Exponential ambiguity of context-free grammars

K. Wich
Fachbereich Mathematik/Informatik, Universität Kassel, 34109 Kassel
E-mail: wich@theory.informatik.uni-kassel.de

Abstract

A context-free grammar G is ambiguous if and only if there is a word that can be generated by G with at least two different derivation trees. Ambiguous grammars are often distinguished by their degree of ambiguity, which is the maximal number of derivation trees for the words generated by them. If there is no such upper bound G is said to be ambiguous of infinite degree. Here as a new tool for examining the ambiguity of cycle-free context-free grammars the ambiguity function is introduced. This function maps the natural number n to the maximal number of derivation trees which a word of length at most n may have. This provides the possibility to distinguish infinitely ambiguous context-free grammars by the growth-rate of their ambiguity functions. We present a necessary and sufficient, but in general undecidable, criterion for exponential ambiguity. In fact violation of this criterion leads to a polynomial upper bound for the ambiguity, which can be effectively constructed from the grammar. Hence for cycle-free context-free grammars the ambiguity function is either an element of $2^{\Theta(n)}$ or of $\mathcal{O}(n^d)$ for some $d \in \mathbb{N}_0$ which can be effectively constructed from G.

1 Introduction

A context-free grammar G is ambiguous if and only if there is a word that can be generated by G with at least two different derivation trees. A context-free language is said to be (inherently) ambiguous if and only if there is no unambiguous context-free grammar for it. In [8] Parikh presents an inherently ambiguous context-free language which is the union of two unambiguous linear languages. Thus ambiguity contributes in an essential way to the generative power of context-free grammars. The problem whether a grammar or a language is ambiguous or not is undecidable.

Ambiguous grammars are often distinguished by their degree of ambiguity, which is the maximal number of derivation trees for the words generated by them. If there is no such upper bound G is said to be ambiguous of infinite degree. In [6] Maurer presents examples of context-free languages with inherent degree of ambiguity k for each $k \in \mathbb{N}$.

As a new tool for examining the ambiguity of cycle-free context-free grammars with infinite degree of ambiguity the ambiguity function am, which is total for cycle-free grammars, is introduced. It maps the natural number n to the maximal number of derivation trees which a word of at most length n may have. This new approach provides the ability to distinguish infinitely ambiguous grammars by the growth-rate of their ambiguity functions.

We present a necessary and sufficient, but in general undecidable, criterion for the exponential ambiguity of cycle-free context-free grammars. In fact violation of this criterion leads to a polynomial upper bound for the ambiguity, which can be effectively constructed from the grammar.

Hence for cycle-free context-free grammars the ambiguity function is either an element of $2^{\Theta(n)}$ or of $\mathcal{O}(n^d)$ for some $d \in \mathbb{N}_0$. This means that there is a gap between polynomial and exponential ambiguity. For example a function of growth $2^{\sqrt{\Theta(n)}}$ cannot be the ambiguity function of any cycle-free context-free grammar.

2 Preliminaries

Let Σ be a finite alphabet. For words $u, v \in \Sigma^*$, a symbol $a \in \Sigma$, and $n \in \mathbb{N}$ the length of u is denoted by $|u|$ and the number of a's in u is denoted by $|u|_a$. The empty word is denoted by ε. Σ^n and $\Sigma^{\leq n}$ denote all words over Σ of length n and of length up to n, respectively. The word v is a prefix of u, denoted by $v \leq u$, if and only if $u = vz$ for some $z \in \Sigma^*$. It is a proper prefix, written $v < u$, if $z \in \Sigma^+$. The cardinality of a set S is denoted by $|S|$.

A *context-free grammar* is a quadruple $G = (N, \Sigma, P, S)$, where N and Σ are finite disjoint alphabets of nonterminals and terminals, respectively, $S \in N$ is the start symbol, and $P \subseteq N \times (N \cup \Sigma)^*$ is a finite set of productions. We usually write $A \to \alpha$ or $(A \to \alpha)$ for the pair (A, α). If $f = (A \to \alpha) \in P$ we say that A is the left-hand side of f and α is the right-hand side of f. The set of productions from P with the left-hand side A is denoted by P_A.

For a context-free grammar $G = (N, \Sigma, P, S)$ and $\alpha, \beta \in (N \cup \Sigma)^*$, we say that α derives β in one step, denoted by $\alpha \Rightarrow_G \beta$, if there are $\alpha_1, \alpha_2, \gamma \in (N \cup \Sigma)^*$ and $A \in N$ such that $\alpha = \alpha_1 A \alpha_2$, $\beta = \alpha_1 \gamma \alpha_2$ and $(A \to \gamma) \in P$. Let $\Rightarrow_G^+$ denote the transitive closure of $\Rightarrow_G$ and $\Rightarrow_G^*$ denote the reflexive closure of $\Rightarrow_G^+$. We say that α derives β if $\alpha \Rightarrow_G^* \beta$. Thus the language generated by G is described by

$$L(G) = \{w \in \Sigma^* | S \Rightarrow_G^* w\}.$$

If the grammar is clear from the context, the subscript is omitted.

A language L is said to be *context-free* if there is a context-free grammar G with $L = L(G)$. Let $G = (N, \Sigma, P, S)$ be a context-free grammar and $\alpha \in (N \cup \Sigma)^*$. We say that α is a *sentential form of* G if $S \Rightarrow^* \alpha$. The grammar G is said to be *cycle-free* if there is no $A \in N$ such that $A \Rightarrow^+ A$. An $A \in N$ is said to be *useful* if there are strings $\alpha, \beta \in (N \cup \Sigma)^*$ and a word $w \in T^*$ such that $S \Rightarrow^* \alpha A \beta \Rightarrow^* w$. A is *useless* if it is not useful. A is said to be an *ε-symbol* if no nonempty string of terminals can be derived from A. A production $f \in P$ is said to be an *ε-production* if $f \in (N \times \{\varepsilon\})$. In contrast to the elimination of ε-productions the elimination of ε-symbols does not increase the size of the grammar. Note that ε-symbols can be deleted from the right-hand side of all productions without effecting the generated language. Afterwards they are useless and all the productions with an ε-symbol on the left-hand side can be eliminated obtaining a smaller grammar which may still contain ε-productions, but no more ε-symbols.

If not stated otherwise we assume throughout this paper that $G = (N, \Sigma, P, S)$ is an arbitrary context-free grammar and that $A \in N$ is an arbitrary nonterminal.

The ambiguity of a word $w \in \Sigma$ is the number of derivation trees for w. Formally we can count the number of derivation trees for w by counting the number of leftmost derivations (see the definition below). This is possible since there is a one-to-one correspondence between derivation trees and the leftmost derivations for words consisting only of *terminals*. In this paper we will consider pumpable trees, which contain at least one leaf labeled with a nonterminal. Since there are trees with nonterminals in the frontier which cannot be generated by a leftmost derivation we define a generalization of the leftmost derivation, called leftskip derivation. This derivation proceeds from left to right marking the current position by a dot. In each step either a production from P is applied to the nonterminal next in line or this nonterminal is skipped, which means that it remains in the sentential form and the dot is advanced to the next nonterminal. If no further nonterminal has to be considered, the dot is removed and the derivation is complete. For trees with a terminal frontier the leftmost and the leftskip derivation are identical.

