Ambiguity of context–free languages as a function of the word length

Mohamed NAJI

Email: MohamedNaji@web.de

WWW: http://www.mohamednaji.de.vu

Mobile +49-160-1127558

November 1998

Abstract

In this paper we discus the concept of ambiguity of context–free languages and grammars. We prove the existence of constant ambiguous, exponential ambiguous and polynomial ambiguous languages and we give examples for these classes of ambiguity

Contents

List of Figures

1 Introduction

The concept of ambiguity plays a fundamental role in formal language theory. Measuring the amount of ambiguity in context–free grammars is well known; see for example [1, Section 7.3]. We define the ambiguity as a function of the word length

2 Preliminaries

We use the following notations and definitions of grammars and languages as introduced in [5]:

2.1 context–free grammar

A context–free grammar (CFG) is a quadruple G=(N, Σ, P, S) where N and Σ are finite disjoint sets of nonterminals and terminals respectively; P is a finite set of productions of the form $A \to \alpha$ where $A \in N$ and $\alpha \in (N \cup \Sigma)^*$; $S \in N$ is the start symbol. If $A \to \alpha$ is in P and α_1, α_2 are in $(N \cup \Sigma)^*$, then we write $\alpha_1 A \alpha_2 \Longrightarrow \alpha_1 \alpha \alpha_2$. $\overset{i}{\Longrightarrow}$ is the i–fold product, $\overset{+}{\Longrightarrow}$ is the transitive, $\overset{*}{\Longrightarrow}$ the reflexive and transitive closure of $\Longrightarrow$. The context–free language (CFL) generated by G is L(G):= $\{w \in \Sigma^* | S \overset{*}{\Longrightarrow} w\}$.

A language L is termed context–free if L=L(G) for a CFG G. $\#_a(w)$ denotes the number of a's in w, $|w|$ the length of w.

2.2 O–Notations

Let $f, g : \mathbb{N} \to \mathbb{R}_+$ be functions

$$\begin{aligned}
g = O(f) &\;:\Leftrightarrow\; (\exists c \in \mathbb{R}_+, \exists n_o \in \mathbb{N}) : (\forall n \geq n_0) : (g(n) \leq cf(n)) \\
g = \Omega(f) &\;:\Leftrightarrow\; (\exists c \in \mathbb{R}_+, \exists n_o \in \mathbb{N}) : (\forall n \geq n_0) : (g(n) \geq cf(n)) \\
g = \Theta(f) &\;:\Leftrightarrow\; g = O(f)\ g = \Omega(f) \\
g = 2^{O(n)} &\;:\Leftrightarrow\; (\exists c \in \mathbb{R}_+, \exists n_o \in \mathbb{N}) : (\forall n \geq n_0) : (g(n) \leq 2^{cn}) \\
g = 2^{\Omega(n)} &\;:\Leftrightarrow\; (\exists c \in \mathbb{R}_+, \exists n_o \in \mathbb{N}) : (\forall n \geq n_0) : (g(n) \geq 2^{cn}) \\
g = 2^{\Theta(n)} &\;:\Leftrightarrow\; g = 2^{O(n)}\ \textit{and}\ g = 2^{\Omega(n)}
\end{aligned}$$

2.3 Ogden's Lemma

[5] Let G=(N, Σ, P, S) be a CFG. Then there is a constant h=h(G), such that for every word z ∈ L(G) with at least h marked positions, there is a factorization z=uvwxy with:

1. w contains at least one of the marked positions
2. *Either* u and v both contain marked positions, *or* x and y both contain marked positions
3. vwx has at most h marked positions
4. ∃A∈N such that $S \overset{+}{\Longrightarrow}$ uAy$\overset{+}{\Longrightarrow}$ uvAxy$\overset{+}{\Longrightarrow} \ldots \overset{+}{\Longrightarrow} uv^qAx^qy \overset{+}{\Longrightarrow} uv^qwx^qy$ ∈L(G) for all integers q≥ 0

Remark 2.1 *Point (4) of* Ogden*'s Lemma (on page 4) says, that each derivation tree of z=uvwxy in G has a subtree rooted at A which could be* *pumped to obtain a derivation tree of uv^qwx^qy in G for $q > 0$. We call such a subtree a A–pumptree. (see Figure 1 on page 5)*

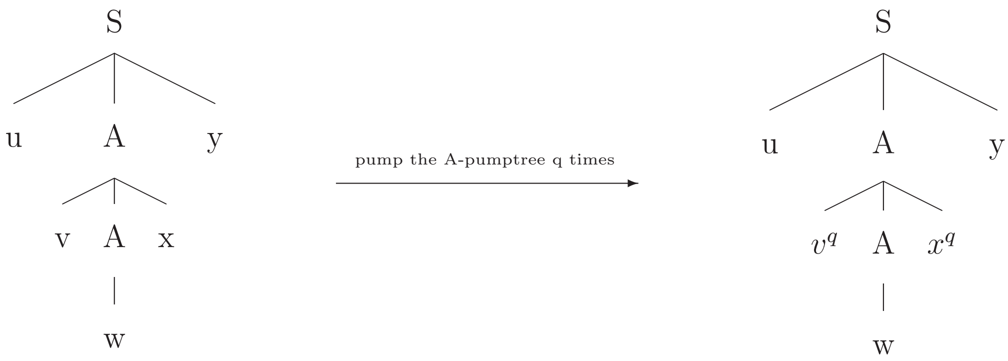

Figure 1: derivation trees and A–pumptrees

3 Ambiguity

Measuring the amount of ambiguity in context–free grammars is well known, see for example, [1, Section 7.3]. We define the ambiguity as a function of the word length n.

Definition 3.1 (Ambiguity of CFG) *Let $k > 0$ be an arbitrary integer, $f : \mathbb{N} \to \mathbb{R}_+$ be a non constant function and $\otimes \in \{O, \Omega, \Theta\}$.*

- *The ambiguity $da_G(w)$ of a word w in a CFG G is $da_G(w)$:=number of derivation trees (leftmost derivations)[1] of w in G.*
- *The ambiguity $da_G(n)$ of a CFG G is $da_G(n)$:=sup$\{da_G(w) | w \in \Sigma^*$ and $|w| \leq n\}$.*

[1] For the definition of derivation and leftmost derivation see [5]

- *G is at least k–ambiguous :⇔ There is a word in L(G) for which there is at least k distinct derivation trees in G.*
- *G is at most k–ambiguous :⇔ There is a word with at most k derivation trees in G.*
- *G is k–ambiguous :⇔ (G is at least k–ambiguous) and (G is at most k–ambiguous).*
- *G is polynomial of degree k ambiguous* $:\Leftrightarrow da_G(n) = \Theta(n^k)$.
- *G is exponential ambiguous* $:\Leftrightarrow da_G(n) = 2^{\Theta(n)}$.
- *G is* $\otimes(f(n))$*–ambiguous* $:\Leftrightarrow da_G(n) = \otimes(f(n))$.
- *G is* $2^{\otimes(f(n))}$*–ambiguous* $:\Leftrightarrow da_G(n) = 2^{\otimes(f(n))}$.

