next, the automaton performs the same transition as above for the third and forth $a$ in the input, Thus,

$$C_4 = [q_0, bcddddd aaaaabcdddd, \perp A^\ell A^\ell A^\ell A^\ell U, AAAA]$$

When $M$ encounters the first $b$ in the input it resets the stack to the left using $(q_0, V) \in \mu_5(q_0, b, \perp x U)$. This yields

$$C_5 = [q_0, cddddd aaaaabcdddd, V, AAAA]$$

When $M$ encounters the first $c$ in the input it resets the stack to the right using $(q_0, \perp) \in \mu_4(q_0, c, V)$. Therefore

$$C_6 = [q_0, dddddd aaaaabcdddd, \perp AAAA, AAAA]$$

When $M$ encounters the first $d$ in the input, it first pops $A$ out of the stack using $(q_0, \epsilon) \in \mu_3(q_0, d, A)$. Thus, $M$’s next configuration is

$$C_7 = [q_0, dddddd aaaaabcdddd, \perp AAAA, AAAA]$$

Similarly, when $M$ encounters the second, third and forth $d$ in the input, it pops $A$ out of the stack, using $(q_0, \epsilon) \in \mu_3(q_0, d, A)$. Thus, $M$’s next configuration is

$$C_{10} = [q_0, aaaaabcdddd, \perp, AAAA]$$

Now, the above process repeats again for the second half of the input. When $M$ encounters the fifth $a$ in the input it pushes $A$ into the stack using $(q_0, \perp) \in \mu_2(q_0, a, \perp)$, yielding

$$C_{11} = [q_0, aaabcdddd, \perp A^\ell U, AAAA]$$

Next, when $M$ encounters the sixth, seventh and eighth $a$ in the input, it applies the following transition three times. First, it pops $U$ from the stack,
then it pushes another $A$ into the stack followed by an $U$, using the transition $(q_0, AU) \in \mu_2(q_0, a, U)$. Thus,

$$C_{14} = [q_0, bcdddd, \bot A^\ell A^\ell A^\ell A^\ell A, AAAA]$$

Now, $M$ encounters the second $b$ in the input. Therefore, it resets the stack to the left using $(q_0, V) \in \mu_5(q_0, b, \bot x^e U)$. Thus,

$$C_{15} = [q_0, cdddd, V, AAAA]$$

Next, $M$ encounters the second $c$ in the input and resets the stack to the right using $(q_0, \bot x) \in \mu_4(q_0, c, V)$. Therefore,

$$C_{16} = [q_0, dddd, \bot AAAA, AAAA]$$

When $M$ encounters the fifth, sixth, seventh and eighth $d$ in the input it pops $A$ out of the stack using $(q_0, \epsilon) \in \mu_3(q_0, d, A)$ four times, which yields

$$C_{20} = [q_0, \epsilon, \bot, AAAA]$$

Since until $C_{20}$ is reached, four buffer resets have been performed, $q_0 \in F$, $\bot \in S$ and $M$ has finished scanning the input we obtain that $w \in L_4(M)$.

### 8.2 From buffer augmented pushdown automata to buffer augmented pregroup grammar

Let $M = (Q, \Sigma, \Gamma, q_0, \gamma_0, \Xi, F, S, \mu)$ be a BAPDA. Consider a BAPGG $G_M = (\Sigma, Q \cup \Gamma, \preceq, \Xi, I, \Delta)$ where $I$ and $\Delta$ in $G_M$ are defined as follows

- For $\sigma \in \Sigma$,

$$I(\sigma) = \{q^r \alpha^r \beta p : (p, \beta) \in \mu_i(q, \sigma, \alpha), \ i = 1, \ldots, 6\},$$

where

- for $i = 5$, by $(\delta'^e x \delta''^e)^r \delta^r$ we mean $\delta'^e x \delta''^e \delta^r \delta$, and

- for $i = 6$, by $(x'^e \delta'^e)^r x'' \delta''^e$ we mean $x'^e \delta'^e x'' \delta''^e$, see clause 6 of the definition of $\mu$.

49
\( \Delta = \{ q \delta \delta' q : \delta \in S \text{ and } q \in F \} \).

Since

- For every \((p, \delta' \xi \delta'') \in \mu_2(q, \sigma, \delta)\) there exists 
  \((p, \xi \delta'') \in \mu_2(p, \sigma, \delta)\)

- For every \((p, \delta) \in \mu_3(q, \sigma, \delta' \xi \delta'')\) there exists
  \((p, \delta) \in \mu_3(q, \sigma, \xi \delta'')\)

- If for an input letter \(\sigma\) there is a transition of the form \((p, \delta'') \in \mu_1(q, \sigma, \delta')\), then \(M\) cannot have transitions of the form \((p, \delta' \xi \delta'') \in \mu_2(q, \sigma, \delta)\) or \((p, \delta) \in \mu_3(q, \sigma, \delta' \xi \delta'')\)

\(G_M\) is indeed a valid buffer augmented pregroup grammar.

**Theorem 33** \(L(M) \subseteq L(G_M)\)

**Proof** The proof is by a straightforward induction on the length of \(w\). We refer the reader to Appendix A for the details of the proof.

**Theorem 34** \(L(G_M) \subseteq L(M)\)

**Proof** The proof is by a straightforward induction on the length of \(w\). We refer the reader to Appendix B for the details of the proof.

**Corollary 35** \(L(M) = L(G_M)\)

**Remark 36** It follows from the proof of Theorems 33 and 34 that \(L_K(M) = L_K(G_M)\) for every non-negative integer \(K\).
8.3 From buffer augmented pregroup grammar to buffer augmented pushdown automata

In the following we deal with one-state automata, only. Thus, in description of such automata and their configurations we shall omit their state component. That is, for a one-state BAPDA \( \langle \{q\}, \Sigma, \Gamma, q, \perp, \Xi, \{q\}, \Delta, \mu \rangle \) and its configuration \( (q, w, \gamma, \theta) \), we shall write just \( \langle \Sigma, \Gamma, \perp, \Xi, \Delta, \mu \rangle \) and \( (w, \gamma, \theta) \), respectively.

Let \( G = \langle \Sigma, \mathcal{B}, \leq, \mathcal{A}, I, \Delta \rangle \) be a BAPGG. We may assume that for each \( \sigma \in \Sigma \) the category assignment \( I(\sigma) \) the following holds.

- If \( \tau \in I(\sigma) \) and \( \tau \leq \tau' \) by \((\text{ind})s\) and \((\text{con})s\) only, then \( \tau' \in I(\sigma) \).

4 Obviously, the set of categories \( \{\tau' : \tau \leq \tau' \text{ by } (\text{ind})s \text{ only} \} \) is finite.

We may also impose a similar constraint on the set of distinguished categories \( \Delta \)

- If \( \delta \in \Delta \) and \( \delta' \leq \delta \) by \((\text{ind})s\) and \((\text{exp})s\) only, then \( \delta' \in \Delta \).

**Corollary 37** Let \( w = \sigma_1 \sigma_2 \cdots \sigma_n \in L(G) \) and let \( \tau_i \in I(\sigma_i) \), \( i = 1, 2, \ldots, n \), and \( \delta \in \Delta \) be such that \( \tau_1 \tau_2 \cdots \tau_n \leq \delta \).

Then there exist \( \tau'_i \in I(\sigma_i) \), \( i = 1, 2, \ldots, n \), and \( \delta' \in \Delta \) such that

\[
\tau'_1 \tau'_2 \cdots \tau'_n \leq \delta'
\]  

(8.1)

by \((\text{con})s\) only and no \((\text{con})\) in the derivation of (8.1) is applied to terms which occur in the same category \( \tau'_i \), \( i = 1, 2, \ldots, n \).