Notation 2.1. *We fix $\{\bullet, s\}$ to be an alphabet which is disjoint from each finite set used in the definition of a context-free grammar throughout the paper.*

For all $\alpha, \beta \in (N \cup \Sigma)^*$ we define

$$\alpha \bullet_{\rightarrow} \beta := \begin{cases} \alpha w \bullet \gamma & \textit{if} \quad \beta = w\gamma \textit{ for some } w \in \Sigma^* \textit{ and } \gamma \in N(N \cup \Sigma)^* \\ \alpha\beta & \textit{otherwise} \end{cases}$$

We use this notation to "find" the next nonterminal from a given position if there is any. In the next definition the symbol s is used as the name of the skipping rule. To outline the relationship between the leftskip (ls) and the leftmost (lm) derivation we define them in parallel.

Definition 2.2. *For every $\alpha, \alpha_1, \alpha_2 \in (N \cup \Sigma)^*$, $A \in N$, $f \in P$ we define*

$$\begin{array}{llllll} (i) & \alpha_1 A \alpha_2 & \Rightarrow^f_{G,lm} & \alpha_1 \alpha \alpha_2 & \textit{if} & f = (A \to \alpha) \in P \quad \wedge \quad \alpha_1 \in \Sigma^* \\ (ii) & \alpha_1 \bullet A \alpha_2 & \Rightarrow^f_{G,ls} & \alpha_1 \bullet_{\rightarrow} \alpha \alpha_2 & \textit{if} & f = (A \to \alpha) \in P \\ (iii) & \alpha_1 \bullet A \alpha_2 & \Rightarrow^s_{G,ls} & \alpha_1 A \bullet_{\rightarrow} \alpha_2 & & \end{array}$$

We can distinguish between leftmost and leftskip derivations by the existence of the dot. Therefore we often omit the subscript.

We extend this definition of the one step derivation in the natural way to $\pi \in (P \cup \{s\})^*$ such that π encodes a sequence of one step derivations.

We summarize some of the properties of the leftskip derivation in the following observation.

Observation 2.3. *Let $A, B \in N$; $\alpha, \beta, \gamma, \delta \in (N \cup \Sigma)^*$; $\pi \in (P \cup \{s\})^*$; $\tau \in P^*$ and $w \in \Sigma^*$. Then:*

$$\begin{array}{rcl c rcl} \alpha \bullet A\beta & \Rightarrow^\pi & \gamma \bullet B\delta & \curvearrowright & \alpha & \leq & \gamma \\ \bullet_{\rightarrow}\alpha & \Rightarrow^\tau & \bullet_{\rightarrow}\beta & \mathrel{\curvearrowleft\!\!\curvearrowright} & \alpha & \Rightarrow^\tau & \beta \\ \bullet A & \Rightarrow^\pi & w & \mathrel{\curvearrowleft\!\!\curvearrowright} & A & \Rightarrow^\pi & w \end{array}$$

Next we define the set of parses for a sentential form β derived from A and the ambiguity of β.

Definition 2.4. *Let $\beta \in (N \cup \Sigma)^*$.*

$$\begin{array}{rcl} parse_G(A, \beta) & := & \{\pi \in (P \cup \{s\})^* \mid \bullet A \Rightarrow^\pi \beta\} \\ am_G(A, \beta) & := & |parse_G(A, \beta)| \end{array}$$

We call $\pi \in (P \cup \{s\})^*$ a parse *if $\pi \in parse_G(A, \beta)$ for some $\beta \in (N \cup \Sigma)^*$. Note that*

(i) after applying a parse to a nonterminal the dot marker disappears;

(ii) that each derivation tree, no matter whether or not there are nonterminals at the frontier, has a unique parse;

(iii) no proper prefix of a parse can ever be a parse;

(iv) $am_G(A, \beta)$ can be infinite if G is not cycle-free, and it is zero if and only if β cannot be derived from A.

Now we define the ambiguity function of G.

Definition 2.5. *Let $G = (N, \Sigma, P, S)$ be a cycle-free context-free grammar. Let $A \in N$, $L \subseteq (N \cup \Sigma)^*$ and $n \in \mathbb{N}$. Then*

$$am_G(n) \quad := \quad \max_{w \in \Sigma^{\leq n}} am_G(S, w).$$

Note that by definition the ambiguity of sentential forms containing nonterminals is not taken into account for the ambiguity function.

Definition 2.6. *A cycle-free context-free grammar G is said to be of* polynomially bounded ambiguity *if $am_G(n) = \mathcal{O}(n^k)$ for some $k \in \mathbb{N}$. It is said to be* exponentially ambiguous *if $am_G(n) = 2^{\Omega(n)}$.*

Definition 2.7. *A cycle-free context-free grammar $G = (N, \Sigma, P, S)$ satisfies criterion $\mathcal{C}_{exp}$, shortly denoted by $\mathcal{C}^G_{exp}$, if the following condition is satisfied:*

$$\exists A \in N;\ \alpha, \beta \in (N \cup \Sigma)^* :\ am_G(A, \alpha A \beta) > 1$$
$$\text{and } \alpha A \beta \text{ contains neither useless nor } \varepsilon\text{-symbols.}$$

3 Criterion $\mathcal{C}_{exp}$ is sufficient for exponential ambiguity

In [4] Kemp presents a decidable sufficient criterion for ambiguity. In the proof he constructs two derivation trees τ_1 and τ_2 such that the "concatenation" $\tau_1\tau_2$ of these two trees and the concatenation $\tau_2\tau_1$ yield two different derivation trees with the same frontier. We generalize this idea, sacrificing decidability, to characterize exponentially ambiguous grammars.

We will prove that a cycle-free context-free grammar G is exponentially ambiguous if and only if it satisfies $\mathcal{C}_{exp}$ (see Def 2.7). In other words, there are two different derivation trees with a common root and a common frontier such that the frontier does not contain useless or ε-symbols and it contains at least one node labeled with the same nonterminal as the root. If this criterion is satisfied we can pump up the ambiguity by concatenating these trees and choose randomly in each step the first or the second one. Thus the number of combinations grows exponentially with respect to the length of the frontiers obtained by this method. In this section we prove the sufficiency of the criterion, and in the next one the necessity. There are two problems to take care of in this section.

- We have to prove that different combinations always yield different trees. It is not obvious that the trees cannot "commute".
- Ambiguity of a grammar is defined by the number of derivation trees for *terminal* strings only. There are examples of unambiguous grammars (on the terminal strings) with exponential ambiguity in the sentential forms. Thus we have to find conditions under which the ambiguity of each sentential form is carried to a terminal strings which is longer at most by a constant factor.

The proof is divided in an algebraic and a combinatorial part.

3.1 The free monoid of pumping trees

A pumping tree for a nonterminal A is a derivation tree with the root A and at least one leaf labeled A. One of those leaves is designated to be the linkage point (called link) for further pumping trees. We concatenate two pumping trees by identifying the root of the second one with the link of the first. This yields a pumping tree that inherits the root from the first and the link from the second pumping tree. The parse of a pumping tree is factorized by the link into a derivation before and a derivation after skipping the link. We now define pumping trees and their concatenation formally.

Definition 3.1. *A pumping tree of G with root A is a pair $\vartheta = (\pi, \tau)$ where $\pi, \tau \in (P \cup \{s\})^*$ such that*

- $A \Rightarrow^{\pi} \alpha \bullet A\gamma \Rightarrow^{s} \alpha A \bullet_{\rightarrow}\gamma \Rightarrow^{\tau} \alpha A \beta$ *for some $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$.*

The set of pumping trees of G with root A is denoted by Φ_G^A. We write λ for $(\varepsilon, \varepsilon)$. One can easily verify that $\lambda \in \Phi_G^A$. A pumping tree $\vartheta \in \Phi_G^A$ is said to be a proper pumping tree if $\vartheta \neq \lambda$.