Definition 3.2 (Ambiguity of CFL) *Let* $k > 0$ *be an arbitrary integer and* $f : \mathbb{N} \to \mathbb{R}_+$ *be a non constant function.*

- *A CFL L is k–ambiguous :⇔ each CFG for L is at least k–ambiguous and there is an at most k–ambiguous CFG for L.*
- *A CFL L is polynomial of degree k ambiguous :⇔ each CFG for L is* $\Omega(n^k)$*–ambiguous and there is a* $O(n^k)$*–ambiguous CFG for L.*
- *A CFL L is exponential ambiguous :⇔ each CFG for L is* $2^{\Omega(n)}$ *–ambiguous and there is a* $2^{O(n)}$*–ambiguous CFG for L.*
- *A CFL L is* $\Theta(f(n))$*–ambiguous :⇔ each CFG for L is* $\Omega(f(n))$*–ambiguous and there is a* $O(f(n))$*–ambiguous CFG for L.*

Theorem 3.1 *For all cycle–free*[2] *CFG G,* $da_G(n) \leq 2^{cn}$ *for some* $c > 0$*.*

Proof Let G=(N, Σ, P, S) be a cycle–free CFG.

The number of derivation trees, which can be obtained in i leftmost derivations steps, is at most $|P|^i$.

For every cycle–free grammar there are integers a, b such that $(A \overset{i}{\Longrightarrow} w)$ implies $(i \leq a|w| + b)$ [2, Theorem 4.1].

Thus the number of derivation trees of a word w in a cycle–free CFG G is at most $|P|^{a|w|+b} = 2^{(an+b)\log|P|}$, where $n := |w|$ and log denotes the binary logarithm.■

Remark 3.1

- *By Theorem 3.1 there isn't any CFL which has an ambiguity bigger than* $2^{\Theta(n)}$ *(e. g.*$\Theta(n^n)$*).*
- Wich *[6] has proven, that there isn't any grammar (and so there isn't any language) with ambiguity bigger than polynomial but smaller than proper exponential (e. g.* $\Theta(2^{\sqrt{n}})$*)*

4 Constant ambiguous languages

Maurer [3] has proven the existence of context–free languages which are inherently ambiguous of any degree. We reprove this result using Ogden's Lemma (on page 4) and another (less complicated) language

Theorem 4.1 *Let k be a constant from* $\mathbb{N}$*.*

$L_k := \{a^m b_1^{m_1} b_2^{m_2} \dots b_k^{m_k} | m, m_1, m_2, \dots, m_k \geq 1, \exists\ i \text{ with } m = m_i\}$ *is k–ambiguous.*

[2] A CFG is cycle–free if there is no derivation of the form $A \overset{+}{\Longrightarrow} A$ for any nonterminal A.

Proof For k=1 we obtain the well known unambiguous language $L_1 := \{a^m b_1^m | m \geq 1\}$.

Let $k \geq 2$, $L_k = L(G)$ for some CFG G=(N, Σ, P, S) and h be the constant for G from OGDEN's Lemma (on page 4). Now we consider the words

$$z_i := a^h b_1^{h_1} b_2^{h_2} \dots b_k^{h_k} \ with \quad h_j := \begin{cases} h & , \ if \ j = i \\ h + h! & , \ otherweise \end{cases} , for \quad i = 1, \dots, k$$

where all the a's are marked. It's not difficult to prove, that for every factorization $z_i = u_i v_i w_i x_i y_i$ satisfying conditions (1)-(4) of OGDEN's Lemma (on page 4)

$$\begin{aligned} u_i &= a^{r_i} & 1 \leq r_i \leq h-2, \\ v_i &= a^{s_i} & 1 \leq s_i \leq h-2, \\ w_i &= a^{h-s_i-r_i} b_1^{h+h!} \dots b_{i-1}^{h+h!} b_i^{t_i} & 0 \leq t_i \leq h-1, \\ x_i &= b_i^{s_i} \\ y_i &= b_i^{h-s_i-t_i} b_{i+1}^{h+h!} \dots b_k^{h+h!}. \end{aligned}$$

Since
$S \overset{+}{\Longrightarrow} u_i A_i y_i \overset{+}{\Longrightarrow} u_i v_i A_i x_i y_i \overset{+}{\Longrightarrow} u_i v_i w_i x_i y_i = z_i$,
every derivation tree B_i of z_i in G has an A_i–pumptree (see Figure 2 on page 9)

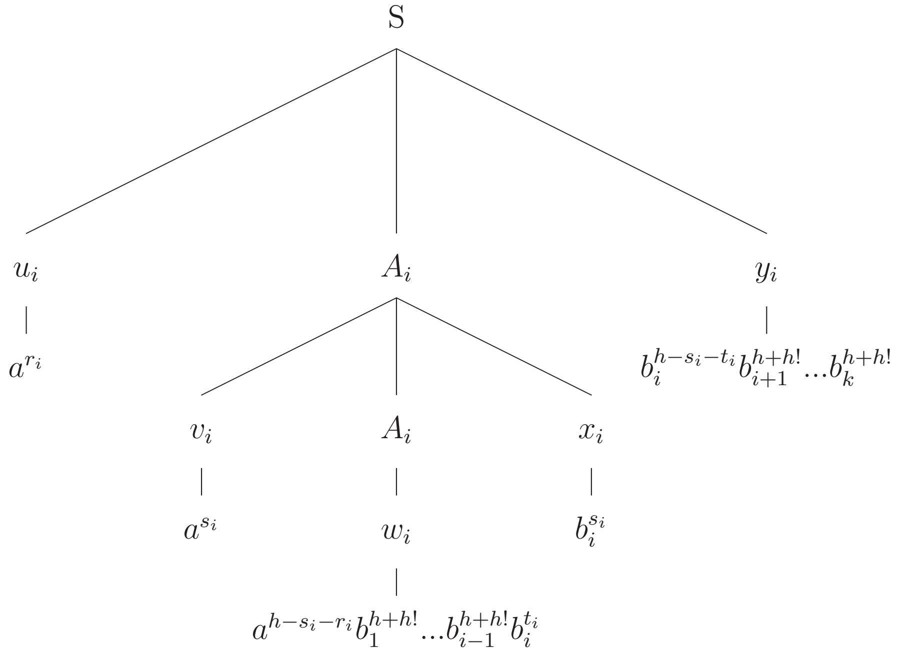

Figure 2: derivation tree B_i with A_i–pumptree for $z_i := a^h b_1^{h+h!} \dots b_{i-1}^{h+h!} b_i^h b_{i+1}^{h+h!} \dots b_k^{h+h!}$

We pump the A_i–pumptree (of the derivation tree B_i) $q_i := \frac{h!}{s_i} + 1$ times, we obtain a derivation tree T_i for the word $z := a^{h+h!} b_1^{h+h!} b_2^{h+h!} \dots b_k^{h+h!}$ in G.