Consider the one-state BAPDA \( M_G = \langle \Sigma, \Gamma, \epsilon, \mathcal{A}, \Delta, \mu \rangle \), where

\[
\Gamma = \{ B^n : B \in (\mathcal{B} \setminus \mathcal{A}) \} \text{ and } |n| \leq d(\Delta \cup \bigcup_{\sigma \in \Sigma} I(\sigma)) + 1 \cup \mathcal{A}
\]

and \( \mu \) is defined as follows. Let \( \Sigma = \Sigma_{(i)} \cup \Sigma_{(ii)} \), \( \Sigma_{(i)} \) and \( \Sigma_{(ii)} \) consists of the symbols whose category assignments are of types \((i)\) and \((ii)\), respectively. Then
\[\mu_1 = \{((\sigma, \alpha^\ell), \beta) : \sigma \in \Sigma(i) \text{ and } \alpha \beta \in I(\sigma)\},\]
\[\mu_2 = \{((\sigma, \alpha^\ell), \beta) : \sigma \in \Sigma(ii), \alpha \in \kappa(B \setminus A), \text{ and } \alpha \beta = \delta'^\ell \delta'' \in I(\sigma)\},^5\]
\[\mu_3 = \{((\sigma, \alpha^\ell), \beta) : \sigma \in \Sigma, \beta \in \kappa(B \setminus A), \text{ and } \alpha \beta = \delta'^\ell \delta'' \in I(\sigma)\},\]
\[\mu_4 = \{((\sigma, \alpha^\ell), \beta) : \sigma \in \Sigma, \beta \in (\kappa(B \setminus A) \setminus \{1\}) \{x\} \kappa(B \setminus A), \text{ and } \alpha \beta \in I(\sigma)\},\]
\[\mu_5 = \{((\sigma, \alpha^\ell), \beta) : \sigma \in \Sigma, \alpha \in \kappa(B \setminus A) \{x\} (\kappa(B \setminus A) \setminus \{1\}), \text{ and } \alpha \beta \in I(\sigma)\},\]
and
\[\mu_6 = \{((\sigma, (\alpha x')^\ell, x'' \beta) : \sigma \in \Sigma, \text{ and } \alpha x \beta \in I(\sigma)\}.\]

Since
- If \(I(\sigma)\) contains type (i) categories it does not contain categories of type (ii),
- for each \(\tau = \delta A^\ell \delta' \in I(\sigma)\) there is \(\tau' = \delta A^\ell \delta'' \in I(\sigma)\) such that \(\delta'' \delta \leq 1\)
- and for each \(\tau = \delta A^\ell \delta' \in I(\sigma)\) there is \(\tau' = \delta'' A^\ell \delta' \in I(\sigma)\) such that \(\delta' \delta'' \leq 1\)

\(\mu\) is a valid transition function.

The equivalence of \(G\) and \(M_G\) follows from Corollary 37 and Theorems 38 and 39 below.

**Theorem 38** \(L(M_G) \subseteq L(G)\)

**Proof** The proof is by a straightforward induction on the length of \(w\). We refer the reader to Appendix C for the details of the proof. 

**Theorem 39** \(L(G) \subseteq L(M_G)\)

**Proof** The proof is by a straightforward induction on the length of \(w\). We refer the reader to Appendix D for the details of the proof.

**Remark 40** It follows from the proof of Theorems 38 and 39 that \(L_K(G) = L_K(M_G)\) for every non-negative integer \(K\).

---

5The symbols A, A_1, A_2, ... refer to elements of \(\mathcal{A}\).
Chapter 9

Concluding remarks

In our thesis 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 such as polynomial parsing, semi-linearity, and a pumping lemma. Furthermore, we presented a hierarchy inside the class of buffer augmented pregroup grammars. In addition, we introduced a natural automaton counterpart to buffer augmented pregroup grammars, called (restricted) buffer augmented pushdown automata which are pushdown automata augmented with only once written additional memory – the buffer. It has been shown that 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.

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

Definition 41 (Cf. Definition 11.) A $q$-buffer augmented pregroup grammar ($q$-BAPGG) is a tuple $G = (\Sigma, \mathcal{B}, \leq, \mathcal{A}, \mathcal{V}, I, \Delta)$, where the components of $G$ are as follows.

- $\Sigma$ is a finite set of terminals (the alphabet).
- $\langle \mathcal{B}, \leq \rangle$ is a partially ordered finite set.
- $\mathcal{A} \subseteq \mathcal{B}$ is the set of the buffer elements.
• $\mathcal{V} = \{x_1, \ldots, x_q\}$ is a set of variables (buffers) disjoint from $\kappa(B)$.

• $I$ is a mapping that assigns to each element of $\Sigma$ a finite set of categories from $\kappa(B \cup \mathcal{V})$ such that for all $\sigma \in \Sigma$, each $\tau \in I(\sigma)$ is of one of the following forms:
  
  (i) $\tau \in \kappa(B \setminus A)$,
  
  (ii) $\tau = \alpha A^{(\pm 1)} \beta$, where $A \in \mathcal{A}$, $\alpha, \beta \in \kappa(B \setminus A)$, or
  
  (iii) $\tau = \alpha x_i \beta$, where $\alpha, \beta \in \kappa(B \setminus A)$ and $i = 1, \ldots, q$.

In addition,

– for each $\tau = \alpha A^r \beta \in I(\sigma)$ there is $\tau' = \alpha A'^r \beta' \in I(\sigma)$ such that $\beta' \alpha \leq 1$, and

– for each $\tau = \alpha A^\ell \beta \in I(\sigma)$ there is $\tau' = \alpha'^\ell \beta \in I(\sigma)$ 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).

• $\Delta \subset \kappa(B \setminus A)$ is a finite set of distinguished categories.

The language generated by $G$ is defined by

$L(G) = \{w : \text{there exist } \tau \in I(w), \theta_i \in A^+, i = 1, \ldots, q, \text{ and } \delta \in \Delta$

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 $\theta_i$ for $x_i, i = 1, \ldots, q$, in $\tau$.

We end this thesis with an open problem related to this extension of buffer augmented pregroup grammars. Does there exist a hierarchy inside the class of “multi-buffer augmented” pregroup grammar languages based on the number of buffers used? i.e. does the class of $q$-BAPGG languages constitute a proper subclass of the $(q + 1)$-BAPGG ones?

Daniel
Appendix A. Proof of Theorem 33

The proof of Theorem 33 immediately follows from Theorem 42 below.

**Theorem 42**

Let $\theta = A_1 A_2 \cdots A_m \in \Xi^+$ and let

$$(p_1, \sigma_1 \sigma_2 \cdots \sigma_n, \gamma_1, \theta) \vdash (p_2, \sigma_2 \cdots \sigma_n, \gamma_2, \theta) \vdash \cdots \vdash (p_n, \sigma_n, \gamma_n, \theta) \vdash (p_{n+1}, \epsilon, \gamma_{n+1}, \theta),$$

where $p_1 = q_0$, $\gamma_1 = \gamma_0$, and the computation step

$$(p_i, \sigma_i \sigma_{i+1} \cdots \sigma_n, \gamma_i, \theta) \vdash (p_{i+1}, \sigma_{i+1} \cdots \sigma_n, \gamma_{i+1}, \theta),$$

$i = 1, 2, \ldots, n$, is by transition $(p_{i+1}, \beta_i) \in \mu(p_i, \sigma_i, \alpha_i)$. Then for the type assignment $\tau_i = p_i^r \alpha_i^r \beta_i p_{i+1} \in I(\sigma_i)$, $i = 1, 2, \ldots, n$,

$$\tau_1 \tau_2 \cdots \tau_n[x := \theta] \leq q_0^{r \tau} \gamma_{n+1} p_{n+1}.$$

**Proof** The proof is by induction on $n$. The basis $n = 1$ is immediate and for the and for the induction step assume that the lemma claim holds for $n$. Let $w = \sigma_1 \sigma_2 \cdots \sigma_n \sigma_{n+1} \in \Sigma^*$ and let

$$(p_1, \sigma_1 \sigma_2 \cdots \sigma_n \sigma_{n+1}, \gamma_1, \theta) \vdash (p_2, \sigma_2 \cdots \sigma_n \sigma_{n+1}, \gamma_2, \theta) \vdash \cdots \vdash (p_n, \sigma_n \sigma_{n+1}, \gamma_n, \theta)$$

$$\vdash (p_{n+1}, \sigma_{n+1}, \gamma_{n+1}, \theta) \vdash (p_{n+2}, \epsilon, \gamma_{n+2}, \theta) \quad (A.1)$$

where $p_1 = q_0$, $\gamma_1 = \gamma_0$, and the computation step

$$(p_i, \sigma_i \sigma_{i+1} \cdots \sigma_n, \gamma_i, \theta) \vdash (p_{i+1}, \sigma_{i+1} \cdots \sigma_n, \gamma_{i+1}, \theta),$$
$i = 1, 2, \ldots, n, n+1$, is by transition $(p_{i+1}, \beta_i) \in \mu(p_i, \sigma_i, \alpha_i)$. Then,

$$(p_1, \sigma_1 \sigma_2 \cdots \sigma_n, \gamma_1, \theta) \vdash (p_2, \sigma_2 \cdots \sigma_n, \gamma_2, \theta) \vdash \cdots \vdash (p_n, \sigma_n, \gamma_n, \theta) \vdash (p_{n+1}, \epsilon, \gamma_{n+1}, \theta),$$

and, by the induction hypothesis, for the type assignment $\tau_i = p_i^r \alpha_i^r \beta_i p_{i+1} \in I(\sigma_i), i = 1, 2, \ldots, n$,

$$(\tau_1 \tau_2 \cdots \tau_n)[x := \theta] \leq q_0^r \gamma_0^r \gamma_{n+1} p_{n+1}.$$ 

Since $\tau_{n+1} = p_{n+1}^r \alpha_{n+1}^r \beta_{n+1} p_{n+2}$ and $p_{n+1} \leq 1$, the proof will be complete if we show that

$$q_0^r \gamma_0^r \gamma_{n+1} (\alpha_{n+1}^r \beta_{n+1})[x := \theta] p_{n+2} \leq q_0^r \gamma_0^r \gamma_{n+2} p_{n+2}. \tag{A.2}$$

For the proof of (A.2) we shall distinguish among the following forms of

$\tau_{n+1} = p_{n+1}^r \alpha_{n+1}^r \beta_{n+1} p_{n+2} \in I(\sigma_{n+1})$.

