Universal Inherence of cycle-free context-free Ambiguity Functions

Klaus Wich
E-mail: wich@informatik.uni-stuttgart.de

Institut für Informatik, Universität Stuttgart,
Breitwiesenstr. 20-22, 70565 Stuttgart.

Abstract. It is shown that the set of inherent ambiguity functions for context-free languages and the set of ambiguity functions for cycle-free context-free grammars coincide. Moreover for each census function γ of an unambiguous context-free language the least monotone function larger than or equal to γ is an inherent ambiguity function. Both results are based on a more general theorem. Informally it states that the loss of information induced by a length preserving homomorphism on an unambiguous context-free language can be turned into inherent ambiguity.

1 Introduction

A context-free (for short cf) grammar G is ambiguous if there is some word w which can be derived by G with at least two different derivation trees. Otherwise G is unambiguous. A cf language is (inherently) ambiguous if it cannot be generated by an unambiguous cf grammar. The existence of ambiguous cf languages is shown in [5]. Ambiguous cf grammars and languages can be distinguished by their degree of ambiguity, that is, the least upper bound for the number of derivation trees which a word may have. There are examples for k ambiguous languages for all $k \in \mathbb{N}$ [3]. But even languages with infinite degree of ambiguity exist [2]. They can be distinguished by the asymptotic behaviour of their ambiguity with respect to the length of the words. In [6] it has been shown that each cf grammar is either $2^{\Omega(n)}$- or $\mathcal{O}(n^k)$-ambiguous for some computable $k \in \mathbb{N}$. Languages with inherent ambiguity $2^{\Theta(n)}$ and $\Theta(n^k)$ for all $k \in \mathbb{N}$ are presented in [4]. Languages with sublinear ambiguity are presented in [7].

So far the questions whether for a given function f there is an f-ambiguous cf grammar or whether there is an f-ambiguous cf language have been studied separately. The latter question is considered to be much harder then the first one. In this paper we reduce the problem for cf languages to the corresponding problem for cycle-free cf grammars.

In Section 3 for each pair consisting of an unambiguous cf language L and a length preserving homomorphism h we show a strong connection between the loss of information induced by h on L, and the ambiguity of the constructed cf language. In Section 4 we apply the results of Section 3 and obtain:

- The least monotone function $\hat{\gamma}$ larger than or equal to the census function γ of an arbitrary unambiguous cf language is an inherent ambiguity function. (See Definition 4.1.)
- The set of ambiguity functions for cycle-free context-free grammars and the set of inherent ambiguity functions for cf languages coincide.

2 Preliminaries

Let A be a set. Then $|A|$ denotes the cardinality of A and 2^A the power set of A. For arbitrary $i, j \in \mathbb{N}$ the *interval* from i to j is $[i,j] := \{k \in \mathbb{N} \mid i \leq k \leq j\}$. We generally assume alphabets to be finite. Let A be an alphabet. Let $u := a_1 \cdots a_n \in A^*$ be a word, where $a_i \in A$ for all $i \in [1,n]$. The symbol at *position* i is $u[i] := a_i$. The *length* of u is $|u| = n$. The words over A of length at most n are denoted by $A^{\leq n} := \{w \in A^* \mid |w| \leq n\}$. The *empty word* ε is the unique word with length 0. For all $i \in [1, n+1]$ and $j \in [0,n]$ we define the *factor* of u from position i to position j as $u[i,j] := a_i \cdots a_j$. If $j < i$ then $u[i,j] = \varepsilon$. The word $u[1,j]$ is a *prefix* of u, it is a *proper* prefix if $j < n$. The word $u[i,n]$ is a *suffix* of u. A homomorphism $h : A^* \to \Gamma^*$ is *length preserving* if $|h(X)| = 1$ for all $X \in A$. The *projection* on a subalphabet $\Gamma \subseteq A$ is the homomorphism $\pi_\Gamma : A^* \to \Gamma^*$ given by $\pi_\Gamma(X) = X$ for $X \in \Gamma$ and $\pi_\Gamma(X) = \varepsilon$ for $X \in A \setminus \Gamma$. If A and Γ are two alphabets then we call a homomorphism $h : A^* \to 2^{\Gamma^*}$ a *substitution*, where the operation on 2^{Γ^*} is the concatenation of languages defined by $L_1 \cdot L_2 := \{uv \mid u \in L_1 \text{ and } v \in L_2\}$.

A context free grammar is a quadruple $G = (N, A, P, S)$ where N and A are two disjoint alphabets of *nonterminals* and *terminals*, respectively, $P \subseteq N \times (N \cup A)^*$ is a finite set of *productions*, and $S \in N$ is the *start symbol*.

The usual way to continue at this point is to introduce a derivation relation and sentential forms. A sentential form can be considered as the sequence of leaves obtained from the preorder traversal of a derivation tree. But for ambiguity considerations we need a formalism which describes derivation trees completely. The well-known left parse of a derivation tree can be used for this purpose if we restrict ourselves to trees without nonterminal leaves. The left parse of a tree can be seen as the result of a preorder traversal of a derivation tree, where the internal nodes are represented by the productions applied to them, while the leaves are omitted. Thus sentential forms and left parses are complementary parts of derivation trees. It is useful to shuffle both according to the preorder thus forming a single more general tree formalism.