Since i=1, …,k, we obtain k derivation trees T_1, T_2, …, T_k for the word $z := a^{h+h!} b_1^{h+h!} b_2^{h+h!} \dots b_k^{h+h!}$ in G.

We now prove that these k derivation trees are distinct.

Suppose there are $i, j \in \{1, \dots, k\}$ with $i \neq j$ but $T_i = T_j = T$.

The derivation tree T must have both nodes A_i (because $T = T_i$) and nodes A_j (because $T = T_j$).

Case 1: Neither A_i nor A_j appears (in the tree T) as a descendant of the other.

w. l. o. g. A_i appears on the left of A_j (see Figure 3 on page 10)

The frontier of T is a word in which b's would precede a's and hence is not in L_k, a contradiction (see Figure 3 on page 10)

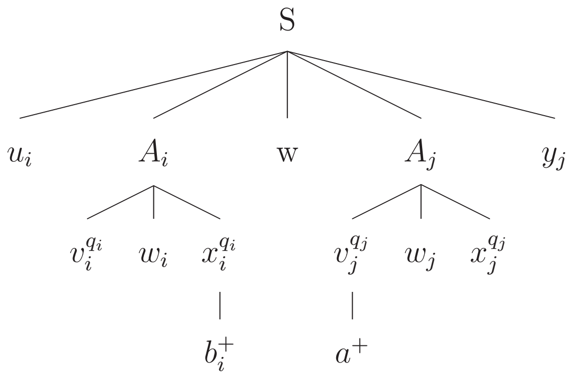

Figure 3: A_i on the left of A_j in the tree T

Case 2: Either A_i or A_j appears (in the tree T) as a descendant of the other

w. l. o. g. A_i is a descendant of A_j. (see Figure 4 on page 11)

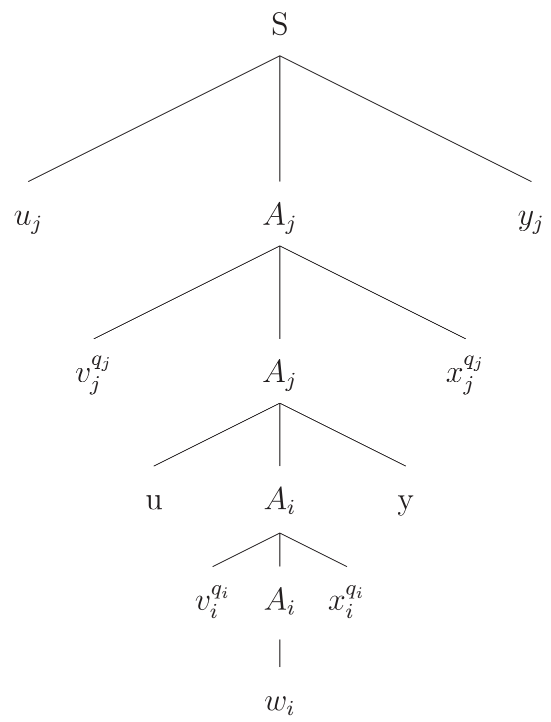

Figure 4: A_i is a descendant of A_j in the tree T for z=$a^{h+h!}b_1^{h+h!}b_2^{h+h!}\dots b_k^{h+h!}$

We obtain:

$$\begin{aligned} S &\stackrel{+}{\Longrightarrow} u_jA_jy_j \\ &\stackrel{+}{\Longrightarrow} u_jv_j^{q_j}A_jx_j^{q_j}y_j \\ &\stackrel{+}{\Longrightarrow} u_jv_j^{q_j}uA_iyx_j^{q_j}y_j \\ &\stackrel{+}{\Longrightarrow} u_jv_j^{q_j}uv_i^{q_i}w_ix_i^{q_i}yx_j^{q_j}y_j \\ &= z \in L_k \end{aligned}$$

where $\#_a(z) = \#_{b_r}(z) = h + h! \ \ \forall r \in \{1, \dots, i, \dots, j, \dots, k\}$

But if we pump the A_i–pumptree of the A_j–pumptree (in the tree T), then we obtain:

$$\begin{aligned} S &\stackrel{+}{\Longrightarrow} u_j A_j y_j \\ &\stackrel{+}{\Longrightarrow} u_j v_j^{q_j+1} A_j x_j^{q_j+1} y_j \\ &\stackrel{+}{\Longrightarrow} u_j v_j^{q_j+1} u A_i y x_j^{q_j+1} y_j \\ &\stackrel{+}{\Longrightarrow} u_j v_j^{q_j+1} u v_i^{q_i+1} w_i x_i^{q_i+1} y x_j^{q_j+1} y_j \\ &:= \tilde{z} \in L_k \end{aligned}$$

where:

$$\begin{aligned} \#_a(\tilde{z}) &= \#_a(z) + |v_j| + |v_i| = h + h! + |v_j| + |v_i| \\ \#_{b_i}(\tilde{z}) &= \#_{b_i}(z) + |x_i| = h + h! + |v_i| \\ \#_{b_j}(\tilde{z}) &= \#_{b_j}(z) + |x_j| = h + h! + |v_j| \\ \#_{b_r}(\tilde{z}) &= \#_{b_r}(z) = h + h! \end{aligned}$$

Thus
$\forall r \in \{1, \ldots, k\}, \#_a(\tilde{z}) \neq \#_{b_r}(\tilde{z})$, a contradiction of $u_j v_j^{q_j+1} u v_i^{q_i+1} w_i x_i^{q_i+1} y x_j^{q_j+1} y_j := \tilde{z} \in L_k$.

Each CFG for L_k is therefore at least k–ambiguous.■

It is not difficult to give an at most k–ambiguous CFG for L_k. An at most k–ambiguous CFG for L_k can be found in [4].