- Assume $(p_{n+2}, \beta_{n+1}) \in \mu(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})$. That is, $\alpha_{n+1}, \beta_{n+1} \in (\Gamma \setminus \Xi)^*$, which implies

$$(\alpha_{n+1}^r \beta_{n+1})[x := \theta] = \alpha_{n+1}^r \beta_{n+1}.$$ 

It follows from (A.1) that $\gamma_{n+1}$ is of the form

$$\gamma_{n+1} = \gamma_{n+1}^\prime \alpha_{n+1}$$

Therefore,

$$q_0^r \gamma_0^r \gamma_{n+1}^\prime \alpha_{n+1}^r \beta_{n+1} p_{n+2} \leq q_0^r \gamma_0^r \gamma_{n+1}^\prime \alpha_{n+1}^r \beta_{n+1} p_{n+2} \leq q_0^r \gamma_0^r \gamma_{n+1}^\prime \beta_{n+1} p_{n+2} = q_0^r \gamma_0^r \gamma_{n+2} p_{n+2},$$

which is (A.2).

- Assume $(p_{n+2}, \beta_{n+1}) \in \mu(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})$. That is, $\alpha_{n+1} \in (\Gamma \setminus \Xi)^*$ and $\beta_{n+1} = \delta^\prime A^\prime \delta^\prime$, where $A \in \Xi, \delta^\prime, \delta^\prime \in (\Gamma \setminus \Xi)^*$. Therefore,

$$(\alpha_{n+1}^r \beta_{n+1})[x := \theta] = \alpha_{n+1}^r \beta_{n+1}.$$
It follows from (A.1) that $\gamma_{n+1}$ is of the form

$$\gamma_{n+1} = \gamma'_{n+1} \alpha_{n+1}$$

which implies

$$q_0^r \gamma_0 \gamma_{n+1} \alpha_{n+1} \beta_{n+1} p_{n+2} = q_0^r \gamma_0 \gamma_{n+1} \alpha_{n+1} \alpha_{n+1} \delta' A \delta'' p_{n+2}$$

$$\leq q_0^r \gamma_0 \gamma_{n+1} \delta' A \delta'' p_{n+2}$$

$$= q_0^r \gamma_0 \gamma_{n+1} p_{n+2},$$

which is (A.2).

• Assume $(p_{n+2}, \beta_{n+1}) \in \mu_3(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})$. That is, $\alpha_{n+1} = \delta'A \delta''$, where $A \in \Xi$, $\delta', \delta'' \in (\Gamma \setminus \Xi)^*$, and $\beta_{n+1} \in (\Gamma \setminus \Xi)^*$. Therefore,

$$(\alpha_{n+1}^r \beta_{n+1})[x := \theta] = \alpha_{n+1}^r \beta_{n+1}.$$  

It follows from (A.1) that $\gamma_{n+1}$ is of the form

$$\gamma_{n+1} = \gamma'_{n+1} \alpha_{n+1}$$

which implies

$$q_0^r \gamma_0 \gamma_{n+1} \alpha_{n+1} \beta_{n+1} p_{n+2} = q_0^r \gamma_0 \gamma_{n+1} \alpha_{n+1} \alpha_{n+1} \beta_{n+1} p_{n+2}$$

$$\leq q_0^r \gamma_0 \gamma_{n+1} \beta_{n+1} p_{n+2}$$

$$= q_0^r \gamma_0 \gamma_{n+1} p_{n+2},$$

which is (A.2).

• Assume $(p_{n+2}, \beta_{n+1}) \in \mu_4(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})$. That is, $\alpha_{n+1} \in (\Gamma \setminus \Xi)^+$ and $\beta_{n+1} = \delta' x \delta''$, where $\delta' \in (\Gamma \setminus \Xi)^+$ and $\delta'' \in (\Gamma \setminus \Xi)^*$. Therefore,

$$(\alpha_{n+1}^r \beta_{n+1})[x := \theta] = \alpha_{n+1}^r \delta' \delta''.$$  

It follows from (A.1) that $\gamma_{n+1}$ is of the form

$$\gamma_{n+1} = \gamma'_{n+1} \alpha_{n+1}$$

which implies

$$q_0^r \gamma_0 \gamma_{n+1} \alpha_{n+1} \delta' \theta \delta'' p_{n+2} = q_0^r \gamma_0 \gamma_{n+1} \alpha_{n+1} \alpha_{n+1} \delta' \theta \delta'' p_{n+2}$$

$$\leq q_0^r \gamma_0 \gamma_{n+1} \delta' \theta \delta'' p_{n+2}$$

$$= q_0^r \gamma_0 \gamma_{n+1} p_{n+2},$$

which is (A.2).
• Assume \((p_{n+2}, \beta_{n+1}) \in \mu_5(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})\). That is, \(\alpha_{n+1} = \delta' x^t \delta''\), where \(\delta' \in (\Gamma \setminus \Xi)^+\) and \(\delta'' \in (\Gamma \setminus \Xi)^*\), and \(\beta_{n+1} \in (\Gamma \setminus \Xi)^*\). Therefore,
\[
(\alpha_{n+1}^r \beta_{n+1})[x := \theta] = (\delta' x^t \delta'')^r \beta_{n+1}.
\]
It follows from (A.1) that \(\gamma_{n+1}\) is of the form
\[
\gamma_{n+1} = \gamma_{n+1}' \delta' x^t \delta''
\]
which implies
\[
q_0^r \gamma_{n+1}' (\delta' x^t \delta'')^r \beta_{n+1} p_{n+2} = q_0^r \gamma_{n+1}' \delta' x^t \delta'' (\delta' x^t \delta'')^r \beta_{n+1} p_{n+2} \leq q_0^r \gamma_{n+1}' \beta_{n+1} p_{n+2} = q_0^r \gamma_{n+1}' \beta_{n+1} p_{n+2},
\]
which is (A.2).

• Assume \((p_{n+2}, \beta_{n+1}) \in \mu_6(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})\). That is, \(\alpha_{n+1} = x^t \delta'\) and \(\beta_{n+1} = x^t \delta''\), where \(\delta', \delta'' \in (\Gamma \setminus \Xi)^*\). Therefore,
\[
(\alpha_{n+1}^r \beta_{n+1})[x := \theta] = \delta' x^t \delta''.
\]
It follows from (A.1) that \(\gamma_{n+1}\) and \(\gamma_{n+2}\) are of the forms
\[
\gamma_{n+1} = \gamma_{n+1}' A_t^f \cdots A_1^f \delta'
\]
and
\[
\gamma_{n+2} = \gamma_{n+1}' A_t+1^f \cdots A_m^f \delta''
\]
respectively, for some \(t = 0, 1, \ldots, m\). Therefore,
\[
q_0^r \gamma_{n+1}' \delta' x^t \delta'' p_{n+2} = q_0^r \gamma_{n+1}' A_t^f \cdots A_1^f \delta' x^t \delta'' A_1 \cdots A_{t+1} A_{t+1}^f \cdots A_m^f \delta'' p_{n+2} \leq q_0^r \gamma_{n+1}' A_t+1^f \cdots A_m^f \delta'' p_{n+2} = q_0^r \gamma_{n+1}' A_t+1^f \cdots A_m^f \delta'' p_{n+2},
\]
which is (A.2).
Appendix B. Proof of Theorem 34

The proof of Theorem 34 immediately follows from Theorem 43 below.