For a given pumping tree the strings α, β, and γ in the definition above are uniquely specified. In some proofs we will consider these strings. Therefore we define the following mappings:

Definition 3.2. *Let $\vartheta := (\pi, \tau) \in \Phi_G^A$. Then we define α_G^A, β_G^A, and γ_G^A as mappings $\Phi_G^A \to (N \cup \Sigma)^*$ such that*

$$\bullet A \Rightarrow^{\pi} \alpha_G^A(\vartheta) \bullet A\gamma_G^A(\vartheta) \text{ and } \bullet_{\to}\gamma_G^A(\vartheta) \Rightarrow^{\tau} \beta_G^A(\vartheta)$$

Definition 3.3. *The concatenation $\odot$ of $\vartheta_1, \vartheta_2 \in \Phi_G^A$, where $\vartheta_1 := (\pi_1, \tau_1)$ and $\vartheta_2 := (\pi_2, \tau_2)$ for some $\pi_1, \pi_2, \tau_1, \tau_2 \in (P \cup \{s\})^*$, is defined as follows:*

$$\vartheta_1 \odot \vartheta_2 := (\pi_1\pi_2, \tau_2\tau_1).$$

The symbol $\odot$ is often omitted.

One can easily verify that the concatenation of two pumping trees again yields a pumping tree.

Observation 3.4. *Let $\vartheta_1, \vartheta_2 \in \Phi_G^A$. Then $\alpha_G^A(\vartheta_1\vartheta_2) = \alpha_G^A(\vartheta_1)\alpha_G^A(\vartheta_2)$, $\beta_G^A(\vartheta_1\vartheta_2) = \beta_G^A(\vartheta_2)\beta_G^A(\vartheta_1)$, $\gamma_G^A(\vartheta_1\vartheta_2) = \gamma_G^A(\vartheta_2)\gamma_G^A(\vartheta_1)$.*

Theorem 3.5. *$(\Phi_G^A, \odot)$ is a monoid with left and right cancellation.*

Proof. The concatenation of pumping trees is an operation on the set Φ_G^A with the neutral element λ. The operation $\odot$ is associative, since $\odot$ performs a component-wise concatenation of strings which is an associative operation. Similarly left and right cancellation are inherited. □

Definition 3.6. *Let $\vartheta := (\pi, \tau) \in \Phi_G^A$. The size of ϑ is defined by $|\vartheta| := |\pi| + |\tau|$.*

Observation 3.7. *Let $\vartheta, \vartheta_1, \vartheta_2 \in \Phi_G^A$. Then $|\vartheta_1\vartheta_2| = |\vartheta_1| + |\vartheta_2|$. Moreover $|\vartheta| = 0$ if and only if $\vartheta = \lambda$.*

Definition 3.8. *A pumping tree $\vartheta \in \Phi_G^A \setminus \{\lambda\}$ is prime if and only if it is not the product of two proper pumping trees. That is:*

$$\forall \vartheta_1, \vartheta_2 \in \Phi_G^A \setminus \{\lambda\} : \vartheta_1\vartheta_2 \neq \vartheta.$$

The set of prime pumping trees is denoted by Ψ_G^A.

A pumping tree $\chi \in \Phi_G^A$ is prime if and only if on the unique path from the link to the root there is no other node labeled with A.

We can factorize $\chi \in \Phi_G^A$ by the following method. If the link is already the root, then $\chi = \lambda$ and we are done. Otherwise we move up along the unique path from the link of χ to the root until we meet the first node labeled with A. Since the root is labeled A, there is surely such a node. We split χ at this position into $\vartheta \in \Phi_G^A$ and $\theta \in \Psi_G^A$ such that $\chi = \vartheta\theta$. Thus we can split off a prime pumping tree from the right side of χ. We must not skip any occurrence of an A on the path because then one factor would not be prime. Now we proceed recursively with ϑ. Since the path to the root has finite length, the algorithm eventually stops.

We exploit this idea without formalizing derivation trees and the indicated algorithm.

Theorem 3.9. *Each pumping tree $\chi \in \Phi_G^A \setminus \lambda$ has a unique prime factorization, that is, $\exists! k \in \mathbb{N} : \exists! \theta_1, \ldots, \theta_k \in \Psi_G^A : \chi = \theta_1 \odot \cdots \odot \theta_k$*

Proof. Assume $|\chi| = 1$. Then χ is prime by Observation 3.7 and hence χ is already a prime factorization with $k = 1$. Now by the definition of primeness this is the only factorization. Assume the claim has been proved for all χ with $|\chi| \leq n$.

Now let $|\chi| = n + 1$. Again if χ is prime, it is already the unique prime factorization. Otherwise $\chi = \vartheta_1\vartheta_2$ for some $\vartheta_1, \vartheta_2 \in \Phi_G^A \setminus \lambda$. Now by Observation 3.7 we have $|\vartheta_2| \leq n$. Hence by the inductive hypothesis $\vartheta_2 = \theta_1 \odot \cdots \odot \theta_{\bar{k}}$ for some $\bar{k} \in \mathbb{N}$ and some $\theta_1, \ldots, \theta_{\bar{k}} \in \Psi_G^A$. Let $\vartheta := \vartheta_1\theta_1 \odot \cdots \odot \theta_{\bar{k}-1}$ and $\theta := \theta_{\bar{k}}$. Then by the associativity of the concatenation $\chi = \vartheta_1\vartheta_2 = \vartheta_1 \odot (\theta_1 \odot \cdots \odot \theta_{\bar{k}}) = (\vartheta_1 \odot \theta_1 \odot \cdots \odot \theta_{\bar{k}-1}) \odot \theta_{\bar{k}} = \vartheta\theta$. Thus we can split off a prime pumping tree from χ. Now assume we also have $\chi = \bar{\vartheta}\bar{\theta}$ for some $\bar{\vartheta} \in \Phi_G^A$ and some $\bar{\theta} \in \Psi_G^A$.

$$\begin{array}{c}\text{Let } \pi_l, \pi_r, \bar{\pi}_l, \bar{\pi}_r, \tau_l, \tau_r, \bar{\tau}_l, \bar{\tau}_r \in (P \cup \{s\})^*;\\ \alpha_l, \alpha_r, \bar{\alpha}_l, \bar{\alpha}_r, \beta_l, \beta_r, \bar{\beta}_l, \bar{\beta}_r, \gamma_l, \gamma_r, \bar{\gamma}_l, \bar{\gamma}_r \in (N \cup \Sigma)^* \text{ such that}\end{array}$$

$$\begin{array}{rclrclrclrcl}
\vartheta &=& (\pi_l, \tau_l), & \alpha_G^A(\vartheta) &=& \alpha_l, & \beta_G^A(\vartheta) &=& \beta_l, & \gamma_G^A(\vartheta) &=& \gamma_l,\\
\theta &=& (\pi_r, \tau_r), & \alpha_G^A(\theta) &=& \alpha_r, & \beta_G^A(\theta) &=& \beta_r, & \gamma_G^A(\theta) &=& \gamma_r,\\
\bar{\vartheta} &=& (\bar{\pi}_l, \bar{\tau}_l), & \alpha_G^A(\bar{\vartheta}) &=& \bar{\alpha}_l, & \beta_G^A(\bar{\vartheta}) &=& \bar{\beta}_l, & \gamma_G^A(\bar{\vartheta}) &=& \bar{\gamma}_l,\\
\bar{\theta} &=& (\bar{\pi}_r, \bar{\tau}_r), & \alpha_G^A(\bar{\theta}) &=& \bar{\alpha}_r, & \beta_G^A(\bar{\theta}) &=& \bar{\beta}_r, & \gamma_G^A(\bar{\theta}) &=& \bar{\gamma}_r.
\end{array}$$