The tree alphabet of a cf grammar $G = (N, A, P, S)$ is $T_G := N \cup A \cup P$. A word $\rho \in T_G^*$ is a *partial derivation tree of* G if

(1) $\rho \in N \cup A$ or
(2) if $\rho = \tau_1 (X, \alpha) \alpha \tau_2$ for some partial derivation tree $\tau_1 X \tau_2$ and $(X, \alpha) \in P$.

Note that (X, α) is a single letter of the tree alphabet. The set of G's partial derivation trees is denoted Λ_G. It is easily seen that $\Lambda_G \subseteq N \cup A \cup P T_G^*$. The root of a partial derivation tree ρ is $\uparrow_G(\rho) := \rho$ if $\rho \in N \cup A$ and $\uparrow_G(\rho) := X$ if $\rho =$

$(X,\alpha)\tau$ for some $(X,\alpha) \in P$ and $\tau \in T_G^*$. The *frontier* of ρ is $\downarrow_G(\rho) := \pi_{N\cup A}(\rho)$. If $\Uparrow_G(\rho) = S$ then the frontier of ρ is a *sentential form.* We drop the subscripts if G is clear from the context. The arrows for the root and the frontier of partial derivation trees point into the direction where they are usually displayed in a diagram. A *node* of ρ is an element of $[1,|\rho|]$. A node i is a *leaf* if $\rho[i] \in N \cup A$, it is an *internal node* if $\rho[i] \in P$. The *label* of a node i is $\rho[i]$ if i is a leaf and it is the left-hand side of the production $\rho[i]$ if i is an internal node, i.e., it is $X \in N$ if $\rho[i] \in \{X\} \times (N \cup A)^*$. The set of G's derivation trees is defined by $\Delta_G := \{\rho \in \Lambda_G \mid \Uparrow(\rho) = S \ \wedge \ \downarrow(\rho) \in A^*\}$. The context-free language *generated by* G is $L(G) = \{\downarrow(\rho) \mid \rho \in \Delta_G\}$. The grammar G is *cycle-free* if for all $\rho \in \Lambda_G$ the equation $\Uparrow(\rho) = \downarrow(\rho)$ implies $\rho \in N \cup A$, otherwise it is called cyclic. It is *reduced* if for each $X \in N$ there are $\tau_1, \tau_2 \in T_G^*$ and $p \in \{X\} \times (N \cup A)^*$ such that $\tau_1 p \tau_2 \in \Delta_G$. Finally it is ε-free if $P \subseteq N \times (N \cup A)^+$.

The set of mappings from a monoid M into a semiring S is denoted $S\langle\langle M\rangle\rangle$. An element $r \in S\langle\langle M\rangle\rangle$ is a *formal power series.* For $m \in M$ the value $r(m)$ is called *coefficient* of m. A formal power series can be represented by a formal sum $r := \sum_{m\in M} r(m)m$. For $r \in \mathbb{N}\langle\langle A^*\rangle\rangle$ we define $\hat{r} : \mathbb{N} \to \mathbb{N}$ by $\hat{r}(n) := \max\{r(w) \mid w \in A^{\leq n}\}$. The *characteristic ambiguity power series* of G is the formal power series $d_G := \sum_{w\in A^*} |\Delta_G(w)| \cdot w$ where $\Delta_G(w) := \{\rho \in \Delta_G \mid \downarrow(\rho) = w\}$ for each $w \in A^*$, i.e., the coefficients of d_G are the numbers of derivation trees for the corresponding words. The function $\hat{d}_G$ is called the *ambiguity function* of G. It maps each $n \in \mathbb{N}$ to the ambiguity of the most ambiguous word of length up to n. The grammar G is $\hat{d}_G$*-ambiguous.* We call G k*-ambiguous* for a $k \in \mathbb{N}$ if $\hat{d}_G$ is bounded by k but not by $k-1$. It is unambiguous if it is 1-ambiguous or 0-ambiguous. Note that a cf grammar is 0-ambiguous if and only if $L(G) = \emptyset$.

Let $f : \mathbb{N} \to \mathbb{N}$ be a monotone function. A cf language L is $\mathcal{O}(f)$-ambiguous if it is generated by a cf grammar G such that $\hat{d}_G \in \mathcal{O}(f)$, it is $\Omega(f)$-ambiguous if it is only generated by cf grammars G' such that $\hat{d}_{G'} \in \Omega(f)$, and it is $\Theta(f)$-ambiguous if it is $\mathcal{O}(f)$- and $\Omega(f)$-ambiguous. But the $\mathcal{O}$ and Ω notations are very rough for low ambiguities and at the same time too precise for exponential ambiguity. For example with this notation all constant degrees of ambiguity would be subsumed to $\Theta(1)$-ambiguity. On the other hand exponentially ambiguous languages are $2^{\Omega(n)}$-ambiguous but not $\Omega(2^{cn})$-ambiguous for any $c \in \mathbb{R}_+$. While the $\mathcal{O}$ and Ω notations specify the value of a function up to a constant factor for a fixed argument, in our setting it is more appropriate to specify the length of a word (argument) up to a constant factor for a fixed ambiguity (value). This leads us to the following definition:

Definition 2.1. *Let L be a cf language and $f : \mathbb{N} \to \mathbb{N}$ a function. The language L is* f-ambiguous *if*

(1) there is a cf grammar G such that $L = L(G)$ and $f = \hat{d}_G$ and
(2) for each cf grammar G' such that $L = L(G')$ there exists a $c \in \mathbb{N}$ such that $f(n) \leq \hat{d}_{G'}(c \cdot n)$ for all $n \in \mathbb{N} \setminus \{0\}$.