5 Exponential ambiguous languages

Theorem 5.1 *Let* $L = \{a^i b^i c^j | i, j \geq 1\} \cup \{a^i b^j c^i | i, j \geq 1\}$. L^* *is exponential ambiguous.*

Proof Let L^*=L(G) for a CFG G=(N, Σ, P, S) and h be the constant from Ogden's Lemma (on page 4) for G. We consider the words of L^* of the form

$z = z_1 z_2 \ldots z_k$, where $z_i \in \{a^h b^h c^{h+h!}, a^h b^{h+h!} c^h\}$ $\forall i \in \{1, \ldots, k\}$ and mark all the a's. Since the number of the marked positions in each z_i is equal to h, for each given i we can find a factorization $z = \hat{u}_i v_i w_i x_i \hat{y}_i$ and we can construct a path π_i in each derivation tree B(z) for z in G (with the same idea as the well known proof of OGDEN's Lemma [5, Theorem 2.24]) such that:

1. w_i contains at least one of the marked positions of z_i
2. *Either* $\hat{u}_i$ and v_i both contain marked positions of z_i, *or* x_i and $\hat{y}_i$ both contain marked positions of z_i.
3. $v_i w_i x_i$ has at most h marked positions of z_i.
4. $$\begin{aligned} S &\overset{+}{\Longrightarrow} \hat{u}_i A_i \hat{y}_i \\ &\overset{+}{\Longrightarrow} \hat{u}_i v_i A_i x_i \hat{y}_i \\ &\overset{+}{\Longrightarrow} \ldots \\ &\overset{+}{\Longrightarrow} \hat{u}_i v_i^q A_i x_i^q \hat{y}_i \\ &\overset{+}{\Longrightarrow} \hat{u}_i v_i^q w_i x_i^q \hat{y}_i \in L^* \ for\ all\ integers\ q \geq 0 \end{aligned}$$

The situation is depicted in Figure (see Figure 5 on page 13)

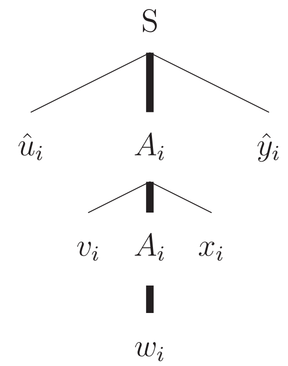

Figure 5: Illustration of the path π_i and the factorization z= $\hat{u}_i v_i^q w_i x_i^q \hat{y}_i$

We can further prove:

$$
\begin{aligned}
z_i = a^h b^h c^{h+h!}: \quad & \hat{u}_i = z_1 \ldots z_{i-1} u_i && u_i = a^{r_i} \textit{ and } 1 \le r_i \le h-2, \\
& v_i = a^{s_i} && 1 \le s_i \le h-2, \\
& w_i = a^{h-r_i-s_i} b^{h-s_i-t_i} && 0 \le t_i \le h-1, \\
& x_i = b^{s_i} && \\
& \hat{y}_i = y_i z_{i+1} \ldots z_k && y_i = b^{t_i} c^{h+h!}.
\end{aligned}
$$

$$
\begin{aligned}
z_i = a^h b^{h+h!} c^h: \quad & \hat{u}_i = z_1 \ldots z_{i-1} u_i && u_i = a^{r_i} \textit{ and } 1 \le r_i \le h-2, \\
& v_i = a^{s_i} && 1 \le s_i \le h-2, \\
& w_i = a^{h-r_i-s_i} b^{h+h!} c^{t_i} && 0 \le t_i \le h-1, \\
& x_i = c^{s_i} && \\
& \hat{y}_i = y_i z_{i+1} \ldots z_k && y_i = c^{h-t_i-s_i}.
\end{aligned}
$$

The proof is straightforward and will be omitted here, you can see [4]

Since $A_i \stackrel{+}{\Longrightarrow} v_i A_i x_i$, the derivation tree B(z) has an A_i–pumptree, whose frontier $v_i w_i x_i$ is a subword of z_i. We can use this argumentation for each $i \in \{1, \ldots, k\}$, thus the derivation tree B(z) consists of the k A_1–, A_2–, ..., A_k–pumptrees, which are in B(z) parallel to themselves. (see Figure 6 on page 15)

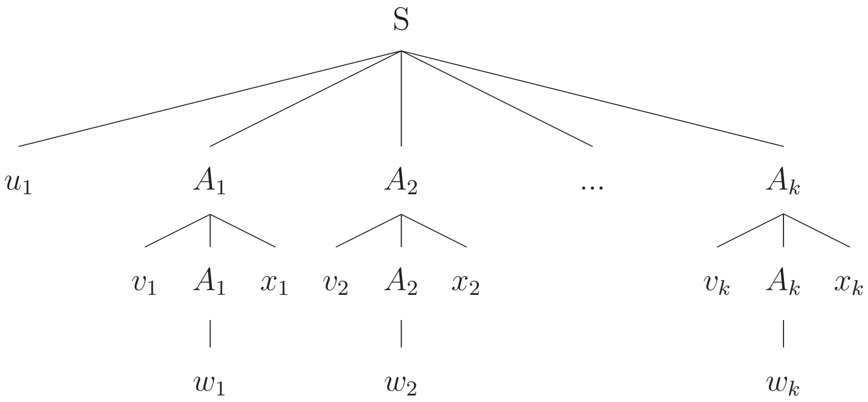

Figure 6: a derivation tree B(z) for a word z from $\{a^h b^h c^{h+h!}, a^h b^{h+h!} c^h\}^k$

If we pump each A_i–pumptree in the tree B(z) $q_i := \frac{h!}{s_i} + 1$ times, we will obtain a derivation tree T(z) for the word $(a^{h+h!} b^{h+h!} c^{h+h!})^k$ (see Figure 7 on page 15)

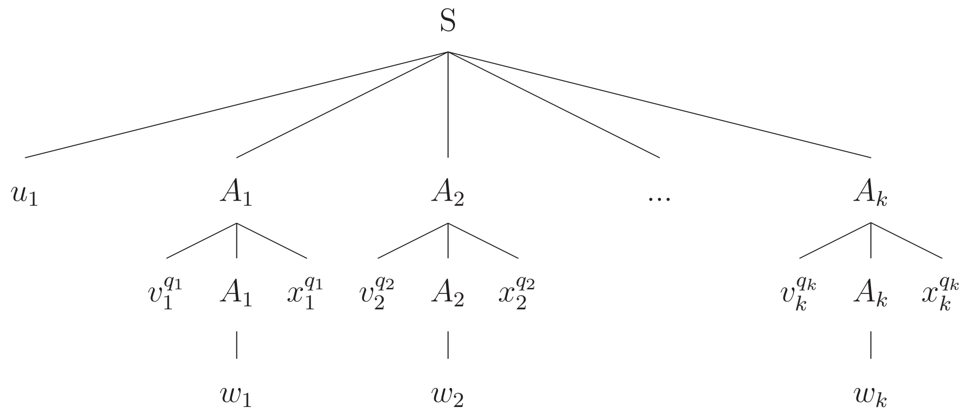

Figure 7: derivation tree T(z) for the word$(a^{h+h!} b^{h+h!} c^{h+h!})^k$

Since there are 2^k words of the form $z = z_1 z_2 \dots z_k$ where $z_i \in \{a^h b^h c^{h+h!}, a^h b^{h+h!} c^h\}$ $\forall i \in \{1, 2, \dots, k\}$, there are 2^k derivation trees of the form T(z) for the word $(a^{h+h!} b^{h+h!} c^{h+h!})^k$.