From $\vartheta\theta = \bar{\vartheta}\bar{\theta}$ we obtain by Observation 3.4:

$$\pi_l\pi_r = \bar{\pi}_l\bar{\pi}_r;\ \tau_r\tau_l = \bar{\tau}_r\bar{\tau}_l;\ \alpha_l\alpha_r = \bar{\alpha}_l\bar{\alpha}_r;\ \beta_r\beta_l = \bar{\beta}_r\bar{\beta}_l;\ \gamma_r\gamma_l = \bar{\gamma}_r\bar{\gamma}_l.$$

Without loss of generality we assume $\bar{\pi}_l \leq \pi_l$. Then $\pi_l = \bar{\pi}_l\pi$ and $\bar{\pi}_r = \pi\pi_r$ for some $\pi \in (P \cup \{s\})^*$. Therefore $\bullet A \Rightarrow^{\bar{\pi}_l} \bar{\alpha}_l \bullet A\bar{\gamma}_l \Rightarrow^{\pi} \alpha_l \bullet A\gamma_l$. Since $\pi < \bar{\pi}_r s\bar{\tau}_r$ the derivation cannot be a parse and we obtain $\bullet A \Rightarrow^{\pi} \alpha \bullet B\gamma$ for some $\alpha, \gamma \in (N \cup \Sigma)^*$ and $B \in N$. With $\bar{\alpha}_l \bullet A\bar{\gamma}_l \Rightarrow^{\pi} \bar{\alpha}_l\alpha \bullet B\gamma\bar{\gamma}_l = \alpha_l \bullet A\gamma_l$ we obtain

$$B = A,\ \alpha_l = \bar{\alpha}_l\alpha,\ \bar{\alpha}_r = \alpha\alpha_r,\ \gamma_l = \gamma\bar{\gamma}_l, \bar{\gamma}_r = \gamma_r\gamma.$$

Assume $\bar{\tau}_r < \tau_r$. Then $\gamma_r \Rightarrow^{\bar{\tau}_r} \delta_1 \bullet C\delta_2$ for some $\delta_1, \delta_2 \in (N \cup \Sigma)^*$ and $C \in N$. Hence we have $\bullet_{\rightarrow}\gamma_r\gamma \Rightarrow^{\bar{\tau}_r} \delta_1 \bullet C\delta_2\gamma \ \wedge\ \bullet_{\rightarrow}\gamma_r\gamma = \bullet_{\rightarrow}\bar{\gamma}_r \Rightarrow^{\bar{\tau}_r} \bar{\beta}_r$. This implies $\delta_1 \bullet C\delta_2\gamma = \bar{\beta}_r$, which is a contradiction, since $\bar{\beta}_r$ does not contain the dot symbol. Thus follows $\tau_r \leq \bar{\tau}_r$. Hence $\bar{\tau}_r = \tau_r\tau$ and $\tau_l = \tau\bar{\tau}_l$ for some $\tau \in (P \cup \{s\})^*$. Now $\bullet_{\rightarrow}\bar{\gamma}_r \Rightarrow^{\bar{\tau}_r} \bar{\beta}_r$ and $\bar{\tau}_r = \tau_r\tau$ implies $\bullet_{\rightarrow}\bar{\gamma}_r \Rightarrow^{\tau_r\tau} \bar{\beta}_r$. Thus $\bullet_{\rightarrow}\bar{\gamma}_r = \bullet_{\rightarrow}\gamma_r\gamma \Rightarrow^{\tau_r} \beta_r \bullet_{\rightarrow}\gamma \Rightarrow^{\tau} \bar{\beta}_r$. Hence $\bar{\beta}_r = \beta_r\beta$ for some $\beta \in (N \cup \Sigma)^*$. Now $\beta_r\beta_l = \bar{\beta}_r\bar{\beta}_l$ whence $\beta_l = \beta\bar{\beta}_l$.

Finally let $\xi := (\pi, \tau)$. Then $\bullet A \Rightarrow^{\pi} \alpha \bullet A\gamma \Rightarrow^{s} \alpha A \bullet_{\rightarrow}\gamma \Rightarrow^{\tau} \alpha A\beta$ implies $\xi \in \Phi_G^A$. Thus $\xi\theta = (\pi\pi_r, \tau_r\tau) = (\bar{\pi}_r, \bar{\tau}_r) = \bar{\theta}$. But $\bar{\theta}$ and θ are assumed to be prime. Hence $\xi = \lambda$ and $\theta = \bar{\theta}$ follow. By right cancellation we obtain $\vartheta = \bar{\vartheta}$. Hence each prime factorizations of χ has θ as the rightmost factor and the factors before form a prime factorization of ϑ. Since $|\vartheta| \leq n$, by the inductive hypothesis ϑ has a unique prime factorization, that is, $\exists! \tilde{k} \in \mathbb{N} : \exists! \theta_1, \ldots, \theta_{\tilde{k}} \in \Psi_G^A$, such that $\vartheta = \theta_1 \odot \cdots \odot \theta_{\tilde{k}}$. Let $k = \tilde{k} + 1$ and $\theta_k := \theta$ then the unique prime factorization of χ is $\theta_1 \odot \cdots \odot \theta_k$. □

From Theorem 3.5 and 3.9 we immediately obtain the following theorem.

Theorem 3.10. *Φ_G^A is a free monoid over Ψ_G^A.*

Thus we can apply [2] (Corollary of Theorem 1.3.3 and Theorem 1.3.4) and obtain:

Theorem 3.11. *Let $\vartheta_1, \vartheta_2 \in \Phi_G^A$. Then*

(i) $\vartheta_1\vartheta_2 = \vartheta_2\vartheta_1 \quad \curvearrowright \quad \vartheta_1 = \vartheta^k$ *and* $\vartheta_2 = \vartheta^l$ *for some* $\vartheta \in \Phi_G^A$ *and some* $k, l \in \mathbb{N}$.
(ii) $\vartheta_1\vartheta_2 \neq \vartheta_2\vartheta_1 \quad \curvearrowright \quad \{\vartheta_1, \vartheta_2\}^*$ *is a free submonoid of* Φ_G^A.

3.2 Pumping up ambiguity

The definition of pumping trees is tailored to allow a simple formalization of the concatenation. For the combinatorial part we now switch to another representation.

Definition 3.12. *Let $\vartheta := (\pi, \tau) \in \Phi_G^A$. We define the parse of ϑ by $p(\vartheta) = \pi s \tau$ and the interface by $i(\vartheta) = (\alpha_G^A(\vartheta), \beta_G^A(\vartheta))$.*

While the parse completely presents the internal structure of a pumping tree, it has no information about the linkage position at its frontier. For example let $\omega = p(\vartheta)$ for some $\vartheta \in \Phi_G^A$. Let $\alpha \in (N \cup \Sigma)^*$ be the corresponding frontier. Then there are $|\alpha|_A$ many different pumping trees with the parse ω, since each occurrence of A in α can be the link. The interface on the other hand represents the linkage position by a factorization of the frontier which does not necessarily determine the internal structure. But we see easily

Observation 3.13. *Two pumping trees $\vartheta_1, \vartheta_2 \in \Phi_G^A$ are equal if and only if they have the same parse and the same interface.*

Now we pump up the ambiguity of sentential forms.