**Theorem 43** Let $\sigma_1 \sigma_2 \cdots \sigma_n \in \Sigma^+$, $\tau_i = p_i^r \alpha_i^r \beta_i p_i+1 \in I(\sigma_i)$, $i = 1, 2, \ldots, n$, where $p_1 = q_0$ and $\alpha_1 = \gamma_0$, $\theta = A_1 A_2 \cdots A_m \in \Xi^+$, and let $\gamma_{n+2} \in (\Gamma \cup \Xi')^*$ be such that

$$(\tau_1 \tau_2 \cdots \tau_n)[x := \theta] \leq q_0^r \gamma_0 \gamma_{n+1} p_{n+1}.$$

Then

1. $p_i^r = p_i$, $i = 2, 3, \ldots, n$, and
2. there exist $\gamma_i \in (\Gamma \cup \Xi')^+$, $i = 2, 3, \ldots, n$, such that

$$(q_0, \sigma_1 \sigma_2 \cdots \sigma_n, \gamma_0, \theta) \vdash (p_2, \sigma_2 \cdots \sigma_n, \gamma_2, \theta) \vdash \cdots \vdash (p_n, \sigma_n, \gamma_n, \theta) \vdash (p_{n+1}, \epsilon, \gamma_{n+1}, \theta).$$

**Proof** The first part of the theorem follows from the fact that all $p^r, p \in Q$, which appear in lexical type assignments can be canceled only from the left.

The proof of the second part of the theorem is by induction on $n$. The basis is immediate and, for the induction step, assume that the theorem claim hold for $n$.

Let $\sigma_1 \sigma_2 \cdots \sigma_n \sigma_{n+1} \in \Sigma^+$, $\tau_i = p_i^r \alpha_i^r \beta_i p_i+1 \in I(\sigma_i)$, $i = 1, 2, \ldots, n$, where $p_1 = q_0$ and $\alpha_1 = \gamma_0$, $\theta = A_1 A_2 \cdots A_m \in \Xi^+$, and let $\gamma_{n+2} \in (\Gamma \cup \Xi')^*$ be

\(^1\text{Recall that } \Xi \subset \Gamma \text{ and the stack alphabet is } \Gamma \cup \Xi'.\)
such that

\[
(\tau_1 \tau_2 \cdots \tau_n \tau_{n+1})[x := \theta] \leq q_0^r \gamma_{n+2} p_{n+2} ^2 .
\]

Let \( \gamma_{n+1} \in (\Gamma \cup \Xi)^+ \) be such that

\[
(\tau_1 \tau_2 \cdots \tau_n \tau_{n+1})[x := \theta] \leq q_0^r \gamma_{n+1} p_{n+1} p_{n+1}^r (\alpha_{n+1}^r \beta_{n+1})[x := \theta]p_{n+2}
\]

and

\[
q_0^r \gamma_{n+1} p_{n+1} p_{n+1}^r (\alpha_{n+1}^r \beta_{n+1})[x := \theta]p_{n+2} \leq q_0^r \gamma_{n+2} p_{n+2} , \tag{B.1}
\]

and all derivation steps in (B.1) (which all are (con)s) involve a term occurring in \( p_{n+1} (\alpha_{n+1}^r \beta_{n+1})[x := \theta]p_{n+2} \).

Then \( \gamma_{n+1} \in (\Gamma \cup \Xi)^+ \) and, by the induction hypothesis,

\[
(q_0, \sigma_1 \sigma_2 \cdots \sigma_n, \gamma_0, \theta) \vdash (p_2, \sigma_2 \cdots \sigma_n, \gamma_2, \theta) \vdash \cdots \vdash
\]

\[
(p_n, \sigma_n, \gamma_n, \theta) \vdash (p_{n+1}, \epsilon, \gamma_{n+1}, \theta) .
\]

Therefore,

\[
(q_0, \sigma_1 \sigma_2 \cdots \sigma_n \sigma_{n+1}, \gamma_0, \theta) \vdash (p_2, \sigma_2 \cdots \sigma_n \sigma_{n+1}, \gamma_2, \theta) \vdash \cdots \vdash
\]

\[
(p_n, \sigma_n \sigma_{n+1}, \gamma_n, \theta) \vdash (p_{n+1}, \sigma_{n+1}, \gamma_{n+1}, \theta)
\]

and it remains to show that

\[
(p_{n+1}, \sigma_{n+1} \gamma_{n+1}, \theta) \vdash (p_{n+2}, \epsilon, \gamma_{n+2}, \theta) . \tag{B.2}
\]

For the proof of (B.2) we shall distinguish among the following forms of \( \tau_{n+1} = p_{n+1}^r \alpha_{n+1}^r \beta_{n+1} p_{n+2} \in I(a_{n+1}) . \)

- Assume \( (p_{n+2}, \beta_{n+1}) \in \mu_1 (p_{n+1}, \sigma_{n+1}, \alpha_{n+1}) \). That is, \( \alpha_{n+1}, \beta_{n+1} \in (\Gamma \setminus \Xi)^* \), which implies

\[
(\alpha_{n+1}^r \beta_{n+1})[x := \theta] = \alpha_{n+1}^r \beta_{n+1} .
\]

Since \( \gamma_{n+2} \in (\Gamma \cup \Xi)^+ \) and \( \alpha_{n+1} \in (\Gamma \setminus \Xi)^* \), it follows from (B.1) that \( \gamma_{n+1} \) is of the form

\[
\gamma_{n+1} = \gamma_{n+1}^r \alpha_{n+1}
\]

and \( \alpha_{n+1} \) is canceled from the right.

60
Which, again, together with (B.1), implies

\[ q_0^r \gamma_0^{n+1} p_{n+1} \alpha_{n+1}^{r} \beta_{n+1} p_{n+2} \leq q_0^r \gamma_0^{n+1} \alpha_{n+1}^{r} \beta_{n+1} p_{n+2} \]
\[ = q_0^r \gamma_0^{n+1} \alpha_{n+1}^{r} \beta_{n+1} p_{n+2} \]
\[ \leq q_0^r \gamma_0^{n+1} \beta_{n+1} p_{n+2} \]
\[ \leq q_0^r \gamma_0^{n+2} p_{n+2}. \]

However, since \( \beta_{n+1} \in (\Gamma \setminus \Xi)^* \), no (con) applies to \( q_0^r \gamma_0^{n} \gamma_{n+1} \beta_{n+1} p_{n+2} \), and, therefore,

\[ q_0^r \gamma_0^{n} \gamma_{n+2} p_{n+2} = q_0^r \gamma_0^{n} \gamma_{n+1} \beta_{n+1} p_{n+2}. \]

That is, \( \gamma_{n+2} \) is of the form \( \gamma_{n+2} \beta_{n+1} \), and, by the above transition, we have (B.2).

- Assume \((p_{n+2}, \beta_{n+1}) \in \mu_2(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})\). That is, \( \alpha_{n+1} \in (\Gamma \setminus \Xi)^* \) and \( \beta_{n+1} = \delta' A^f \delta'' \), where \( A \in \Xi \), \( \delta', \delta'' \in (\Gamma \setminus \Xi)^* \). Therefore,

\[ (\alpha_{n+1}^{r} \beta_{n+1})[x := \theta] = \alpha_{n+1}^{r} \beta_{n+1}. \]

Since \( \gamma_{n+2} \in (\Gamma \cup \Xi)^* \) and \( \alpha_{n+1} \in (\Gamma \setminus \Xi)^* \), it follows from (B.1) that \( \gamma_{n+1} \) is of the form

\[ \gamma_{n+1} = \gamma_{n+1}^{r} \alpha_{n+1} \]

and \( \alpha_{n+1} \) is canceled from the right. Which, again, together with (B.1), implies

\[ q_0^r \gamma_0^{n} \gamma_{n+1} p_{n+1} \alpha_{n+1}^{r} \beta_{n+1} p_{n+2} \leq q_0^r \gamma_0^{n} \gamma_{n+1}^{r} \alpha_{n+1}^{r} \beta_{n+1} p_{n+2} \]
\[ = q_0^r \gamma_0^{n} \gamma_{n+1}^{r} \alpha_{n+1}^{r} \beta_{n+1} p_{n+2} \]
\[ \leq q_0^r \gamma_0^{n} \gamma_{n+1}^{r} \delta' A^f \delta'' p_{n+2} \]
\[ \leq q_0^r \gamma_0^{n} \gamma_{n+2} p_{n+2}. \]

However, since \( \delta', \delta'' \in (\Gamma \setminus \Xi)^* \), no (con) applies to \( q_0^r \gamma_0^{n} \gamma_{n+1}^{r} \delta' A^f \delta'' p_{n+2} \), and, therefore,

\[ q_0^r \gamma_0^{n} \gamma_{n+2} p_{n+2} = q_0^r \gamma_0^{n} \gamma_{n+1}^{r} \delta' A^f \delta'' p_{n+2}. \]