We now prove that these 2^k derivation trees are distinct. Suppose there are $z = z_1 z_2 \ldots z_k$ and $\tilde{z} = \tilde{z}_1 \tilde{z}_2 \ldots \tilde{z}_k$ where $z_i, \tilde{z}_i \in \{a^h b^h c^{h+h!}, a^h b^{h+h!} c^h\}$ with $z \neq \tilde{z}$ but $T(z) = T(\tilde{z}) = T(z, \tilde{z})$.

$z \neq \tilde{z}$ implies there is $i \in \{1, \ldots, k\}$ with $z_i \neq \tilde{z}_i$. W. l. o. g. let $z_i = a^h b^h c^{h+h!}$ and $\tilde{z}_i = a^h b^{h+h!} c^h$.

The tree $T(z, \tilde{z})$ must have both an A_i–pumptree (because $T(z, \tilde{z})$=T(z)) and an $\tilde{A}_i$–pumptree (because $T(z, \tilde{z})$=$T(\tilde{z})$). We discuss the two following cases.

Case 1: Neither the A_i–pumptree nor the $\tilde{A}_i$–pumptree is a subtree of the other.

w. l. o. g. the A_i–pumptree is on the left of the $\tilde{A}_i$–pumptree in the tree $T(z, \tilde{z})$ (see Figure 8 on page 16)

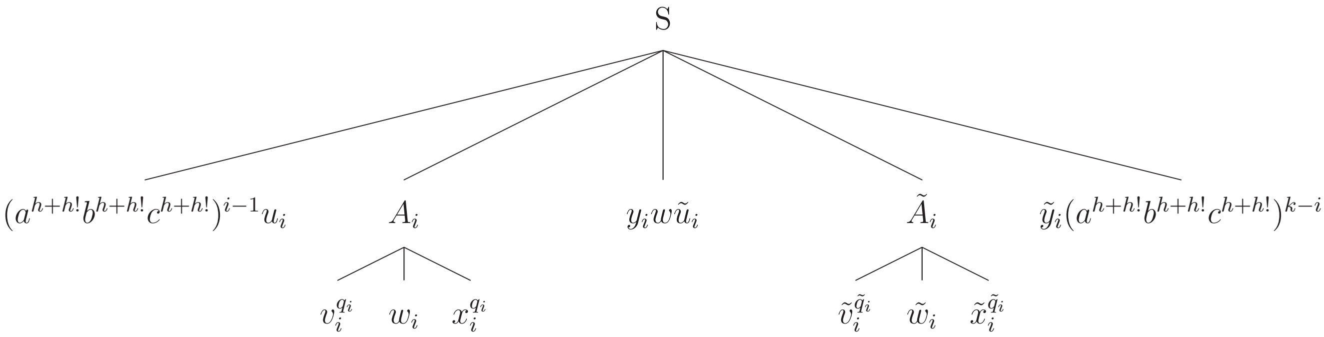

Figure 8: the A_i–pumptree is on the left of the $\tilde{A}_i$–pumptree in $T(z, \tilde{z})$

The frontier of the tree $T(z, \tilde{z})$ would have at least (k+1) subwords of the form $a^{h+h!}b^{h+h!}c^{h+h!}$. But the frontier of $T(z, \tilde{z})$ is the word $(a^{h+h!}b^{h+h!}c^{h+h!})^k$, a contradiction.

Case 2: Either the $\tilde{A}_i$–pumptree or the A_i–pumptree is a subtree of the other

w. l. o. g. A_i is a descendant of $\tilde{A}_i$ (see Figure 9 on page 17)

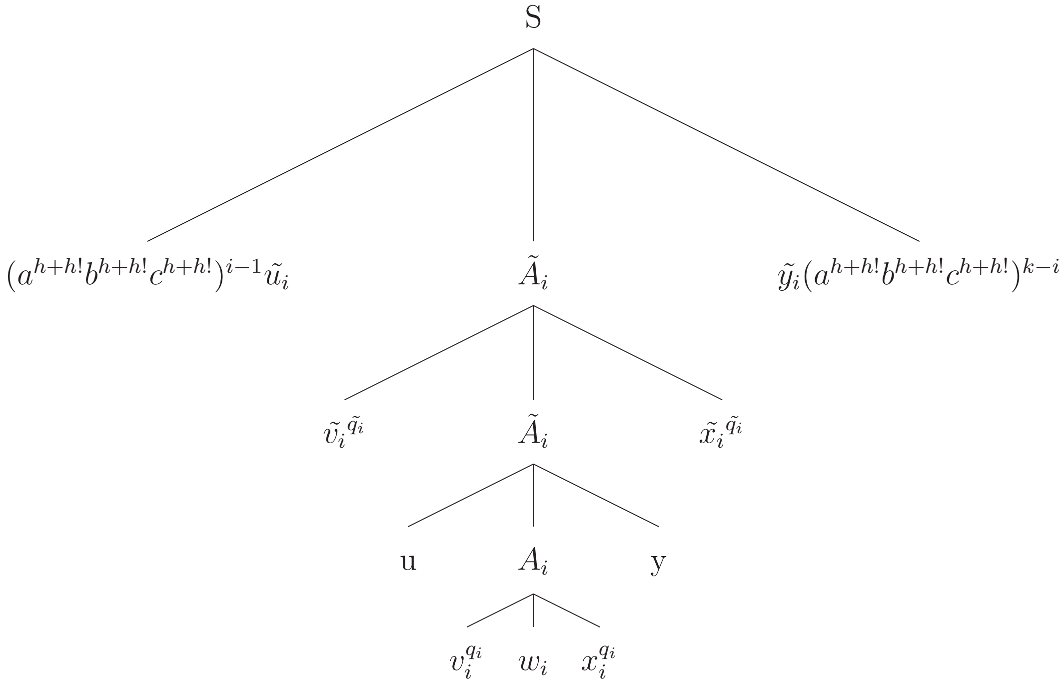

Figure 9: A_i is a descendant of $\tilde{A}_i$

We obtain here:

$$\begin{aligned}
S &\stackrel{+}{\Longrightarrow} (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}\tilde{u}_i\tilde{v}_i^{\tilde{q}_i}\tilde{A}_i\tilde{x}_i^{\tilde{q}_i}\tilde{y}_i(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}\tilde{u}_i\tilde{v}_i^{\tilde{q}_i}uA_iy\tilde{x}_i^{\tilde{q}_i}\tilde{y}_i(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}\tilde{u}_i\tilde{v}_i^{\tilde{q}_i}uv_i^{q_i}A_ix_i^{q_i}y\tilde{x}_i^{\tilde{q}_i}\tilde{y}_i(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}\underbrace{\tilde{u}_i\tilde{v}_i^{\tilde{q}_i}uv_i^{q_i}w_ix_i^{q_i}y\tilde{x}_i^{\tilde{q}_i}\tilde{y}_i}_{t_1}(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\\
&= (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}t_1(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i} \in L^*.
\end{aligned}$$

Since the frontier of $T(z, \tilde{z})$ is the word $(a^{h+h!}b^{h+h!}c^{h+h!})^k$, $t_1 = a^{h+h!}b^{h+h!}c^{h+h!}$.