We implicitly identify the constant $k \in \mathbb{N}$ with the corresponding constant function. A language is unambiguous if it is 1-ambiguous or 0-ambiguous.

A function $f : \mathbb{N} \to \mathbb{N}$ is an *inherent ambiguity function* if there is a cf language L such that L is f-ambiguous. If L is f-ambiguous then L is f'-ambiguous for all monotone functions f' such that f' agrees with f for all but a finite number of arguments. Note that it is not clear whether each context-free language has an inherent ambiguity function.

It is easily seen that derivation trees cannot overlap, i.e., no non empty suffix of a partial derivation tree is a proper prefix of a partial derivation tree. Moreover each position in a partial derivation tree $\rho \in \Lambda_G$ is the beginning of a uniquely determined partial derivation tree. That is, for each $i \in [1, |\rho|]$ there is a uniquely defined $j \in [i, |\rho|]$ such that $\rho[i, j] \in \Lambda_G$. If $\rho[i, j] \in \Lambda_G$ we call $\rho[i, j]$ a *subtree* of ρ and the interval $[i, j]$ a *phrase* of ρ. Then the word $\rho[1, i-1] \cdot \Uparrow(\rho[i, j]) \cdot \rho[j+1, |\rho|] \in \Lambda_G$ is called the *remainder tree* obtained by *truncation* of the phrase $[i, j]$. Obviously we can append a partial derivation tree ρ' with root X to a leaf i of a partial derivation tree ρ, labelled with X, by a replacement of $\rho[i]$ with ρ'. Let $G = (N, A, P, S)$ be a reduced cf grammar. The terminals of a partial derivation tree $\rho \in \Lambda_G \setminus A$ can be retrieved from the remaining symbols, i.e., the restriction of the projection $\pi_{P \cup N}$ to $\Lambda_G \setminus A$ is injective. Therefore we define the *parse* of a partial derivation tree $\rho \in \Lambda_G \setminus A$, as a more compact tree representation, by $parse_G(\rho) := \pi_{P \cup N}(\rho)$. The reader familiar with the notion of left parses may note that $parse(\rho)$ and the left parse of ρ coincides for all derivation trees. But in contrast to the left parse, which is only defined for partial derivation trees of the form $(P \cup A)^*(N \cup A)^*$, our parse notion is a unique representation for all partial derivation trees, but those in A. We extend the parse notion in the natural way to sets and observe:

Lemma 2.2. *For each context-free grammar G the sets Δ_G, Λ_G, $parse_G(\Delta_G)$, and $parse_G(\Lambda_G)$ are unambiguous context-free languages.*

We take Ogden's iteration Lemma for cf grammars and for cf languages presented in [1, Lemma 2.3 and 2.5] and combine them to:

Lemma 2.3. *For each context-free grammar $G = (N, A, P, S)$ there is an integer $n \in \mathbb{N}$ such that for each $\rho \in \Delta_G$ and any choice of at least n marked positions in ρ there are $\alpha, \beta, \gamma, \delta, \eta \in T_G^*$ and a nonterminal $X \in N$ such that:*

(1) $\rho = \alpha\beta\gamma\delta\eta$.
(2) (α and β and γ) or (γ and δ and η) contain at least one marked position.
(3) $\beta\delta$ contains at most n marked positions.
(4) $\alpha\beta^i\gamma\delta^i\eta \in \Delta_G$ and $\alpha\beta^i X\delta^i\eta$, $\beta^i\gamma\delta^i$, $\beta^i X\delta^i \in \Lambda_G$ for all $i \in \mathbb{N}$.

A tuple $\vartheta = (|\alpha| + 1, |\alpha\beta|, |\alpha\beta\gamma| + 1, |\alpha\beta\gamma\delta|)$ satisfying the conditions above is called a *pumping phrase* and $\beta\gamma\delta$ the *subtree corresponding to ϑ*. Note that $\rho \in \Delta_G$ implies $\downarrow(\rho) \in L(G)$. Therefore, if we only mark leaves in ρ we obtain Ogden's iteration Lemma for cf languages. The advantage of pumping derivation trees instead of their frontiers is that they have a unique phrase structure even if the generated words are ambiguous. As we will see this additional information can be useful if we generate a sequence of derivation trees by applications of Ogden's iteration Lemma with intermediate shifts of the marked positions.

3 The Hiding Theorem

In this section it is shown how the loss of information induced by a length preserving homomorphism can be turned into inherent ambiguity.