Lemma 3.14. *Let $\vartheta_1, \vartheta_2 \in \Phi_G^A$; $n \in \mathbb{N}$; and $i(\vartheta_1) = i(\vartheta_2) = (\alpha, \beta)$ for some $\alpha, \beta \in (N \cup \Sigma)^*$. Then*

$$\vartheta_1\vartheta_2 \neq \vartheta_2\vartheta_1 \quad \curvearrowright \quad am_G(A, \alpha^n A \beta^n) \geq 2^n.$$

Proof. By Theorem 3.11 we know $\vartheta_1\vartheta_2 \neq \vartheta_2\vartheta_1$ implies $|\{\vartheta_1, \vartheta_2\}^n| = 2^n$. Additionally we have $i(\vartheta) = (\alpha^n, \beta^n)$ for all $\vartheta \in \{\vartheta_1, \vartheta_2\}^n$. Thus by Observation 3.13 the elements of $\{\vartheta_1, \vartheta_2\}^n$ have pairwise different parses. Hence $am_G(A, \alpha^n A \beta^n) \geq 2^n$. □

Different pumping trees $\vartheta_1, \vartheta_2 \in \Phi_G^A$ with a common interface can never commute in a cycle-free grammar:

Lemma 3.15. *Let $G = (N, \Sigma, P, S)$ be a* cycle-free *context-free grammar; $A \in N$ and $\vartheta_1, \vartheta_2 \in \Phi_G^A$. Then*

$$i(\vartheta_1) = i(\vartheta_2) \text{ and } \vartheta_1 \neq \vartheta_2 \quad \curvearrowright \quad \vartheta_1\vartheta_2 \neq \vartheta_2\vartheta_1$$

Proof. Assume $i(\vartheta_1) = i(\vartheta_2)$ and $\vartheta_1\vartheta_2 = \vartheta_2\vartheta_1$. Then by Theorem 3.11 $\vartheta_1 = \vartheta^k$ and $\vartheta_2 = \vartheta^l$ for some $\vartheta \in \Phi_G^A$ and some $k, l \in \mathbb{N}_0$. Now $\vartheta_1 \neq \vartheta_2$ implies $\vartheta \neq \lambda$ and $l \neq k$. For some $\alpha, \beta \in (N \cup \Sigma)^*$ we have $i(\vartheta) = (\alpha, \beta)$. Thus $i(\vartheta_1) = (\alpha^k, \beta^k) = (\alpha^l, \beta^l) = i(\vartheta_2)$. Since $k \neq l$ this implies $\alpha = \beta = \varepsilon$. Finally since $\vartheta \neq \lambda$, this implies $A \Rightarrow^+ A$, which is a contradiction to the cycle-freeness of G. □

Ambiguity of sentential forms, without useless and without ε-symbols, is carried over to terminal strings:

Lemma 3.16. *Let $\alpha \in (N \cup \Sigma)^*$ such that α contains no ε-symbols. Further let $h : (N \cup \Sigma)^* \to \Sigma^*$ be a morphism satisfying*

$$h(X) = \begin{cases} X & \text{if} \quad X \in \Sigma. \\ w \in \Sigma^+ \text{ where } X \Rightarrow^+ w & \text{if} \quad X \in N \text{ occurs in } \alpha. \end{cases}$$

Then $am_G(A, \alpha) \leq am_G(A, h(\alpha))$.

Proof. (sketch) Let $A, B \in N$, $u \in \Sigma^*$, $\beta, \gamma \in (N \cup \Sigma)^*$ and for $j \in \{1, 2\}$, let $\pi_j \in P^*$ and $\tau_j \in (P \cup \{s\})^*$ such that $\bullet\, A \Rightarrow^{\pi_j} u \bullet B\gamma \Rightarrow^s uB \bullet_{\to} \gamma \Rightarrow^{\tau_j} uB\beta$ and $\pi_1 s \tau_1 \neq \pi_2 s \tau_2$. Let $z = h(B)$ and $\omega \in (P \cup \{s\})^*$ such that $B \Rightarrow^\omega z$. Now assume $\pi_1 \omega \tau_1 = \pi_2 \omega \tau_2$, that is, by appending a given tree with a nonempty frontier to the first nonterminal in the common frontier of two different derivation trees, respectively, we obtain the same tree. We will prove that this is impossible if the frontier does not contain ε-symbols. First we observe that either $\pi_1 < \pi_2$ or $\pi_2 < \pi_1$. Without loss of generality we assume the first. It can be shown that no proper prefix of a parse ω is a suffix of ω. Thus, there is a $\rho \in (P \cup \{s\})^*$ such that $\pi_1\omega\rho = \pi_2$. Since $z \neq \varepsilon$ after applying $\pi_1\omega$ the dot in the derivation is advanced at least one position behind u. But by definition the dot can never move back to the left. Thus after applying $\pi_2 = \pi_1\omega\rho$ the dot is not immediately behind u which is a contradiction to the definition of π_2. Thus $\pi_1\omega\tau_1 \neq \pi_2\omega\tau_2$, that is, the ambiguity does not collapse, when a tr ee with a nonempty frontier is appended. Therefore $am(A, uB\beta) \leq am(A, uz\beta)$. Now for each $\alpha \in (N \cup \Sigma)^*$ we can easily prove by induction over the number of nonterminals in α that $am(A, \alpha) \leq am(A, h(\alpha))$. □

Note that if a sentential form contains an ε-symbol then the required ε-free homomorphism does not exist. In fact there are examples of context-free grammars where the set of sentential forms have higher ambiguity than the grammar itself. (Recall that the ambiguity of a grammar is defined by the ambiguity of the terminal strings only.)

Example 3.17. *Let $G = (\{S, A\}, \{a\}, \{f_1 = (S \to SaAA), (S \to \varepsilon), f_2 = (A \to \varepsilon)\}, S)$. For each $i \in \mathbb{N}$ we have $am_G(S, S(aA)^j) = 2^j$. But $L(G)$ is a^* and for each $j \in \mathbb{N}$ the word a^j has the unique parse $f_1^j f_2^{2j}$. Hence G is unambiguous.*

Theorem 3.18. *Let $G = (N, \Sigma, P, S)$ be a cycle-free context-free grammar.*

$$\mathcal{C}^G_{exp} \curvearrowright am_G(n) = 2^{\Omega(n)} \quad \textit{(see Def 2.5 and 2.7).}$$

Proof. Assume G satisfies criterion $\mathcal{C}_{exp}$. Then there are $\vartheta_1, \vartheta_2 \in \Phi^A_G$ such that $p(\vartheta_1) \neq p(\vartheta_2)$ and $i(\vartheta_1) = i(\vartheta_2) = (\alpha, \beta)$ for some $\alpha, \beta \in (N \cup \Sigma)^*$. Then $\vartheta_1 \neq \vartheta_2$ by Observation 3.13. By Lemma 3.15 this implies $\vartheta_1\vartheta_2 \neq \vartheta_2\vartheta_1$. Thus by Lemma 3.14 we obtain $am_G(A, \alpha^n A\beta^n) \geq 2^n$. Since A is not useless there are $u, v \in \Sigma^*$ such that $\bullet\, S \Rightarrow^+ uAv$. Now appending at the top is not critical, therefore $am_G(S, u\alpha^n A\beta^n v) \geq 2^n$. Since $u\alpha A\beta v$ contains neither useless nor ε-symbols we can define a morphism h that satisfies the conditions of Lemma 3.16. Thus we obtain $am_G(S, h(u\alpha^n A\beta^n v)) \geq 2^n$. Since $\alpha\beta \neq \varepsilon$ by the cycle-freeness of G we finally obtain $am_G(n) = 2^{\Theta(n)}$. □

The obvious way to prove $\mathcal{C}^G_{exp}$ for a cycle-free context-free grammar G is to look for two different pumping trees with common interface. But it is often helpful to have the following corollary, with slightly weaker conditions on the pumping trees, in mind.