However if we pump the A_i–pumptree and the $\tilde{A}_i$–pumptree in the tree $T(z,\tilde{z})$, then we obtain:

$$\begin{aligned}
S &\stackrel{+}{\Longrightarrow} (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}\tilde{u}_i\tilde{v}_i^{\tilde{q}_i+1}\tilde{A}_i\tilde{x}_i^{\tilde{q}_i+1}\tilde{y}_i(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}\tilde{u}_i\tilde{v}_i^{\tilde{q}_i+1}uA_iy\tilde{x}_i^{\tilde{q}_i+1}\tilde{y}_i(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}\tilde{u}_i\tilde{v}_i^{\tilde{q}_i+1}uv_i^{q_i+1}A_ix_i^{q_i+1}y\tilde{x}_i^{\tilde{q}_i+1}\tilde{y}_i(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}\underbrace{\tilde{u}_i\tilde{v}_i^{\tilde{q}_i+1}uv_i^{q_i+1}w_ix_i^{q_i+1}y\tilde{x}_i^{\tilde{q}_i+1}\tilde{y}_i}_{t_2}(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\\
&= (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}t_2(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i} \in L^*.
\end{aligned}$$

$$\begin{aligned}
\#_a(t_2) &= \#_a(t_1)+|\tilde{v}_i|+|v_i| = h+h!+|\tilde{v}_i|+|v_i|\\
\#_b(t_2) &= \#_a(t_1)+|x_i| = h+h!+|v_i|\\
\#_c(t_2) &= \#_a(t_1)+|\tilde{x}_i| = h+h!+|\tilde{v}_i|
\end{aligned}$$

Thus $\#_a(t_2)\neq\#_b(t_2)$ and $\#_a(t_2)\neq\#_c(t_2)$ and therefore $t_2\notin L$. A contradiction of $= (a^{h+h!}b^{h+h!}c^{h+h!})^{i-1}t_2(a^{h+h!}b^{h+h!}c^{h+h!})^{k-i}\in L^*$. We can now conclude, that the 2^k derivation trees are distinct, and each CFG for L^* is therefore $2^{\Omega(n)}$–ambiguous. By Theorem 3.1 (on page 7) and Remark 3.1 (on page 7) there isn't any language, which has an ambiguity bigger than $2^{\Theta(n)}$. Thus L^* is exponential ambiguous.■

6 Polynomial ambiguous languages

Theorem 6.1 *Let* $L := \{a^m b^{m_1} c b^{m_2} c \ldots b^{m_p} c | p\in\mathbb{N};\ m, m_1, m_2, \ldots, m_p \in \mathbb{N}; \exists i\in\{1,2,\ldots,p\}$ *with* $m = m_i\}$. L^k *is polynomial of degree* k *ambiguous.*

Proof Let $L^k = L(G)$ for some CFG G=(N, Σ, P, S) and h be the constant for G from Ogden's Lemma (on page 4). Now we consider the words of L^k

of the form $z = z_{i_1} z_{i_2} \ldots z_{i_k}$ where $z_{i_j} := a^h (b^{h+h!}c)^{i_j - 1} b^h c (b^{h+h!}c)^{p-i_j}$, j=1, $\ldots$,k and $i_j = 1, \ldots, p$ and mark all the a's in each z_{i_α} with $\alpha \in \{1, 2, \ldots, k\}$. Similar to the proof of Theorem 6.1 we can prove, that each derivation tree B(z) for z in G consists of k A_{i_1}–, A_{i_2}–, A_{i_k}–pumptrees, which are parallel to themselves in the tree B(z). (see Figure 10 on page 19)

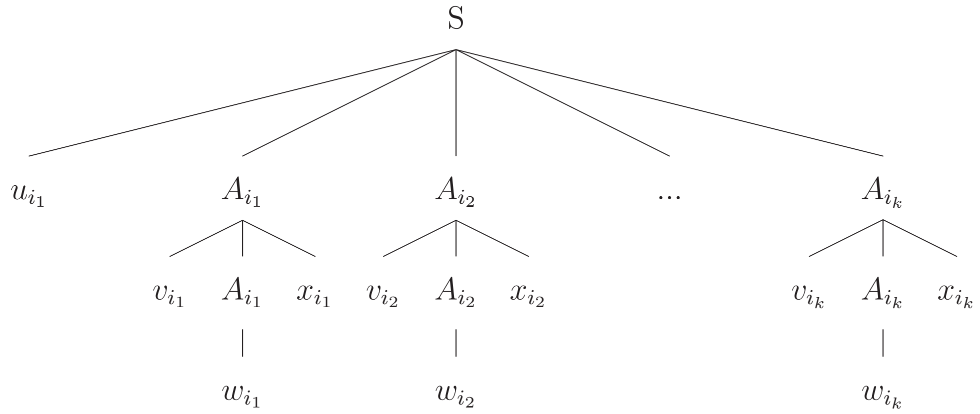

Figure 10: a derivation tree B(z) for a word $z = z_{i_1} z_{i_2} \ldots z_{i_k}$

We now pump each A_{i_j}–pumptree of the tree B(z) $q_{i_j} = \frac{h!}{s_{i_j}} + 1$ times, we obtain a derivation tree T(z) for the word $(a^{h+h!}(b^{h+h!}c)^p)^k$. (see Figure 11 on page 20)

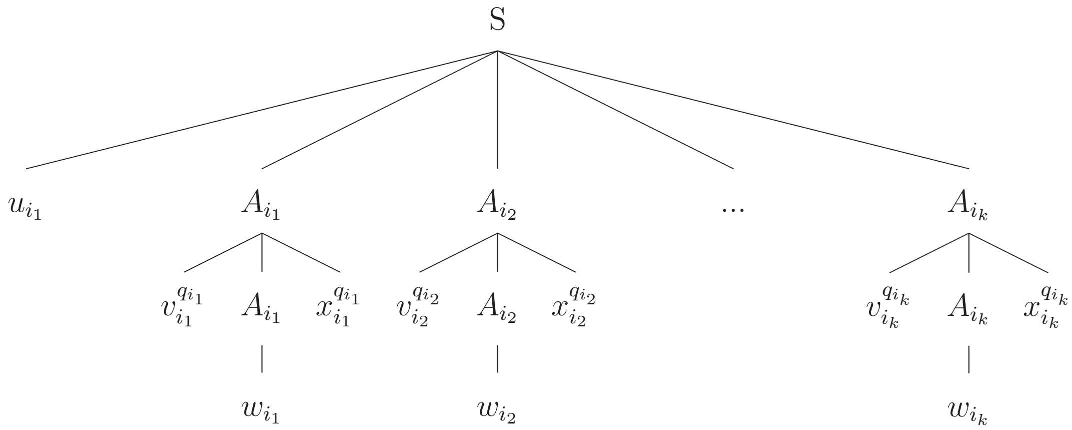

Figure 11: a derivation tree T(z) for the word $(a^{h+h!}(b^{h+h!}c)^p)^k$

Since there are p^k words of the form $z = z_{i_1}z_{i_2}\ldots z_{i_k}$ where $z_{i_j} := a^h(b^{h+h!}c)^{i_j-1}b^hc(b^{h+h!}c)^{p-i_j}$, j=1, ...,k and $i_j = 1,\ldots,p$, there are p^k derivation trees of the form T(z).