Definition 3.1. *For the remainder of the section we define the pairwise disjoint alphabets Γ, $\{0,1\}$, and $A := \{a_1, \dots, a_k\}$. Further $L \subseteq A^*$ is an unambiguous context-free language, $h : A^* \to \Gamma^*$ a length preserving homomorphism, $p \in \mathbb{N}$ a positive integer, and $q := p! + p$.*

First we define a system of languages which has an "inherent capacity" to hide information:

Definition 3.2. *For arbitrary $j \in \mathbb{N}$ we write $[j] := 0^j 1$. For $i \in [1,k]$ we define:*

$$L_i := \{\varepsilon\} \cup \{[j_0] \cdots [j_k] \mid j_0, \dots, j_k \in \mathbb{N} \text{ and } j_0 = j_i\}.$$

All the languages defined in the previous definition are unambiguous.

Definition 3.3. *We define:*

- *The formal power series $r_{h,L} := \sum_{w \in \Gamma^*} |h^{-1}(w) \cap L| \cdot w$.*
- *The substitution $\sigma_h : A^* \to 2^{(\Gamma \cup \{0,1\})^*}$ given by*
  $\sigma_h(a_i) := \{h(a_i)\} L_i$ *for all* $i \in [1,k]$.
- *The homomorphism $\mathit{fill}_p : \Gamma^* \to (\Gamma \cup \{0,1\})^*$ defined by*
  $\mathit{fill}_p(X) := X[q]^{k+1}$ *for all* $X \in \Gamma$.
- *The homomorphism $\mathit{code}_{h,p} : A^* \to (\Gamma \cup \{0,1\})^*$ defined by*
  $\mathit{code}_{h,p}(a_i) := h(a_i)[p][q]^{i-1}[p][q]^{k-i}$ *for all* $i \in [1,k]$.

Words in $\sigma_h(L)$ can be broken into blocks *and* subblocks*. A block is an element of $\sigma_h(a_i)$ for some $i \in [1,k]$. They are numbered from left to right beginning with 1. The blocks are uniquely determined since they have the form $\Gamma\{0,1\}^*$ and do not end before the end of the word or the beginning of the next block. A subblock is a word of 0^*1 not immediately preceded by a 0-symbol. The subblocks are numbered from left to right beginning with 0.*

The main idea of this work is outlined as follows: For each $w \in \Gamma^*$ the coefficient $r_{h,L}(w)$ is the number of words in L which are mapped by h onto w. Thus it can be seen as the degree of information hiding induced by h on L. The mapping $\mathit{code}_{h,p}$ is injective since the mapped symbol is coded in the blocks of 0- and 1-symbols trailing the image under h. Now for each $u \in A^*$ both $\mathit{code}_{h,p}(u)$ and $(\mathit{fill}_p \circ h)(u)$ are elements of $\sigma_h(u)$. Thus $\sigma_h(u)$ contains at the same time words which allow to retrieve u and words which hide all information about u but $h(u)$. Let G be an arbitrary cf grammar generating $\sigma_h(L)$ and let $u \in L$. We will see that for large enough $p \in \mathbb{N}$ the set Δ_G contains a derivation tree with frontier $(\mathit{fill}_p \circ h)(u)$ obtained by pumping a derivation tree with frontier $\mathit{code}_{h,p}(u)$. Assume $w = h(u_1) = h(u_2)$ for two different words $u_1, u_2 \in L$. Then there are derivation trees ω_1 and ω_2, both having the frontier $\mathit{fill}_p(w)$ obtained by pumping up trees with frontiers $code_{h,p}(u_1)$ and $code_{h,p}(u_2)$, respectively. The main point of Theorem 3.6 is to show that these trees cannot coincide. Thus the "information hiding" which h induces from the "outside" of L is an inherent feature of $\sigma_h(L)$ "carried out" by the "internal pumping structure" of $\sigma_h(L)$.

As an immediate consequence of the definition we observe:

Lemma 3.4.

(1) $\forall u \in A^* : \sigma_h(u) \cap \Gamma^* = \{h(u)\}$ *and*
(2) $\{h(a_i)\}L_i = \sigma_h(a_i)$ *is an unambiguous cf language for all* $i \in [1, k]$.

Definition 3.5. *For each* $i \in [1, k]$ *let* $G_i = (N_i, \Gamma \cup \{0, 1\}, P_i, a_i)$ *be an unambiguous cf grammar generating* $\sigma_h(a_i)$. *Further let* $G_L = (N_L, A, P_L, S)$ *be an unambiguous cf grammar generating* L, *such that* $N_1, \ldots, N_k$, *and* N_L *are pairwise disjoint. We compose the cf grammar:*

$$G(h, L) := (N_L \cup (\cup_{i\in[1,k]} N_i), \Gamma \cup \{0, 1\}, P_L \cup (\cup_{i\in[1,k]} P_i), S).$$

Note that $L(G(h, L)) = \sigma_h(L)$.

Theorem 3.6. *The substitution* σ_h *has the following properties:*

(1) *For all* $v \in (\Gamma \cup \{0, 1\})^*$ *we have* $r_{h,L}(\pi_\Gamma(v)) = d_{G(h,L)}(\pi_\Gamma(v)) \geq d_{G(h,L)}(v)$.
(2) *For each cf grammar* G' *such that* $\sigma_h(L) = L(G')$ *there is a constant* $p \in \mathbb{N}$ *such that* $r_{h,L}(w) \leq d_{G'}(\mathit{fill}_p(w))$ *for all* $w \in \Gamma^*$.