Corollary 3.19. *Let $G = (N, \Sigma, P, S)$ be a cycle-free context-free grammar.*

$$\begin{aligned}&\big(\exists \vartheta_1, \vartheta_2 \in \Phi^A_G;\ \alpha, \beta \in (N \cup \Sigma)^*;\ k, l, m, n \in \mathbb{N}\ :\\ &i(\vartheta_1) = (\alpha^k, \beta^l)\ \wedge\ i(\vartheta_2) = (\alpha^m, \beta^n)\ \wedge\ \vartheta_1\vartheta_2 \neq \vartheta_2\vartheta_1\big) \quad \curvearrowright \quad \mathcal{C}^G_{exp}\end{aligned}$$

Proof. If the left-hand side is satisfied then $i(\vartheta_1\vartheta_2) = (\alpha^{k+m}, \beta^{l+n}) = i(\vartheta_2\vartheta_1)$. Thus $\vartheta_1\vartheta_2 \neq \vartheta_2\vartheta_1$ implies $p(\vartheta_1\vartheta_2) \neq p(\vartheta_2\vartheta_1)$ by Observation 3.13 and $\mathcal{C}^G_{exp}$ follows. If G satisfies $\mathcal{C}_{exp}$ then there are $\vartheta_1, \vartheta_2 \in \Phi^A_G$ such that $i(\vartheta_1) = i(\vartheta_2)$ and $p(\vartheta_1) \neq p(\vartheta_2)$. By Lemma 3.15 this implies $\vartheta_1\vartheta_2 \neq \vartheta_2\vartheta_1$. Thus for $k := l := m := n := 1$, $\alpha := \alpha^A_G(\vartheta_1)$ and $\beta := \beta^A_G(\vartheta_1)$ the left-hand side of the claim is satisfied. □

The following sufficient criterion for $\mathcal{C}^G_{exp}$ depends only on the interfaces of two pumping trees.

Corollary 3.20. *Let $G = (N, \Sigma, P, S)$ be a cycle-free context-free grammar.*

$$\begin{aligned}&\big(\exists \vartheta_1, \vartheta_2 \in \Phi^A_G;\ \alpha, \beta \in (N \cup \Sigma)^+;\ k, l, m, n \in \mathbb{N}\ :\\ &i(\vartheta_1) = (\alpha^k, \beta^l)\ \wedge\ i(\vartheta_2) = (\alpha^m, \beta^n)\ \wedge\ lm \neq kn\big) \quad \curvearrowright \quad \mathcal{C}^G_{exp}\end{aligned}$$

Proof. If the left-hand side is satisfied then $i(\vartheta_1\vartheta_2) = (\alpha^{k+m}, \beta^{l+n}) = i(\vartheta_2, \vartheta_1)$. Assume $p(\vartheta_1\vartheta_2) = p(\vartheta_2, \vartheta_1)$. Then since the interfaces coincide we obtain $\vartheta_1\vartheta_2 = \vartheta_2\vartheta_1$ by Observation 3.13. By Theorem 3.11 then $\vartheta_1 = \vartheta^p$ and $\vartheta_2 = \vartheta^q$ for some $p, q \in \mathbb{N}$ where $i(\vartheta) = (\bar{\alpha}, \bar{\beta})$ for some $\bar{\alpha}, \bar{\beta} \in (N \cup \Sigma)^+$. This implies $k\cdot|\alpha| = p\cdot|\bar{\alpha}| \wedge l\cdot|\beta| = p\cdot|\bar{\beta}| \wedge m\cdot|\alpha| = q\cdot|\bar{\alpha}| \wedge n\cdot|\beta| = q\cdot|\bar{\beta}|$. By multiplication we obtain $k\cdot|\alpha|\cdot n\cdot|\beta| = p\cdot|\bar{\alpha}|\cdot q\cdot|\bar{\beta}| = l\cdot|\beta|\cdot m\cdot|\alpha|$. Since $|\alpha| \cdot |\beta| > 0$ we obtain by devision $kn = lm$ which is a contradiction. Hence $p(\vartheta_1\vartheta_2) \neq p(\vartheta_2, \vartheta_1)$ and $\mathcal{C}^G_{exp}$ follows. □

Finally we prove a decidable sufficient criterion for exponential ambiguity, which is even for grammars in Chomsky normal form, a proper generalization of the ambiguity criterion presented by Kemp [4].

Corollary 3.21. *A cycle-free context-free grammar $G = (N, \Sigma, P, S)$ is exponentially ambiguous if it contains a nonterminal which is left as well as right recursive. Formally, that is,* $\exists A \in N;\ \alpha, \beta \in (N \cup \Sigma)^* : A \Rightarrow^+ \alpha A \ \wedge \ A \Rightarrow^+ A\beta) \quad \curvearrowright \quad \mathcal{C}^G_{exp}$.

Proof. If G is cycle-free and $A \in N$ is as well left and right recursive, then there are $\vartheta_1, \vartheta_2 \in \Phi^A_G$ such that $i(\vartheta_1) = (\gamma, \varepsilon)$ and $i(\vartheta_2) = (\varepsilon, \delta)$ for some $\gamma, \delta \in (N \cup \Sigma)^+$. Thus for $k := n := 1,\ l := m := 0,\ \alpha := \gamma$ and $\beta := \delta$ the left-hand side of Corollary 3.20 is satisfied an we obtain $\mathcal{C}^G_{exp}$.

□

4 Criterion $\mathcal{C}_{exp}$ is necessary for exponential ambiguity

In this section we prove that a cycle-free context-free grammar which does not satisfy $\mathcal{C}_{exp}$ (see Def 2.7) is at most polynomially ambiguous.

For a given derivation tree with root labeled A all but the subtree dominated by the last preorder occurrence of the symbol A is covered by a pumping tree. The violation of $\mathcal{C}_{exp}$ implies that this pumping tree is unambiguous. This observation applies recursively to the trees dominated by the direct descendants of the last preorder occurrence of A. But the descendants does not contain nodes labeled with A. We use this observation to prove our claim by induction on the number of nonterminals in the grammar.

By the following definitions we will consider two pumping trees as equivalent if the same production is applied to the last preorder occurrence of A and if this occurrence and its direct descendants impose the same factorization on the frontier.

Definition 4.1. *For each $w \in L(G)$, we define*

(i) $\bar{\Sigma} := \Sigma \cup \{\diamondsuit_f | f \in P\} \cup \{\diamondsuit\}$.

(ii) *For $f = (A \to u_0B_1u_1 \dots B_mu_m) \in P$ with $m \in \mathbb{N}$, $A, B_1, \dots, B_m \in N$, and $u_0, \dots, u_m \in \Sigma^*$ define $\bar{f} := (A \to \diamondsuit_f u_0B_1\diamondsuit \dots B_m\diamondsuit u_m)$ and $c_f := m + 1$, that is, c_f is the number of symbols from $\bar{\Sigma} - \Sigma$ on the right-hand side of $\bar{f}$.*

(iii) $\bar{P} := P \cup \{\bar{f} | f \in P\}$.

(iv) $\bar{G} := (N, \bar{\Sigma}, \bar{P}, S)$.