That is, \( \gamma_{n+2} \) is of the form \( \gamma_{n+2} \beta_{n+1} \) and, by the above transition, we have (B.2).
• Assume \((p_{n+2}, \beta_{n+1}) \in \mu_3(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})\). That is, \(\alpha_{n+1} = \delta'A\delta''\), where \(A \in \Xi\), \(\delta', \delta'' \in (\Gamma \setminus \Xi)^*\), and \(\beta_{n+1} \in (\Gamma \setminus \Xi)^*\). Therefore,

\[
(\alpha_{n+1} \beta_{n+1})[x := \theta] = \alpha_{n+1} \beta_{n+1}.
\]

Since \(\gamma_{n+2} \in (\Gamma \cup \Xi)^*\) and \(\alpha_{n+1} \in (\Gamma \setminus \Xi)^* (\Gamma \setminus \Xi)^*\), it follows from (B.1) that \(\gamma_{n+1}\) is of the form

\[
\gamma_{n+1} = \gamma_{n+1}^' \alpha_{n+1}
\]

and \(\delta'A\delta''\) is canceled from the right. Which, again, together with (B.1), implies

\[
q_r^r \gamma_{n+1} \alpha_{n+1} \beta_{n+1} \leq q_r^r \gamma_{n+1} \alpha_{n+1} \beta_{n+1} + q_r^r \gamma_{n+1} \alpha_{n+1} \beta_{n+1} p_{n+2} = q_r^r \gamma_{n+1} \alpha_{n+1} \beta_{n+1} p_{n+2}.
\]

However, since \(\beta_{n+1} \in (\Gamma \setminus \Xi)^*\), no (con) applies to \(q_r^r \gamma_{n+1} \alpha_{n+1} \beta_{n+1} p_{n+2}\), and, therefore,

\[
q_r^r \gamma_{n+1} \beta_{n+1} p_{n+2} = q_r^r \gamma_{n+1} \beta_{n+1} p_{n+2}.
\]

That is, \(\gamma_{n+2}\) is of the form \(\gamma_{n+2}^' \beta_{n+1}\) and, by the above transition, we have (B.2).

• Assume \((p_{n+2}, \beta_{n+1}) \in \mu_4(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})\). That is, \(\alpha_{n+1} \in (\Gamma \setminus \Xi)^*\) and \(\beta_{n+1} = \delta' x \delta''\), where \(\delta' \in (\Gamma \setminus \Xi)^+\) and \(\delta'' \in (\Gamma \setminus \Xi)^*\). Therefore,

\[
(\alpha_{n+1} \beta_{n+1})[x := \theta] = (\alpha_{n+1} \delta' x \delta'')[x := \theta]
\]

\[
= \alpha_{n+1} \delta' \theta \delta''
\]

Since \(\gamma_{n+2} \in (\Gamma \cup \Xi)^*\) and \(\alpha_{n+1} \in (\Gamma \setminus \Xi)^*\), it follows from (B.1) that \(\gamma_{n+1}\) is of the form

\[
\gamma_{n+1} = \gamma_{n+1}^' \alpha_{n+1}
\]

and \(\alpha_{n+1}\) is canceled from the right.
Which, again, together with (B.1), implies

\[
q_0^r \gamma_0 \gamma_{n+1} p_{n+1}^r \alpha_{n+1}^r \delta' \theta \delta'' p_{n+2} \leq q_0^r \gamma_0 \gamma_{n+1} \alpha_{n+1}^r \delta' \theta \delta'' p_{n+2} \\
= q_0^r \gamma_0 \gamma_{n+1} \alpha_{n+1}^l \alpha_{n+1}^r \delta' \theta \delta'' p_{n+2} \\
\leq q_0^r \gamma_0 \gamma_{n+2} p_{n+2}.
\]

However, since \( \delta' \in (\Gamma \setminus \Xi)^+ \), no (con) applies to \( q_0^r \gamma_0 \gamma_{n+1} \delta' \theta \delta'' p_{n+2} \), and, therefore,

\[
q_0^r \gamma_0 \gamma_{n+2} p_{n+2} = q_0^r \gamma_0 \gamma_{n+1} \delta' \theta \delta'' p_{n+2}
\]

That is, \( \gamma_{n+2} \) is of the form \( q_0^r \gamma_0 \gamma_{n+1} \delta' \theta \delta'' \) and, by the above transition, we have (B.2).

- Assume \((p_{n+2}, \beta_{n+1}) \in \mu_5(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})\). That is, \( \alpha_{n+1} = \delta' x^l \theta'' \), where \( \delta' \in (\Gamma \setminus \Xi)^+ \), \( \theta'' \in (\Gamma \setminus \Xi)^* \), and \( \beta_{n+1} \in (\Gamma \setminus \Xi)^* \). Therefore,

\[
(\alpha_{n+1}^r \beta_{n+1})[x := \theta] = ((\delta' x^l \theta'')^r \beta_{n+1})[x := \theta] = (\delta' \theta \delta'')^r \beta_{n+1}
\]

Since \( \gamma_{n+2} \in (\Gamma \cup \Xi')^* \), \( \delta' \in (\Gamma \setminus \Xi)^+ \) and \( \theta'' \in (\Gamma \setminus \Xi)^* \), it follows from (B.1) that \( \gamma_{n+1} \) is of the form

\[
\gamma_{n+1} = \gamma_{n+1}^l \delta' \theta \delta''
\]

and \( \delta' \theta \delta'' \) is canceled from the right. Which, again, together with (B.1), implies

\[
q_0^r \gamma_0 \gamma_{n+1} p_{n+1}^r \beta_{n+1}^r (\delta' \theta \delta'')^r \beta_{n+1} p_{n+2} \leq q_0^r \gamma_0 \gamma_{n+1} (\delta' \theta \delta'')^r \beta_{n+1} p_{n+2} \\
= q_0^r \gamma_0 \gamma_{n+1} \delta' \theta \delta'' (\delta' \theta \delta'')^r \beta_{n+1} p_{n+2} \\
\leq q_0^r \gamma_0 \gamma_{n+1} \beta_{n+1} p_{n+2} \\
\leq q_0^r \gamma_0 \gamma_{n+2} p_{n+2}.
\]

However, since \( \beta_{n+1} \in (\Gamma \setminus \Xi)^* \), no (con) applies to \( q_0^r \gamma_0 \gamma_{n+1} \beta_{n+1} p_{n+2} \), and, therefore,

\[
(q_0^r \gamma_0 \gamma_{n+2} p_{n+2})[x := \theta] = (q_0^r \gamma_0 \gamma_{n+1})[x := \theta] \beta_{n+1} p_{n+2}
\]

That is, \( \gamma_{n+2} \) is of the form \( \gamma_{n+2}^l \beta_{n+1} \) and, by the above transition, we have (B.2).
• Assume \((p_{n+2}, \beta_{n+1}) \in \mu_6(p_{n+1}, \sigma_{n+1}, \alpha_{n+1})\). That is, \(\alpha_{n+1} = x^{\ell}\delta'\) and \(\beta_{n+1} = x^m\delta''\), where \(\delta', \delta'' \in (\Gamma \setminus \Xi)^*\). Therefore,

\[
\begin{align*}
(\alpha_{n+1} \beta_{n+1})[x := \theta] &= ((x^{\ell}\delta') x^m\delta'')[x := \theta] \\
&= 2(\delta'\theta\delta'')[x := \theta] \\
&= \delta''\delta''
\end{align*}
\]

Since \(\gamma_{n+2} \in (\Gamma \cup \Xi)^*\) and \(\delta'\in (\Gamma \setminus \Xi)^*\), it follows from (B.1) that \(\gamma_{n+1}\) is of the form

\[
\gamma_{n+1} = \gamma'_{n+1}\delta'
\]

and \(\delta'\theta\delta''\) is canceled from the right. This, again, together with (B.1), implies

\[
\begin{align*}
q_0 \gamma_0 \gamma_{n+1} \delta' \theta \delta'' p_{n+2} &\leq q_0 \gamma_0 \gamma_{n+1} \delta'' \theta \delta'' p_{n+2} \\
&= q_0 \gamma_0 \gamma_{n+1} \delta' \delta'' A_1 A_2 \ldots A_m \delta'' p_{n+2} \\
&\leq q_0 \gamma_0 \gamma_{n+1} A_1 A_2 \ldots A_m \delta'' p_{n+2} \\
&\leq q_0 \gamma_0 \gamma_{n+1} p_{n+2}.
\end{align*}
\]

Since \(\gamma'_{n+1} \in (\Gamma \cup \Xi)^+\), all (con)s in the derivation

\[
\begin{align*}
q_0 \gamma_0 \gamma_{n+1} A_1 A_2 \ldots A_m \delta'' p_{n+2} \leq q_0 \gamma_0 \gamma_{n+1} p_{n+2}
\end{align*}
\]

cancel some prefix \(A_1 A_2 \ldots A_t\) of \(A_1 A_2 \ldots A_m = \theta\). Therefore, \(\gamma'_{n+1}\) is of the form

\[
\gamma'_{n+1} = \gamma''_t A^{\ell}_1 \ldots A^{\ell}_t
\]

which, in turn, implies that \(\gamma_{n+1}\) and \(\gamma_{n+2}\) are of the forms

\[
\begin{align*}
\gamma_{n+1} &= \gamma''_t A^{\ell}_1 \ldots A^{\ell}_2 A^{\ell}_1 \delta'' \\
\gamma'_{n+1} \\
\gamma_{n+2} &= \gamma''_t A^{\ell}_1 \ldots A_m \delta''
\end{align*}
\]

respectively. Thus, by the above transition, we have (B.2). and this completes our proof.