The proof of Theorem 3.6 contains all the definitions and lemmas until Theorem 3.15.

Proof of Theorem 3.6 part *(1)***:** Each derivation tree $\rho \in \Delta_{G(h,L)}$ consists of a partial derivation tree $\rho' \in \Delta_{G_L} \subset \Lambda_{G(h,L)}$ appended with subtrees belonging to $\cup_{i\in[1,k]} \Delta_{G_i} \subset \Lambda_{G(h,L)}$. We often need to refer to the remainder tree ρ' in the sequel. Therefore we define:

Definition 3.7. *The* G_L remainder *of a derivation tree* $\rho \in \Delta_{G(h,L)}$, *denoted by* $\mathit{rem}(\rho)$, *is the uniquely defined derivation tree in* Δ_{G_L} *obtained from* ρ *by truncation of all phrases* $[j, j']$ *for which* $\rho[j, j'] \in \cup_{i\in[1,k]} \Delta_{G_i}$.

Lemma 3.8. *For all* $\rho \in \Delta_{G(h,L)}$ *the statement* $\downarrow(\rho) \in \sigma_h((\downarrow \circ\, \mathit{rem})(\rho))$ *is true.*

Proof. The expression $\sigma_h((\downarrow \circ\, \mathit{rem})(\rho))$ describes the set of words in $\sigma_h(L)$ which are frontiers of those derivation trees in $\Delta_{G(h,L)}$ having the G_L remainder $\mathit{rem}(\rho)$. Obviously ρ is such a tree. Therefore $\downarrow(\rho) \in \sigma_h((\downarrow \circ\, \mathit{rem})(\rho))$. ☐

Lemma 3.9. *For* $w \in \Gamma^*$ *and* $\rho \in \Delta_{G(h,L)}(w)$ *we have* $(\downarrow \circ\, \mathit{rem})(\rho) \in h^{-1}(w) \cap L$.

Proof. Let $w \in \Gamma^*$ and $\rho \in \Delta_{G(h,L)}(w)$. By definition $\mathit{rem}(\rho) \in \Delta_{G_L}$. Thus $(\downarrow \circ\, \mathit{rem})(\rho) \in L$. It remains to show that $(\downarrow \circ\, \mathit{rem})(\rho) \in h^{-1}(w)$. By Lemma 3.8 we obtain $w = \downarrow(\rho) \in \sigma_h((\downarrow \circ\, \mathit{rem})(\rho))$. Since $w \in \Gamma^*$ we obtain $w \in \sigma_h(u) \cap \Gamma^*$ for $u := (\downarrow \circ\, \mathit{rem})(\rho) \in A^*$. Then $w = h((\downarrow \circ\, \mathit{rem})(\rho))$ follows by Lemma 3.4. This implies $h^{-1}(w) = (h^{-1} \circ h)((\downarrow \circ\, \mathit{rem})(\rho)) \ni (\downarrow \circ\, \mathit{rem})(\rho)$. ☐

Lemma 3.10. *For arbitrary $v \in (\Gamma \cup \{0,1\})^*$ the restriction of rem to the set $\Delta_{G(h,L)}(v)$ is injective.*

Proof. Let $rem(\rho) = rem(\rho')$ for some $\rho, \rho' \in \Delta_{G(h,L)}(v)$ and let $n = |rem(\rho)|$. We can retrieve $rem(\rho)$ from ρ and ρ' by truncation of all those phrases which correspond to subtrees with roots in A. Let $\rho_1, \ldots, \rho_n \in \cup_{i \in [1,k]} \Delta_{G_i}$ and $\rho'_1, \ldots, \rho'_n \in \cup_{i \in [1,k]} \Delta_{G_i}$ be these subtrees for ρ and ρ' in a left to right order, respectively. For all $i \in [1,n]$ we observe $\Uparrow(\rho_i) = \Uparrow(\rho'_i)$. Thus ρ_i and ρ'_i both must be generated by the same grammar G_{j_i} for some $j_i \in [1,k]$. Since ρ_i and ρ'_i generates the i-th block of v we have $\Downarrow(\rho_i) = \Downarrow(\rho'_i)$ as well. But G_{j_i} is unambiguous for all $i \in [1,n]$. Hence $\rho_i = \rho'_i$ and we finally obtain $\rho = \rho'$. □

Definition 3.11. *For all $i \in [1,k]$ let $\omega_i \in \Delta_{G_i}$ be a derivation tree such that $\Uparrow(\omega_i) = a_i$ and $\Downarrow(\omega_i) = h(a_i)$. Trees with these properties must exist, since $h(a_i) \in \sigma_h(a_i) = L(G_i)$. The homomorphism append : $\Delta_{G_L} \rightarrow \Delta_{G(h,L)}$ is defined by $append(p) = p$ if $p \in P_L$ and $append(a_i) = \omega_i$ for all $i \in [1,k]$.*

Note that for all $\rho \in \Delta_{G_L}$ we have $(\Downarrow \circ\, append)(\rho) = (h \circ \Downarrow)(\rho)$. Since $\rho = rem(\rho)$ and the G_L remainder is invariant under appending trees we observe that *append* is injective.