(v) *For $\pi = \omega f \tau \in parse(S, w)$, where $\omega \in P^*$, $f \in P_S$, and $\tau \in (P - P_S)^*$, define $\bar{\pi} = \omega\bar{f}\tau$. Note that $\bar{\pi}$ is well-defined, as f replaces exactly the last occurrence of a production with left-hand side S in the parse π.*

(vi) $\nu_w : parse(S, w) \to \bar{\Sigma}^* \ : \ \nu_w(\pi) = \bar{w}$, *where $\bar{w}$ is the unique word satisfying* $S \Rightarrow^{\bar{\pi}}_{lm,\bar{G}} \bar{w}$.

(vii) *For $\pi_1, \pi_2 \in parse(S, w)$ we say that π_1 is equivalent to π_2, in symbols $\pi_1 \cong_w \pi_2$, if and only if $\nu_w(\pi_1) = \nu_w(\pi_2)$ (obviously $\cong_w$ is an equivalence relation).*

(viii) *$index(\cong_w)$ is the number of equivalence classes of $\cong_w$.*

(ix) $parse_{G,f}(S, w) := \{\pi \in parse(S, w) | \pi = \omega f \tau$ *for some* $\omega \in P^*$, $\tau \in (P - P_S)^*\}$. *The subscript G is omitted when it is clear from the context.*

Theorem 4.2. *Let $c = max\{c_f | f \in P_S\}$. Then for each $w \in L(G)$,*

$$index(\cong_w) \leq \sum_{f \in P_S} \binom{|w| + c_f}{c_f} = \mathcal{O}(|w|^c).$$

Proof. Let $w \in L(G)$ and $f \in P_S$; $\pi_1, \pi_2 \in parse_f(S, w)$ such that $w_1 := \nu_w(\pi_1) \neq \nu_w(\pi_2) =: w_2$. Then w_1 and w_2 have the form

$$w_1 = u_0 \diamondsuit_f u_1 \diamondsuit \ldots \diamondsuit u_{c_f} \text{ and } w_2 = v_0 \diamondsuit_f v_1 \diamondsuit \ldots \diamondsuit v_{c_f}$$

with $u_0 \ldots u_{c_f} = v_0 \ldots v_{c_f} = w$. That is, w_1 and w_2 differ in the positions of symbols from $\bar{\Sigma} - \Sigma$ which are c_f many symbols out of $|w| + c_f$. Thus

$$\Big|\{\nu_w(\pi) | \pi \in parse_f(S, w)\}\Big| \leq \binom{|w| + c_f}{c_f}.$$

As $parse(S, w) = \bigcup_{f \in P_S} \; parse_f(S, w)$ we finally obtain

$$index(\cong_w) \leq \sum_{f \in P_S} \binom{|w| + c_f}{c_f} \leq |P_S| \cdot \binom{|w| + c}{c} = \mathcal{O}(|w|^c)$$

□

Theorem 4.3. *For each context-free grammar $G = (N, \Sigma, P, S)$ that does not satisfy $\mathcal{C}_{exp}$ and each word $w \in L(G)$, the number of derivation trees for w is bounded by an expression which is polynomial in $|w|$.*

Proof. We proceed by induction on $|N|$.

Assume $|N| = 1$. Let $f \in P$. Assume $\pi_1, \pi_2 \in parse_{G,f}(S, w)$ are equivalent parses, that is, $\nu_w(\pi_1) = v_0 \diamondsuit_f u v_1 = \nu_w(\pi_2)$, where $f = (S \to u)$. By the definition of $parse_{G,f}(S, w)$ we have $\pi_1 = \pi_1' f \tau_1$ and $\pi_2 = \pi_2' f \tau_2$ with $\pi_1', \pi_2' \in P^*$ and $\tau_1, \tau_2 \in (P - P_S)^*$. But $P - P_S = \emptyset$. Hence $\tau_1 = \tau_2 = \varepsilon$.

Thus $\pi_1 = \pi_1' f$ and $\pi_2 = \pi_2' f$, implying that $\bar{\pi}_1 = \pi_1' \bar{f}$ and $\bar{\pi}_2 = \pi_2' \bar{f}$. Now we get $S \Rightarrow^{\pi_1'} v_0 S v_1$ and $S \Rightarrow^{\pi_2'} v_0 S v_1$. Thus $\pi_1' = \pi_2'$, because otherwise $\mathcal{C}_{exp}$ would be satisfied. Hence every equivalence class of $\cong_w$ contains at most one parse. By Theorem 4.2 this means that the number of derivation trees is bounded by a polynomial expression in $|w|$.

Assume the claim is true for $|N| \leq n$. Let $|N| = n+1$ and $f = (S \to u_0 B_1 u_1 \ldots B_m u_m) \in P_S$ for some $m \in \mathbb{N}$.
Assume further that $\pi_1, \pi_2 \in parse_{G,f}(S, w)$ with $\pi_1 \neq \pi_2$ are equivalent. Then

$$\nu_w(\pi_1) = v_0 \diamondsuit_f u_0 v_1 \diamondsuit \ldots u_{m-1} v_m \diamondsuit u_m v_{m+1} = \nu_w(\pi_2).$$

For $k \in \{1, 2\}$ we can write π_k as $\omega_k f \pi_{k1} \ldots \pi_{km} \tau_k$ such that

$$\begin{gathered}(\omega_k, \tau_k) \in \Phi_G^S \;\wedge\; i((\omega_k, \tau_k)) = (v_0, v_{m+1}) \\ S \Rightarrow^{\bar{f}} \diamondsuit_f u_0 B_1 \diamondsuit \ldots u_{m-1} B_m \diamondsuit u_m \text{ and} \\ \pi_{kj} \in parse_G(B_j, v_j) \text{ for each } 1 \leq j \leq m.\end{gathered}$$

Thus $p((\omega_1, \tau_1)) = p((\omega_2, \tau_2))$, because otherwise $\mathcal{C}_{exp}$ would be satisfied. Thus all equivalent derivations of w differ only in the π_{kj} which are parses not containing rules from P_S. For $1 \leq j \leq m$ we define $G_{B_j} := (N \setminus S, \Sigma, P - P_S, B_j)$. Then $\pi_{kj} \in parse_{G_{B_j}}(B_j, v_j)$. For each $1 \leq j \leq m$, the grammar G_{B_j} has n nonterminals and therefore by the induction hypothesis $parse_{G_{B_j}}(B_j, v_j)$ is bounded by an expression which is polynomial in $|v_j|$. Thus the number of parses which are in the same equivalence class as π_1 is bounded by the product of m polynomial expressions in $|w|$. Finally, as index $(\cong_w)$ is polynomial (Theorem 4.2), the claim follows. □

Actually it can be shown that for each context-free grammar $G = (N, \Sigma, P, S)$ not satisfying $\mathcal{C}_{exp}$ we have $am_G(n) = \mathcal{O}(n^{2c-1})$, where $c := \sum_{k=0}^{|N|-1} l^k$ and l is the maximal number of nonterminals on the right-hand sides of the productions.

From Theorems 3.18 and 4.3 we obtain the following.

Theorem 4.4. *A context-free grammar is exponentially ambiguous if and only if it satisfies* $\mathcal{C}_{exp}$.