\footnote{Recall that \((x^{\ell}\delta') x^m \delta'' = \delta'' x\delta''\)}
Appendix C. Proof of Theorem 38

The proof of Theorem 38 immediately follows form Theorem 44 below.

Theorem 44 Let $\theta = A_1 A_2 \cdots A_m \in \Xi^+$ and let

$$(\sigma_1 \sigma_2 \cdots \sigma_n, \gamma_1, \theta) \vdash (\sigma_2 \cdots \sigma_n, \gamma_2, \theta) \vdash \cdots \vdash (\sigma_n, \gamma_n, \theta) \vdash (\epsilon, \gamma_{n+1}, \theta),$$

where $\gamma_1 = \epsilon$ and the computation step

$$(\sigma_i \sigma_2 \cdots \sigma_n, \gamma_i, \theta) \vdash (\sigma_{i+1} \cdots \sigma_n, \gamma_{i+1}, \theta),$$

$i = 1, 2, \ldots, n$, is by transition $\beta_i \in \mu(\sigma_i, \alpha_i)$. Then for the category assignment $\tau_i = \alpha_i \beta_i \in I(\sigma_i), i = 1, 2, \ldots, n,$

$$(\tau_1 \tau_2 \cdots \tau_n)[x := \theta] \leq \gamma_{n+1}.$$

Proof The proof is by induction on $n$. The basis $n = 1$ is immediate and for the and for the induction step assume that the lemma claim holds for $n$. Let $w = \sigma_1 \sigma_2 \cdots \sigma_n \sigma_{n+1} \in \Sigma^*$ and let

$$(\sigma_1 \sigma_2 \cdots \sigma_n \sigma_{n+1}, \gamma_1, \theta) \vdash (\sigma_2 \cdots \sigma_n \sigma_{n+1}, \gamma_2, \theta) \vdash \cdots \vdash (\sigma_n \sigma_{n+1}, \gamma_n, \theta) \vdash (\sigma_{n+1}, \gamma_{n+1}, \theta) \vdash (\epsilon, \gamma_{n+2}, \theta) \quad (C.1)$$

where $\gamma_1 = \epsilon$, and the computation step

$$(\sigma_1 \sigma_2 \cdots \sigma_n, \gamma_i, \theta) \vdash (\sigma_2 \cdots \sigma_n, \gamma_{i+1}, \theta),$$
\(i = 1, 2, \ldots, n, n + 1\), is by transition \((\beta_i) \in \mu(\sigma_i, \alpha_i)\). Then,

\[(\sigma_1 \sigma_2 \cdots \sigma_n, \gamma_1, \theta) \vdash (\sigma_2 \cdots \sigma_n, \gamma_2, \theta) \vdash \cdots \vdash (\sigma_n, \gamma_n, \theta) \vdash (p_{n+1}, \epsilon, \gamma_{n+1}, \theta)\]

and, by the induction hypothesis, for the category assignment \(\tau_i = \alpha_i \beta_i \in I(\sigma_i), i = 1, 2, \ldots, n\),

\[(\tau_1 \tau_2 \cdots \tau_n)[x := \theta] \leq \gamma_{n+1}.\]

Since \(\tau_{n+1} = \alpha_{n+1} \beta_{n+1}\), the proof will be complete if we show that

\[\gamma_{n+1}(\alpha_{n+1} \beta_{n+1})[x := \theta] \leq \gamma_{n+2}.\]

(C.2)

For the proof of (C.2) we shall distinguish among the following forms of \(\tau_{n+1} = \alpha_{n+1} \beta_{n+1} \in I(a_{n+1})\).

- Assume \(\beta_{n+1} \in \mu_1(a_{n+1}, \alpha_{n+1}^\ell)\). That is, \(\alpha_{n+1}, \beta_{n+1} \in (B \setminus A)^*\), which implies

\[(\alpha_{n+1} \beta_{n+1})[x := \theta] = \alpha_{n+1} \beta_{n+1}.\]

It follows from (C.1) that \(\gamma_{n+1}\) is of the form \(\gamma_{n+1}^\ell \alpha_{n+1}^\ell\). Therefore,

\[\gamma_{n+1} \alpha_{n+1} \beta_{n+1} = \gamma_{n+1}^\ell \alpha_{n+1}^\ell \alpha_{n+1} \beta_{n+1} \leq \gamma_{n+1}^\ell \beta_{n+1} = \gamma_{n+2}\]

which is (C.2).

- Assume \(\beta_{n+1} \in \mu_2(a_{n+1}, \alpha_{n+1}^\ell)\). That is, \(\alpha_{n+1} \in (B \setminus A)^*\) and \(\beta_{n+1} = \delta'A^\ell \delta''\), where \(A \in A, \delta', \delta'' \in (B \setminus A)^*\). Therefore,

\[(\alpha_{n+1} \beta_{n+1})[x := \theta] = \alpha_{n+1} \beta_{n+1}.\]

It follows from (C.1) that \(\gamma_{n+1}\) is of the form \(\gamma_{n+1}^\ell \alpha_{n+1}^\ell\), which implies

\[\gamma_{n+1} \alpha_{n+1} \beta_{n+1} = \gamma_{n+1}^\ell \alpha_{n+1}^\ell \alpha_{n+1} \delta'A^\ell \delta'' \leq \gamma_{n+1}^\ell \delta'A^\ell \delta'' = \gamma_{n+2}\]

which is (C.2).
• Assume $\beta_{n+1} \in \mu_3(a_{n+1}, \alpha_{n+1})$. That is, $\alpha_{n+1} = \delta' A' \delta''$, where $A \in \mathcal{A}$, $\delta', \delta'' \in (B \setminus \mathcal{A})^*$, and $\beta_{n+1} \in (B \setminus \mathcal{A})^*$. Therefore,

$$(\alpha_{n+1} \beta_{n+1})[x := \theta] = \alpha_{n+1} \beta_{n+1}$$

It follows from (C.1) that $\gamma_{n+1}$ is of the form $\gamma'_{n+1} \alpha_{n+1} \beta_{n+1}$, which implies

$$\gamma_{n+1} \alpha_{n+1} \beta_{n+1} = \gamma'_{n+1} \alpha_{n+1} \beta_{n+1}$$

$$\leq \gamma'_{n+1} \beta_{n+1}$$

$$= \gamma_{n+2}$$

which is (C.2).

• Assume $(\beta_{n+1}) \in \mu_4(a_{n+1}, \alpha_{n+1})$. That is, $\alpha_{n+1} \in (B \setminus \mathcal{A})^*$ and $\beta_{n+1} = \delta' x \delta''$, where $\delta' \in (B \setminus \mathcal{A})^+$ and $\delta'' \in (B \setminus \mathcal{A})^*$. Therefore,

$$(\alpha_{n+1} \beta_{n+1})[x := \theta] = \alpha_{n+1} \delta' \delta''.$$ 

It follows from (C.1) that $\gamma_{n+1}$ is of the form $\gamma'_{n+1} \alpha_{n+1} \ell$, which implies

$$\gamma_{n+1} \alpha_{n+1} \delta' \delta'' = \gamma'_{n+1} \alpha_{n+1} \delta' \delta''$$

$$\leq \gamma'_{n+1} \delta' \delta''$$

$$= \gamma_{n+2},$$

which is (C.2).