Lemma 3.12. *For all $w \in \Gamma^*$ the restriction of $(\Downarrow \circ\, rem)$ to $\Delta_{G(h,L)}(w)$ is onto $h^{-1}(w) \cap L$.*

Proof. Let $u \in h^{-1}(w) \cap L$. Since $u \in L$ there is a $\rho \in \Delta_{G_L}$ with $u = \Downarrow(\rho)$. Since all the symbols and productions of G_L are contained in $G(h,L)$ we obtain $\rho \in \Lambda_{G(h,L)}$. Let $\rho' := append(\rho) \in \Delta_{G(h,L)}$. Obviously $\rho = rem(\rho')$. Therefore $u = \Downarrow(\rho) = \Downarrow(rem(\rho')) = (\Downarrow \circ\, rem)(\rho')$. It remains to show that $\Downarrow(\rho') = w$. Since $u \in h^{-1}(w)$ we have $h(u) = w$ and eventually $\Downarrow(\rho') = \Downarrow(append(\rho)) = (\Downarrow \circ\, append)(\rho) = (h \circ \Downarrow)(\rho) = h(\Downarrow(\rho)) = h(u) = w$. □

Lemma 3.13. *The equation $r_{h,L}(w) = d_{G(h,L)}(w)$ holds for all $w \in \Gamma^*$.*

Proof. Since $r_{h,L}(w) = |h^{-1}(w) \cap L|$ and $d_{G(h,L)}(w) = |\Delta_{G(h,L)}(w)|$ it is sufficient to show that the restriction of $(\Downarrow \circ\, rem)$ to $\Delta_{G(h,L)}(w)$ is a bijection onto $h^{-1}(w) \cap L$. Let $(\Downarrow \circ\, rem)(\rho) = (\Downarrow \circ\, rem)(\rho')$ for some $\rho, \rho' \in \Delta_{G(h,L)}(w)$. Since $rem(\rho), rem(\rho') \in \Delta_{G_L}$ and G_L is unambiguous, $rem(\rho) = rem(\rho')$ follows. By Lemma 3.10 this implies $\rho = \rho'$. Hence the restriction of $(\Downarrow \circ\, rem)$ to $\Delta_{G(h,L)}(w)$ is injective. Moreover by Lemma 3.9 it is a mapping into $h^{-1}(w) \cap L$, and by Lemma 3.12 it is a mapping onto $h^{-1}(w) \cap L$. □

Lemma 3.14. *The inequality $d_{G(h,L)}(\pi_\Gamma(v)) \geq d_{G(h,L)}(v)$ holds for all $v \in (\Gamma \cup \{0,1\})^*$.*

Proof. Since $d_{G(h,L)}(v) = |\Delta_{G(h,L)}(v)|$ and $d_{G(h,L)}(\pi_\Gamma(v)) = |\Delta_{G(h,L)}(\pi_\Gamma(v))|$, it suffices to show that the restriction of $(append \circ rem)$ to the set $\Delta_{G(h,L)}(v)$ is an injection into the set $\Delta_{G(h,L)}(\pi_\Gamma(v))$. First we show that this mapping is into $\Delta_{G(h,L)}(\pi_\Gamma(v))$. If $\Delta_{G(h,L)}(v) = \emptyset$ this is trivial, otherwise let $\rho \in \Delta_{G(h,L)}(v)$.

Since $rem(\rho) \in \Delta_{G_L}$ we obtain $(\downarrow \circ rem)(\rho) \in L \subseteq A^*$. Thus we can write $(\downarrow \circ rem)(\rho) = a_{j_1} \cdots a_{j_n}$ for some $j_1, \ldots, j_n \in [1, k]$ and some $n \in \mathbb{N}$. By Lemma 3.8 we obtain:

$$\begin{aligned} \pi_\Gamma(\downarrow(\rho)) \in \pi_\Gamma(\sigma_h((\downarrow \circ rem)(\rho))) &\subseteq \pi_\Gamma(\sigma_h(a_{j_1} \cdots a_{j_n})) \\ \subseteq \quad \pi_\Gamma(h(a_{j_1})\{0,1\}^* \cdots h(a_{j_n})\{0,1\}^*) &= \{h(a_{j_1} \cdots a_{j_n})\} \end{aligned}$$

By the use of $v = \downarrow(\rho)$ this implies:

$$\begin{aligned} \pi_\Gamma(v) = \pi_\Gamma(\downarrow(\rho)) &= h(a_{j_1} \cdots a_{j_n}) = h((\downarrow \circ rem)(\rho)) \\ = \quad (h \circ \downarrow)(rem(\rho)) = (\downarrow \circ append)(rem(\rho)) &= \downarrow((append \circ rem)(\rho)). \end{aligned}$$

Therefore $(append \circ rem)(\rho) \in \Delta_{G(h,L)}(\pi_\Gamma(v))$. It remains to show that the restriction of $(append \circ rem)$ to $\Delta_{G(h,L)}(v)$ is injective. This follows by Lemma 3.10 and the observation that *append* is injective. □