We now prove, that these p^k derivation trees of the form T(z) are distinct.

Suppose	*there*	*are*
$z = z_{i_1}z_{i_2}\ldots z_{i_k}$	*where*	$z_{i_j} := a^h(b^{h+h!}c)^{i_j-1}b^hc(b^{h+h!}c)^{p-i_j}$
and		
$\tilde{z} = z_{\tilde{i}_1}z_{\tilde{i}_2}\ldots z_{\tilde{i}_k}$	*where*	$z_{\tilde{i}_j} := a^h(b^{h+h!}c)^{\tilde{i}_j-1}b^hc(b^{h+h!}c)^{p-\tilde{i}_j}$

$z \neq \tilde{z}$ implies there is j such that $i_j \neq \tilde{i}_j$.

The tree $T(z,\tilde{z})$ must have both an A_{i_j}–pumptree (because $T(z,\tilde{z})$=T(z)) and an $A_{\tilde{i}_j}$–pumptree (because $T(z,\tilde{z})$=$T(\tilde{z})$. We discuss the two following cases.

Case 1: Neither the A_{i_j}–pumptree nor the $A_{\tilde{i}_j}$–pumptree is a subtree of the other

w. l. o. g. the A_{i_j}–pumptree is on the left of the $A_{\tilde{i}_j}$–pumptree in the tree $T(z, \tilde{z})$ (see Figure 12 on page 21)

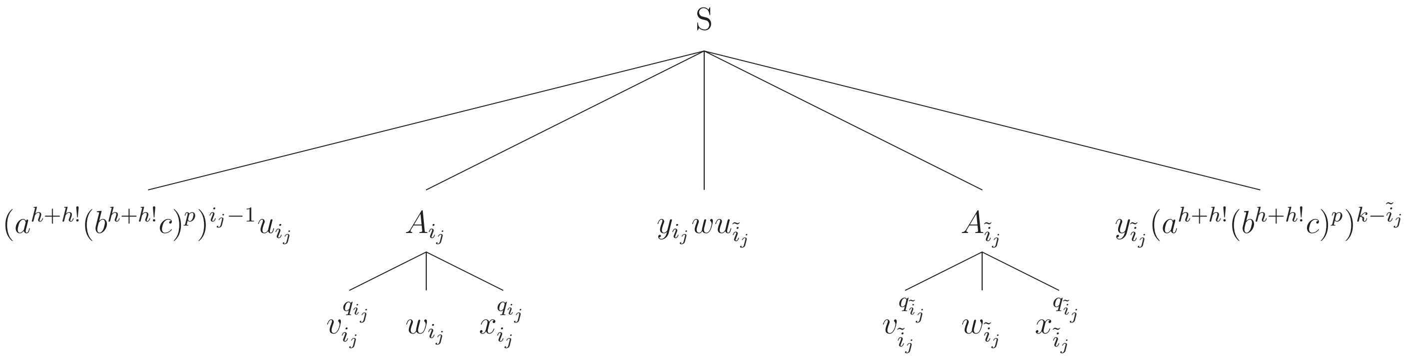

Figure 12: A_{i_j} on the left of $A_{\tilde{i}_j}$ in $T(z, \tilde{z})$

The frontier of the tree $T(z, \tilde{z})$ would have at least (k+1) subtrees of the form $a^{h+h!}(b^{h+h!}c)^p$. But the frontier of the tree $T(z, \tilde{z})$ is the word $(a^{h+h!}(b^{h+h!}c)^p)^k$, a contradiction.

Case 2: Either the A_{i_j}–pumptree or the $A_{\tilde{i}_j}$–pumptree is a subtree of the other

w. l. o. g. A_{i_j} is a descendant of $A_{\tilde{i}_j}$ (see Figure 13 on page 22)

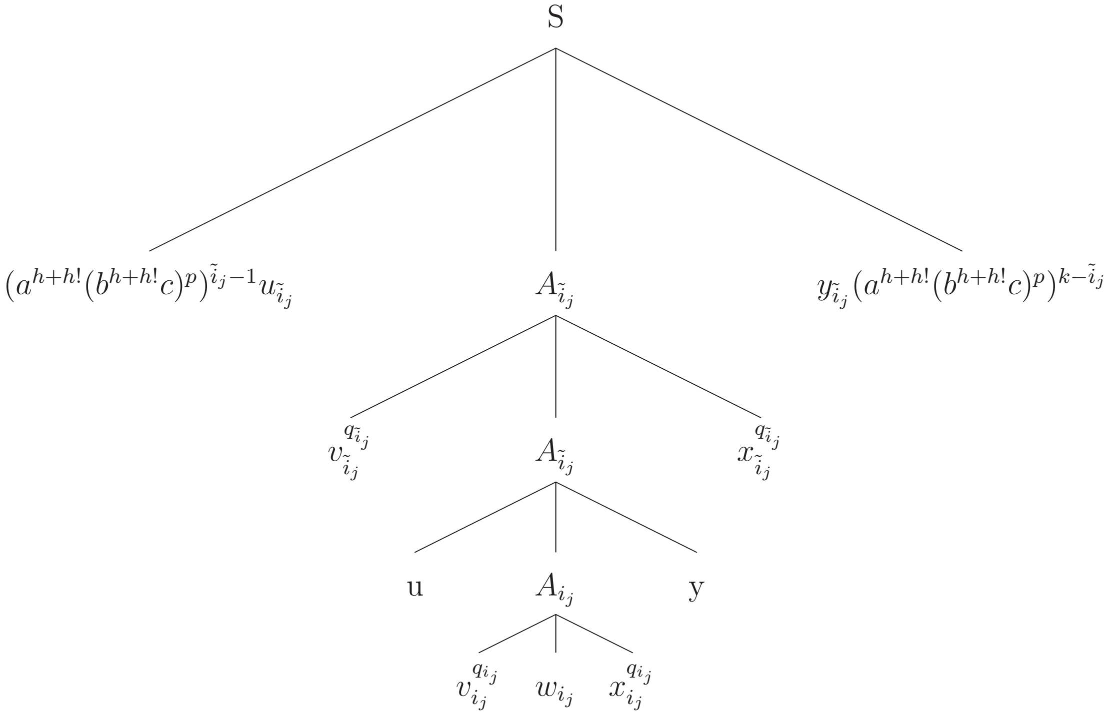

Figure 13: A_{i_j} is a descendant of $A_{\tilde{i}_j}$

We obtain here:

$$\begin{aligned}
S &\stackrel{+}{\Longrightarrow} (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}u_{\tilde{i}_j}v_{\tilde{i}_j}^{q_{\tilde{i}_j}}A_{\tilde{i}_j}x_{\tilde{i}_j}^{q_{\tilde{i}_j}}y_{\tilde{i}_j}(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}u_{\tilde{i}_j}v_{\tilde{i}_j}^{q_{\tilde{i}_j}}uA_{i_j}yx_{\tilde{i}_j}^{q_{\tilde{i}_j}}y_{\tilde{i}_j}(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}u_{\tilde{i}_j}v_{\tilde{i}_j}^{q_{\tilde{i}_j}}uv_{i_j}^{q_{i_j}}A_{i_j}x_{i_j}^{q_{i_j}}yx_{\tilde{i}_j}^{q_{\tilde{i}_j}}y_{\tilde{i}_j}(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \\
&\stackrel{+}{\Longrightarrow} (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}\underbrace{u_{\tilde{i}_j}v_{\tilde{i}_j}^{q_{\tilde{i}_j}}uv_{i_j}^{q_{i_j}}w_{i_j}x_{i_j}^{q_{i_j}}yx_{\tilde{i}_j}^{q_{\tilde{i}_j}}y_{\tilde{i}_j}}_{t_1}(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \\
&= (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}t_1(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \in L^k
\end{aligned}$$

Since the frontier of $T(z,\tilde{z})$ is the word $(a^{h+h!}(b^{h+h!}c)^p)^k$, $t_1 = a^{h+h!}(b^{h+h!}c)^p$.

if we pump however the A_i–pumptree and the $\tilde{A}_i$–pumptree in the tree $T(z,\tilde{z})$, then we obtain:

$$
\begin{aligned}
S &\overset{+}{\Longrightarrow} (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}u_{\tilde{i}_j}v_{\tilde{i}_j}^{q_{\tilde{i}_j}+1}A_{\tilde{i}_j}x_{\tilde{i}_j}^{q_{\tilde{i}_j}+1}y_{\tilde{i}_j}(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \\
&\overset{+}{\Longrightarrow} (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}u_{\tilde{i}_j}v_{\tilde{i}_j}^{q_{\tilde{i}_j}+1}uA_{i_j}yx_{\tilde{i}_j}^{q_{\tilde{i}_j}+1}y_{\tilde{i}_j}(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \\
&\overset{+}{\Longrightarrow} (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}u_{\tilde{i}_j}v_{\tilde{i}_j}^{q_{\tilde{i}_j}+1}uv_{i_j}^{q_{i_j}+1}A_{i_j}x_{i_j}^{q_{i_j}+1}yx_{\tilde{i}_j}^{q_{\tilde{i}_j}+1}y_{\tilde{i}_j}(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \\
&\overset{+}{\Longrightarrow} (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}\underbrace{u_{\tilde{i}_j}v_{\tilde{i}_j}^{q_{\tilde{i}_j}+1}uv_{i_j}^{q_{i_j}+1}w_{i_j}x_{i_j}^{q_{i_j}+1}yx_{\tilde{i}_j}^{q_{\tilde{i}_j}+1}y_{\tilde{i}_j}}_{t_2}(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \\
&= (a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}t_2(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \in L^k
\end{aligned}
$$

$$\#_a(t_2) = \#_a(t_1) + |v_{\tilde{i}_j}| + |v_{i_j}| = h + h! + |v_{\tilde{i}_j}| + |v_{i_j}|$$

The number of the b's in each b–Block of t_2 is either h+h! or $h+h!+|x_{\tilde{i}_j}|$ or $h+h!+|x_{i_j}|$ and therefore unequal to the numbere of the a's in t_2. Thus $t_2 \notin L$.

This is a contradiction to $(a^{h+h!}(b^{h+h!}c)^p)^{\tilde{i}_j-1}t_2(a^{h+h!}(b^{h+h!}c)^p)^{k-\tilde{i}_j} \in L^k$

.

We can conclude, that the word $a^{h+h!}(b^{h+h!}c)^p)^k$ has at least p^k derivation trees in G.

Since $n := |(a^{h+h!}(b^{h+h!}c)^p)^k| = k(p(h+h!+1)+h+h!)$, $da_G(n) = \Omega(n^k)$.■

The grammar with the productions:

$S \to E^k$

$E \to aTbcA|aTbc$

$T \to aTb|\varepsilon|A$

$A \to bA|bcA|bc$

produces L^k and is $O(n^k)$–ambiguous. [4]

7 Conclusion

From this work we obtain the following classes of CFL:

- constant ambiguous languages: e.g. $L_k := \{a^m b_1^{m_1} b_2^{m_2} \dots b_k^{m_k} | m, m_1, m_2, \dots, m_k \geq 1, \exists\ i\ with\ m = m_i\}$
- polynomial ambiguous languages: e.g. L^k where $L := \{a^m b^{m_1} c b^{m_2} c \dots b^{m_p} c | p \in \mathbb{N}; m, m_1, m_2, \dots, m_p \in \mathbb{N}; \exists i \in \{1, 2, \dots, p\}\ with\ m = m_i\}$
- "subbexponential" ambiguous languages (e.g. $\Theta(2^{\sqrt{n}})$–ambiguous languages): There isn't any language
- exponential ambiguous languages: e.g. L^* where $L = \{a^i b^i c^j | i, j \geq 1\} \cup \{a^i b^j c^i | i, j \geq 1\}$
- Languages, whose ambiguity bigger than exponential (e.g. $\Theta(n^n)$–ambiguous languages): There isn't any language

However there remain the following questions:

1. Is there any $\Theta(n^r)$–ambiguous languages, where r is a non natural number?
2. Is there any "sublinear" ambiguous languages (e. g. $\Theta(\log(n))$–ambiguous languages)?

References

[1] M. A. Harrison, *Introduction to Formal language Theory*, Addison–Wesley, 1978.

[2] S. Heilbrunner, *Parsing automata approach to LR theory*, Theoret. Comput. Sci. (1981), no. 15, 117–157.

[3] H. Maurer, *The existence of context–free languages which are inherently ambiguous of any degree*, Department of Mathematics Research Series; University of Calgary, (1968).

[4] M. Naji, *Der Grad der Mehrdeutigkeit kontextfreier Grammatiken und Sprachen*, Master's thesis, Fachbereich Informatik; Universität Frankfurt am Main, 1998.

[5] A. V. Aho & J. D. Ullman, *The Theory of Parsing, Translation and Compiling*, Prentice Hall, 1972.

[6] K. Wich, *Kriterien für die Mehrdeutigkeit kontextfreier Grammatiken*, Master's thesis, Fachbereich Informatik; Universität Frankfurt am Main, 1997.

Index