• Assume $\beta_{n+1} \in \mu_5(a_{n+1}, \alpha_{n+1})$. That is, $\alpha_{n+1} = \delta' x \delta''$, where $\delta' \in (B \setminus \mathcal{A})^*$ and $\delta'' \in (B \setminus \mathcal{A})^+$, and $\beta_{n+1} \in (B \setminus \mathcal{A})^*$. Therefore,

$$(\alpha_{n+1} \beta_{n+1})[x := \theta] = (\delta' \delta'' \beta_{n+1}).$$

It follows from (C.1) that $\gamma_{n+1}$ is of the form $(\gamma'_{n+1} \delta' \delta'' \ell)$ which implies

$$\gamma_{n+1} (\delta' \delta'' \ell) \beta_{n+1} = \gamma'_{n+1} (\delta' \delta'' \ell \delta'' \beta_{n+1})$$

$$\leq \gamma'_{n+1} \beta_{n+1}$$

$$= \gamma_{n+2},$$

which is (C.2).
Assume $\beta_{n+1} \in \mu_6(a_{n+1}, \alpha_{n+1})$. That is, $\alpha_{n+1} = \delta'x'$ and $\beta_{n+1} = x''\delta''$, where $\delta', \delta'' \in (B \setminus A)^*$. Therefore,

$$(\alpha_{n+1}\beta_{n+1})[x := \theta] = \delta'\theta\delta''.$$  

It follows from (C.1) that $\gamma_{n+1}$ and $\gamma_{n+2}$ are of the forms

$$\gamma_{n+1} = \gamma_{n+1}'A_{t}^\ell \cdots A_{1}^\ell\delta'$$

and

$$\gamma_{n+2} = \gamma_{n+1}'A_{t+1} \cdots A_{m}\delta''$$

respectively, for some $t = 0, 1, \ldots, m$. Therefore,

$$\gamma_{n+1} \delta'\theta\delta'' = \gamma_{n+1}'A_{t}^\ell \cdots A_{1}^\ell\delta'\delta'A_{1} \cdots A_{t+1} \cdots A_{m}\delta''$$

$$\leq \gamma_{n+1}'A_{t+1} \cdots A_{m}\delta''$$

$$= \gamma_{n+2},$$

which is (C.2).
Appendix D. Proof of Theorem 39

The proof of Theorem 39 immediately follows form Theorem 45 below.

**Theorem 45** Let \( \sigma_1 \sigma_2 \cdots \sigma_n \in \Sigma^+ \), \( \tau_i = I(\sigma_i), i = 1, 2, \ldots, n \), \( \theta = A_1 A_2 \cdots A_m \in A^+ \), and let \( \gamma_{n+1} \in (\Gamma \cup A^l)^* \) be such that

\[
(\tau_1 \tau_2 \cdots \tau_n)[x := \theta] \leq \gamma_{n+1}.
\]

Then there exist \( \gamma_i \in (\Gamma \cup A^l)^* \), \( i = 2, 3, \ldots, n \), such that

\[
(a_1 a_2 \cdots a_n, \epsilon, \theta) \vdash (a_2 \cdots a_n, \gamma_2, \theta) \vdash \cdots \vdash (a_n, \gamma_n, \theta) \vdash (\epsilon, \gamma_{n+1}, \theta).
\]

**Proof** The proof is by induction on \( n \). The basis is immediate and, for the induction step, assume that the proposition claim hold for \( n \).

Let \( \sigma_1 \sigma_2 \cdots \sigma_n \sigma_{n+1} \in \Sigma^+ \), \( \tau_i = I(\sigma_i), i = 1, 2, \ldots, n \), where \( \theta = A_1 A_2 \cdots A_m \in A^+ \), and let \( \gamma_{n+2} \in (\Gamma \cup A^l)^* \) be such that

\[
(\tau_1 \tau_2 \cdots \tau_{n+1})[x := \theta] \leq \gamma_{n+2}.
\]

Let \( \gamma_{n+1} \in (\Gamma \cup A^l)^+ \) be such that

\[
(\tau_1 \tau_2 \cdots \tau_n \tau_{n+1})[x := \theta] \leq \gamma_{n+1} \tau_{n+1}[x := \theta],
\]

and

\[
\gamma_{n+1} \tau_{n+1}[x := \theta] \leq \gamma_{n+2}; \quad (D.1)
\]
and all derivation steps in (D.1) involve a term occurring in $\tau_{n+1}[x := \theta]$. Then $\gamma_{n+1} \in (\Gamma \cup A^\ell)^*$ and, by the induction hypothesis,

$$(q_0, \sigma_1 \sigma_2 \cdots \sigma_n, \bot, \theta) \vdash (\sigma_2 \cdots \sigma_n, \gamma_2, \theta) \vdash \cdots \vdash (\sigma_n, \gamma_n, \theta) \vdash (\epsilon, \gamma_{n+1}, \theta).$$

Therefore,

$$(q_0, \sigma_1 \sigma_2 \cdots \sigma_n \sigma_{n+1}, \bot, \theta) \vdash (\sigma_2 \cdots \sigma_n \sigma_{n+1}, \gamma_2, \theta) \vdash \cdots \vdash (\sigma_n \sigma_{n+1}, \gamma_n, \theta) \vdash (\sigma_{n+1}, \gamma_{n+1}, \theta)$$

and it remains to show that

$$(\sigma_{n+1} \gamma_{n+1}, \theta) \vdash (\epsilon, \gamma_{n+2}, \theta).$$

(D.2)

For the proof of (D.2) we shall distinguish among the following types of $\tau_{n+1} \in I(\sigma_{n+1})$.

- Assume $\tau_{n+1}$ is of type (i). That is, $\tau_{n+1} \in (\Gamma \setminus A)^*$, which implies

$$\tau_{n+1}[x := \theta] = \tau_{n+1}.$$ 

By Corollary 37, we may assume that all derivation steps in (D.1) are by (con)s, only. By the same corollary, each derivation step in (D.1) involves a term occurring in $\tau_{n+1}$ and a term occurring in $\gamma_{n+1}$. Therefore, $\tau_{n+1}$ is of the form

$$\tau_{n+1} = \alpha \beta$$

$\gamma_{n+1}$ is of the from

$$\gamma_{n+1} = \gamma \alpha^\ell$$

where $\alpha^\ell$ is canceled from the right and $\gamma_{n+2}$ is of the from

$$\gamma_{n+2} = \gamma \beta$$

Thus, (D.2) is by the transition $\beta \in \mu_1(\sigma_{n+1}, \alpha^\ell)$.

- Assume $\tau_{n+1}$ is of type (ii). Then either $\tau_{n+1} = \delta' A^\ell \delta''$ or $\tau_{n+1} = \delta' A^r \delta''$, and in any case,

$$\tau_{n+1}[x := \theta] = \tau_{n+1}.$$
In the former case, by Corollary 37, we may assume that all derivation steps in (D.1) are by (con)s, only. By the same corollary, each derivation step in (D.1) involves a term occurring in $\tau_{n+1}$ and a term occurring in $\gamma_{n+1}$. Since $A^\ell$ can be canceled only from the right, $\tau_{n+1}$ is of the form

$$\tau_{n+1} = \alpha \beta$$

where $\alpha \in (\Gamma \setminus A)^*$, $\gamma_{n+1}$ is of the from

$$\gamma_{n+1} = \gamma \alpha^\ell$$

where $\alpha^\ell$ is canceled from the right and $\gamma_{n+2}$ is of the from

$$\gamma_{n+2} = \gamma \beta$$

Thus, (D.2) is by the transition $\beta \in \mu_2(\sigma_{n+1}, \alpha^\ell)$.

Similarly, in the latter case, by Corollary 37, we may assume that all derivation steps in (D.1) are by (con)s, only. By the same corollary, each derivation step in (D.1) involves a term occurring in $\tau_{n+1}$ and a term occurring in $\gamma_{n+1}$. Since $A^r$ can be canceled only from the left, $\tau_{n+1}$ is of the form

$$\tau_{n+1} = \alpha \beta$$

where $\beta \in (\Gamma \setminus A)^*$, $\gamma_{n+1}$ is of the from

$$\gamma_{n+1} = \gamma \alpha^\ell$$

where $\alpha^\ell$ is canceled from the right and $\gamma_{n+2}$ is of the from

$$\gamma_{n+2} = \gamma \beta$$

Thus, (D.2) is by the transition $\beta \in \mu_3(\sigma_{n+1}, \alpha^\ell)$.