Lemma 3.13 and Lemma 3.14 immediately imply part *(1)* of Theorem 3.6. □

Proof of Theorem 3.6 part *(2)***:** Assume G' is a cf grammar such that $L(G') = \sigma_h(L)$, p is the maximum of 3 and the pumping constant of G', $q := p! + p$, and $w \in \Gamma^*$. Obviously $code_{h,p}(h^{-1}(w) \cap L) \subseteq \sigma_h(L)$. In case $h^{-1}(w) \cap L = \emptyset$ the inequality of Theorem 3.6 part *(2)* is trivially satisfied. Now assume $h^{-1}(w) \cap L \neq \emptyset$. Let $u \in h^{-1}(w) \cap L$ and $n := |w|$. Then for some $j_1, \ldots, j_n \in [1, k]$ we have:

$$code_{h,p}(u) = h(a_{j_1})[p][q]^{j_1-1}[p][q]^{k-j_1} \cdots h(a_{j_n})[p][q]^{j_n-1}[p][q]^{k-j_n}.$$

We say that an interval of a derivation tree lies *within* a subblock (block) if the corresponding nodes do not contain any leaf belonging to another subblock (block). We prove by induction that for each $i \in [0, n]$ there is a derivation tree $\rho_i \in \Delta_{G'}$ such that

$$\downarrow(\rho_i) = (fill_p \circ h)(a_{j_1} \cdots a_{j_i}) \cdot code_{h,p}(a_{j_{i+1}} \cdots a_{j_n})$$

and for each $m \in [1, i]$ the derivation tree ρ_i has a pumping phrase allowing to pump the same number of 0-symbols into the 0-th subblock and the j_m-th subblock of block m jointly. For $i = 0$ we only have to show that $code_{h,p}(u) = \downarrow(\rho_0)$ for some $\rho_0 \in \Delta_{G'}$. This follows by $code_{h,p}(u) \in \sigma_h(L) = L(G')$. Assume the statement is true for $i - 1$. Then there is a derivation tree $\rho_{i-1} \in \Delta_{G'}$ with the required phrase structure and the sentential form:

$$\downarrow(\rho_{i-1}) = (fill_p \circ h)(a_{j_1} \cdots a_{j_{i-1}}) \cdot h(a_{j_i})[\underline{p}][q]^{j_i-1}[p][q]^{k-j_i} \cdot code_{h,p}(a_{j_{i+1}} \cdots a_{j_n}).$$

The 0-th subblock of the i-th block in $\downarrow(\rho_{i-1})$ is underlined to indicate that the leaves of ρ_{i-1} forming the 0-symbols of this subblock are marked. According to Ogden's Lemma 2.3 the tree $\rho_{i-1} = \alpha\beta\gamma\delta\eta$ for some $\alpha, \beta, \gamma, \delta, \eta \in T_{G'}^*$ such that $\alpha\beta^l\gamma\delta^l\eta \in \Delta_{G'}$ for each $l \in \mathbb{N}$. Moreover $\beta\delta$ must contain at least one marked position and at least one of the intervals $\tau_\beta := [|\alpha| + 1, |\alpha\beta|]$ and $\tau_\delta := [|\alpha\beta\gamma| + 1, |\alpha\beta\gamma\delta|]$ lies within the 0-th subblock of block i. Let $\tau := \tau_\delta$ if τ_β has this property and $\tau := \tau_\beta$ otherwise.

By the choice of p and q the insertion of at most p many 0-symbols into a subblock $[p]$ yields a subblock shorter than $[q]$. We will implicitly apply this argument in the sequel.

Assume τ is not within the i-th block, i.e., it is outside or it overlaps with block i and some neighbouring blocks. Let $i' = i + c$ where c is the number of Γ symbols in β if $\tau = \tau_\beta$ and $i' := i$ otherwise. Then block i' of $\downarrow(\alpha\beta^2\gamma\delta^2\eta)$ equals block i of $\downarrow(\alpha\beta\gamma\delta\eta)$, except for a proper insertion of at most p many 0-symbols in the 0-th subblock of block i'. Therefore within block i' the 0-th subblock does not agree with any other subblock.

Now assume τ lies within block i. Then it cannot contain a 1-symbol because otherwise the i-th block of $\downarrow(\alpha\beta^2\gamma\delta^2\eta)$ would contain more than $k+1$ subblocks. Hence each of τ_β and τ_δ lie within one subblock of the i-th block. We can easily verify that $\downarrow(\alpha\beta^2\gamma\delta^2\eta)$ does not contain more than $2p$ occurrences of 0-symbols in the 0-th subblock of block i in this case. This implies that τ_β lies within the 0-th subblock and τ_δ within the j_i-th subblock of block i and $1 \leq |\downarrow(\beta)| = |\downarrow(\delta)| \leq p$. Thus for $l = p! \cdot |\downarrow(\beta)|^{-1} + 1$ the derivation tree $\rho_i := \alpha\beta^l\gamma\delta^l\eta$ has the property $\downarrow(\rho_i) = (\mathit{fill}_p \circ h)(a_{j_1} \cdots a_{j_i}) \cdot \mathit{code}_{h,p}(a_{j_i+1} \cdots a_{j_k})$. Now ρ_i contains a pumping phrase allowing to pump the same number of 0-symbols into the 0-th subblock and into the j_i-th subblock of block i jointly. Moreover the pumping phrases of ρ_i to the left of block i are the same as in ρ_{i-1}, which completes the induction.