5 $\mathcal{C}_{exp}$ is undecidable

Definition 5.1. *A cycle-free context-free grammar is* extreme *if it is unambiguous or exponentially ambiguous.*

Theorem 5.2. *For the class of extreme cycle-free context-free grammars* $\mathcal{C}_{exp}$ *is undecidable.*

Proof. Let $\Gamma := \{a, b, c, d, \#, A, B, C, S\}$ be an alphabet. Let $\Sigma := \{a, b, c, d, \#\}$ Let $x_1, \dots, x_n,\ y_1, \dots y_n \in \{a, b\}^+$. And let

$$\begin{aligned} P_1 &:= \{S \to A,\ S \to B\} \cup \{A \to x_j A d c^j \mid 1 \le j \le n\} \cup \{B \to y_j A d c^j \mid 1 \le j \le n\}, \\ P_2 &:= \{A \to x_j d c^j \mid 1 \le j \le n\} \cup \{B \to y_j d c^j \mid 1 \le j \le n\}, \\ \bar{P}_2 &:= \{A \to x_j C d c^j \mid 1 \le j \le n\} \cup \{B \to y_j C d c^j \mid 1 \le j \le n\} \cup \{C \to \varepsilon\}. \end{aligned}$$

In [2] (Theorem 8.4.5) it is shown that the grammar $G_1 := (\{S, A, B, \}, \Sigma, P_1 \cup P_2, S)$ is ambiguous if and only if the instance $((x_1, \dots, x_n),(y_1, \dots y_n))$ of Post Correspondence Problem has a solution. The same argument applies to $G_2 := (\{S, A, B, C\}, \Sigma, P_1 \cup \bar{P}_2, S)$. We only guarantee to terminate with a unique ε-production. Now consider $G_3 = (\{S, A, B, C\}, \Sigma, P_1 \cup \bar{P}_2 \cup \{C \to \#S\#\}, S)$. It is easily seen that the production $C \to \#S\#$ does not introduce ambiguity. But if G_1 is already ambiguous then G_3 satisfies $\mathcal{C}_{exp}$ (see Def 2.7). Hence G_3 is extreme, but it is not decidable whether it is unambiguous or exponentially ambiguous. □

6 Conclusion

Our main goal was to prove the necessity and sufficiency of the criterion $\mathcal{C}_{exp}$ (see Def 2.7) for exponential ambiguity. Since a grammar which violates $\mathcal{C}_{exp}$ is at most polynomially ambiguous, we have proved an even stronger result:

Corollary 6.1. *Each context-free grammar is exponentially ambiguous if it satisfies* $\mathcal{C}_{exp}$ *and it is at most polynomially ambiguous otherwise.*

In [7] Naji explored inherent ambiguity. A context-free language L has inherent ambiguity $\Theta(f)$ for a total function $f : \mathbb{N} \to \mathbb{R}_+$ if there is a grammar G_0 with an ambiguity function from $\mathcal{O}(f)$ and each grammar that generates L has an ambiguity function from $\Omega(f)$. Naji presents examples with inherent ambiguity from $2^{\Theta(n)}$ and inherent ambiguities from $\Theta(n^k)$ for each $k \in \mathbb{N}$. Hence there is an infinite proper hierarchy.

Criterion $\mathcal{C}_{exp}$ requires the existence of two different pumping trees having the same root and frontier. This is remarkably close to the definition of ambiguity itself. The only additional requirement is that pumping trees are always involved. But pumping is very natural for context-free grammars. In section 4 we saw that in grammars which violate $\mathcal{C}_{exp}$ each derivation tree is mostly covered by “unambiguous” pumping trees. This shows that violation of $\mathcal{C}_{exp}$ is a strong restriction for context-free grammars. Therefore it is an interesting field of research.

The class of languages with subexponential ambiguity has some natural properties:

- In [1] Crestin has shown that the complete hierarchy of finite degrees can be skipped by a single concatenation. The author exploits the language $C := \{uv \mid u = u^R \vee v = v^R\}$ which already has ambiguity of inherent infinite degree. But $C = L_p^2$ where $L_p := \{u \mid u = u^R\}$ is the unambiguous language of palindromes. From our point of view C has linear ambiguity which is a moderate growth-rate. This leaves room for further distinctions. For example I strongly conjecture that L_p^{k+1} has ambiguity $\Theta(n^k)$ for each $k \in \mathbb{N}$.
- With respect to closure properties languages with polynomially bounded ambiguity fit nicely into the hierarchy depicted by the following table:

	$\cup$	$\cdot$	$*$
unambiguous context-free languages	$-$	$-$	$-$
context-free languages with finite degree of ambiguity	$+$	$-$	$-$
context-free languages where *am* is polynomially bounded	$+$	$+$	$-$
context-free	$+$	$+$	$+$

  where "$\cup$" is the union, "$\cdot$" is the concatenation, and "$*$" is the Kleene-star-operation. The symbols "+" and "–" indicate whether or not a language class is closed under the corresponding operation.
- A new yet unpublished result is, that the class of languages with polynomial bounded ambiguity can be characterized as the closure of unambiguous languages under a restricted type of substitution operation.

Further questions under investigation are:

- Is the upper bound for polynomial ambiguity sharp? I strongly conjecture it is. Prof Y. Kobayashi [5] gave a first example of a linear grammar with only one nonterminal and with linear ambiguity. This is the highest possible subexponential ambiguity for such a grammar. The basic concept can be generalized to grammars with more nonterminals and higher branching (number of nonterminals on the right-hand side).
- What are the possible complexities for the ambiguity functions? This question is of particular interest, as contrary to my expectations languages with sublinear ambiguity have been found recently [10].
- To prove that a given language L is inherently ambiguous is usually tedious, especially when we prove that the complexity of a given function is unavoidable for the grammars that generate L. I strongly conjecture that for each context-free grammar G in Greibach normal form there is a language L ($\neq L(G)$) such that, for each grammar $\bar{G}$ with $L(\bar{G}) = L$, there is a constant $c_{\bar{G}} \in \mathbb{N}$ such that $am_{\bar{G}}(n) \geq am_G(c_{\bar{G}}n)$ for all but finitely many n. For the investigation of the spectrum of ambiguities we would obtain that all ambiguities of Greibach normal form grammars are inherent for some context-free languages, avoiding the tedious direct proof of inherent ambiguity.

Acknowledgments Thanks to Friedrich Otto, Gundula Niemann and Dieter Hofbauer for proofreading, valuable discussions and LaTeXtips.

References

[1] J.P. Crestin. Un langage non ambigu dont le carré est d'ambiguité non bornée. Automata, Languages and Programming (M. Nivat, ed.), pp. 377–390, Amsterdam, North-Holland, 1973.

[2] M.A. Harrison. *Introduction to Formal Language Theory.* Addison-Wesley, 1978

[3] J.E. Hopcroft, J.D. Ullman. *Introduction to Automata Theory, Formal Languages, and Computation.* Addison-Wesley, 1979.

[4] R. Kemp. Mehrdeutigkeit Kontextfreier Sprachen. *LNCS.*, vol 14, pp. 534–546, 1974

[5] Y. Kobayashi. Personal communication 1999

[6] H. Maurer. The existence of context-free languages which are inherently ambiguous of any degree. *Department of Mathematics, Research Series.* University of Calgary, 1968

[7] M. Naji. *Grad der Mehrdeutigkeit kontextfreier Grammatiken und Sprachen.* Diplomarbeit, FB Informatik,Johann–Wolfgang–Goethe–Universität, Frankfurt am Main, 1998.

[8] R.J. Parikh. Language–generating devices. *Quarterly Progress Report, No. 60, Research Laboratory of Electronics, M.I.T., pp. 199–212.* 1961

[9] K. Wich. *Kriterien für die Mehrdeutigkeit kontextfreier Grammatiken.* Diplomarbeit, FB Informatik, Johann–Wolfgang–Goethe–Universität, Frankfurt am Main, 1997.

[10] K. Wich. *Sublinear Ambiguity.* Submitted for publication.