- Assume $\tau_{n+1}$ is of type (iii). Then $\tau_{n+1} = \delta'x\delta''$, where $\delta', \delta'' \in (\Gamma \setminus A)^*$. Therefore,

$$\tau_{n+1}[x := \theta] = \delta' \theta \delta''.$$  

By Corollary 37, we may assume that all derivation steps in (D.1) are
by (con)s, only. By the same corollary, each derivation step in (D.1) involves a term occurring in \( \delta'\theta\delta'' \) and a term occurring in \( \gamma_{n+1} \). Thus, \( \delta'\theta\delta'' \) is of the form \( \alpha'\beta' \), \( \gamma_{n+1} \) is of the form
\[
\gamma_{n+1} = \gamma'_{n+1}\alpha'^\ell
\]
and \( \gamma_{n+2} \) is of the form
\[
\gamma_{n+2} = \gamma'_{n+1}\beta'
\]
We shall distinguish among the three following cases.

1. \( \alpha' \in (\Gamma \setminus \mathcal{A})^* \)
and
\( \beta' \in (\Gamma \setminus \mathcal{A})^+ \{\theta\} (\Gamma \setminus \mathcal{A})^* \)
let
\( \beta \in (\Gamma \setminus \mathcal{A})^+ \{x\} (\Gamma \setminus \mathcal{A})^* \)
be such that \( \beta' = \beta[x := \theta] \). Then, (D.2) is by the transition
\[
\beta \in \mu_4(\sigma_{n+1}, \alpha^\ell)
\]

2. \( \alpha' \in (\Gamma \setminus \mathcal{A})^* \{\theta\} (\Gamma \setminus \mathcal{A})^+ \)
and
\( \beta' \in (\Gamma \setminus \mathcal{A})^* \)
let
\( \alpha \in (\Gamma \setminus \mathcal{A})^* \{x\} (\Gamma \setminus \mathcal{A})^+ \)
be such that \( \alpha' = \alpha[x := \theta] \). Then, (D.2) is by the transition
\[
\beta \in \mu_5(\sigma_{n+1}, \alpha^\ell)
\]

3. \( \alpha' = \delta' A_1 \cdots A_t \)
and
\( \beta' = A_{t+1} \cdots A_m \delta'' \)
for some \( t = 0, 1, \cdots, m \). Then, (D.2) is by the transition
\[
x''\delta'' \in \mu_6(\sigma_{n+1}, (x'\delta')^\ell)
\]

72
Bibliography


לקסיקלייזציה רדיקלית של שפת עדינות הkishar

דניאל冈קי

Technion - Computer Science Department - M.Sc. Thesis MSC-2010-22 - 2010
לקסיקלייזציה רדיקלית של שפות
עדינות הקשורים

הזור על מחקר

לשם مليוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעי מחשב

דניאל冈קי

הונש לוגט הסכמינו – מכון טכנולוגי לישראל
נכתב התutherland
דצמבר 2010
המחקרים נעשים בחנויות פורפ' MiaCal כימיקי בפקולטה למדייני המחשב.

אני מודע לתנאים על התמיכה הבספית הدية בברשתלווניא.

Technion - Computer Science Department - M.Sc. Thesis MSC-2010-22 - 2010
תקציר

dקדוקי קדד推薦 גוורה ושפות פורמליות

מאז התקופה ב- [21] על ידי למבר, דקדוקי קדד בהיותו מי שיבוח תochen ודריך הקדד. השפות הפורמליות (המסות, בחינה) הביעה הקסיקליזציה הדידית ותורת השפות הפורמליות התפשטה לתחום בבלוקו. מהמשאמונים של התוכנה סופר את שפה טיפונית, לפיה כלליו מתכוננים לפני של הדקדוק הפורמלי. הנשיהםagina להיעשות עם התוכנה דויסי, אשר מסופנים את כל ה כאילוIGHL של החלקויים. הליך. התוכנות הגישה ברדיף הפניות עבורה, בponsors מהמשאמונים של התוכנה. הלכתיות שלה של השפות וכתה. הנחיה של תוכנות כ损耗 by means of שפות פורמליות בקרוב גם הוא שהיינו לשון בין דקדוקים המבוססים על כלליית ניובוט. דקדוקים

לנושה סטנדרטיות, המבוססות על כלליית ניובוט, יש כמם מאפיינים בסיסיים:

• תרימיתם הם שיווט את והתפוצה של התוכנה של התוכנה בינוון בכלי.
• לא-תרימיות (הenty 피ין ובחר מִשְׁטְהוּ) חשבה על התוכנה התוכנה משל התוכנה.
• המסה עם קבוצה של תרימיות.
• הבילדיין ב by theほか תקובה של התוכנה לשון (לפי קביעה של התוכנה לשון תרימיות) בקבעה עד ידי.
• הבילדיין דקדוקים, לכל בילדיין דקדוקים ומלועש בילדיין בילדיין בשון של דקדוקים.

שריף מהמשאות מהפעולה היהיה.

בלשנותبرشיות מדרינות היא מכור של בשנות אתatching. הפסקה ומבוססות על ראייה.

שונה של תורת השפות (הסובוט והפרiciary) וידעה בשם קסיקליזציה הדידית.
There are several approaches to natural language processing based on lexicology: we shift the attention to the analysis of formal languages (but all of them share several common features:

1. Terminals are entities with unique characteristics. Features of these entities are defined in a lexicon that maps terminals to "pieces of information" about them, and therefore, the lexicon is in fact a dictionary. Usually, these pieces of information are finite sets of complex structures) categories.

2. Similarly, categories are represented as entities with a structure that represents finite sets of strings.

Differences between languages are defined with respect to their lexicons. There is a universal grammar (shared by all languages) that expands the lexicon to include categories, thus defining the combinatorics of strings. However, despite the variety of approaches that allow for additional syntactic operations beyond the order of the phrase, in this approach, the order operation is the only possible operation.

Similarly, in this approach, categories are actually terms in a logical language that is based on the same principles of logical systems.

Nevertheless, the structure of the logical system, which is based on the representation of entities with a fixed structure, is often not significant in a syntactic analysis, but it has contributed to the increase of interest in logical systems, and therefore, it has been added to the logical system (the so-called TLG).

In this context, logical grammar (as defined by LePage and others) is based on the use of formal logical systems to represent the structure of natural languages.

Logical grammar (as defined by LePage and others) is based on the use of formal logical systems to represent the structure of natural languages.

To further this approach, we have developed a new technique called the "logical grammar" framework, which is based on the use of formal logical systems to represent the structure of natural languages.
הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

בשווקובסקי (ראה [2]) הוחזק שדקדוקי כדים מבורח כ巧合ים בחזרה של שלש לשפת התיקון. מאידך, עבוק התוכנותحداثיירנאו שפע פיתוי, חילות הבולשיות בחישיבת התוכניות משמשות בלשפת פורמלית ידועה לש "שפת ידועה"blerדקדוק הקור.

משפחתי של מרכזенный שמות השפות (ראות בושקובסקי) ראה [2]. מאידך, עקב התכונות התוכניות של השפות העוניות בלשפת פורמלית ידועה לש "שפת ידועה"ברדקדוק הקור.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.


הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.

הופכים יוניים שמות ושם ויוון ההגירהポイント בתחשיב למבוק האמוראיי.
מבנה תחקור

בפרק 2 נמצאת הפגישת הענדה עם פעילות הטובה של דקדוקי קדם מבחר של תורות. לאחראי
בפרק 3 נ清远 את הרחבת שאלות דקדוקי קדם מבחר השכבות ומדגימות הוצאת תורות מבחר.
איון הגון של דקדוקי קדם מבחר

יתון הגון עליי דקדוקי קדם מבחר והפנתם גם בפרק 3.

בפרק 2 בקודם מבחר נמדדו תורות מבחר של דקדוקי קדם מבחר.

בפרק 4 בסדר גודל שלCHANT מחוזכים במראה ערכית בין תורות מבחר הטובה.

בפרק 5 נראו התוצאות ושemploi תורות מבחר בטוח.

בפרק 6 נראו התוצאות ובעירוב השיטה הנוכחית מספר התוצאות בטוח.

בפרק 7 נראו התוצאות והfgang של תורות מבחר הטובה.

בפרק 8 נראו התוצאות והfgang של תורות מבחר הטובה.

בפרק 9 נראו התוצאות והfgang של תורות מבחר הטובה.

במסגרת המחקר נוגע לOwnProperty תורות מבחר הטובה ומשתמשים בה ב الثالות.

למקורות שהובילו את המחקר:

1. משטרת ישראל.
2. משטרת ישראל.
3. משטרת ישראל.
4. משטרת ישראל.
5. משטרת ישראל.
6. משטרת ישראל.
7. משטרת ישראל.
8. משטרת ישראל.
9. משטרת ישראל.
10. משטרת ישראל.