Eventually for $i = n$ we obtain a derivation tree ρ_n with the frontier $\downarrow(\rho_n) = (\mathit{fill}_p \circ h)(u) = \mathit{fill}_p(w)$ and the claimed phrase structure starting from an arbitrary word of $\mathit{code}_{h,p}(h^{-1}(w) \cap L)$. It remains to show that two trees obtained in this way beginning with different words in $h^{-1}(w) \cap L$ cannot coincide. Let $u_1, u_2 \in h^{-1}(w) \cap L$ be two different words and let ω_1 and ω_2 be the corresponding derivation trees obtained by the pumping sequence described above. Then ω_1 and ω_2 both generate $\mathit{fill}_p(w)$. Assume $\omega_1 = \omega_2$. Since h is length preserving we observe $|u_1| = |u_2|$. Therefore u_1 and u_2 differ in at least one position $i \in [1, |u_1|]$. Then $a_j = u_1[i] \neq u_2[i] = a_{j'}$ for some $j, j' \in [1, k]$. W.l.o.g. we assume $j > j'$. Then ω_1 contains a pumping phrase ϑ_1 allowing to pump the 0-th subblock and the j-th subblock of block i jointly and it contains a pumping phrase ϑ_2 allowing to pump the 0-th subblock and the j'-th subblock of block i jointly. Pumping once ω_1 corresponding to ϑ_1 we obtain a derivation tree ω' with a word $\beta \in (P \cup \{0\})^*$ inserted to the left of ϑ_2. Since β does only contain leaves labelled with 0-symbols ϑ_2 is shifted to the right, but remains within the same subblock as in ω_1. Thus in ω' it is still possible to pump the 0-th subblock and the j'-th subblock of block i jointly. This pumping yields a derivation tree ω'' for which the 0-th subblock of block i does no longer agree with any other subblock of block i, which is a contradiction. Hence $\omega_1 \neq \omega_2$. This implies that $\mathit{fill}_p(w)$ can be generated by at least $|\mathit{code}_{h,p}(h^{-1}(w) \cap L)|$ many different derivation trees. Moreover since $\mathit{code}_{h,p}$ is injective we finally obtain $d_{G'}(\mathit{fill}_p(w)) \geq |\mathit{code}_{h,p}(h^{-1}(w) \cap L)| = |h^{-1}(w) \cap L| = r_{h,L}(w)$. □

Theorem 3.15. *The context-free language $\sigma_h(L)$ is $\hat{r}_{h,L}$-ambiguous.*

Proof. Recall that $L(G(h, L)) = \sigma_h(L)$. Now Theorem 3.6 *(1)* implies:

$$
\begin{aligned}
&\max\{d_{G(h,L)}(v) \mid v \in (\Gamma \cup \{0,1\})^{\leq n}\}\\
\geq\ &\max\{d_{G(h,L)}(w) \mid w \in \Gamma^{\leq n}\}\\
=\ &\max\{d_{G(h,L)}(\pi_\Gamma(v)) \mid v \in (\Gamma \cup \{0,1\})^{\leq n}\}\\
\overset{3.6}{\geq}\ &\max\{d_{G(h,L)}(v) \mid v \in (\Gamma \cup \{0,1\})^{\leq n}\}
\end{aligned}
$$

Hence all the expressions above are equal and again by Theorem 3.6 *(1)* we obtain:

$$
\begin{aligned}
\hat{r}_{h,L}(n) &= \max\{r_{h,L}(w) \mid w \in \Gamma^{\leq n}\} \overset{3.6}{=} \max\{d_{G(h,L)}(w) \mid w \in \Gamma^{\leq n}\}\\
&= \max\{d_{G(h,L)}(v) \mid v \in (\Gamma \cup \{0,1\})^{\leq n}\} = \hat{d}_{G(h,L)}(n)
\end{aligned}
$$

Thus $G(h,L)$ is appropriate to satisfy property *(1)* of Definition 2.1. By Theorem 3.6 *(2)* we obtain that for each cf grammar G' such that $L(G') = \sigma_h(L)$ there is a $p \in \mathbb{N}$ such that for all words $w \in \Gamma^*$ we have $r_{h,L}(w) \leq d_{G'}(\mathit{fill}_p(w))$. This implies $\hat{r}_{h,L}(|w|) \leq \hat{d}_{G'}(|\mathit{fill}_p(w)|) = \hat{d}_{G'}(c \cdot |w|)$ where $c = 1 + (k+1)(p! + p + 1)$. Thus $\sigma_h(L)$ and $\hat{r}_{h,L}$ satisfy property *(2)* of Definition 2.1. □

4 Applications

4.1 Census Functions

Definition 4.1. *Let $L \subseteq A^*$ be a formal language. The* census function $\gamma_L : \mathbb{N} \to \mathbb{N}$ *is defined by $\gamma_L(n) := |A^n \cap L|$, and the function $\hat{\gamma}_L : \mathbb{N} \to \mathbb{N}$ is defined by $\hat{\gamma}_L(n) := \max\{\gamma_L(i) \mid i \leq n\}$. The homomorphism $\mathit{hide} : A^* \to \{\$\}$ is defined by $\mathit{hide}(X) := \$$ for all $X \in A$.*

Theorem 4.2. *Let $L \subseteq A^*$ be an unambiguous context-free language. Then*

$$\sigma_{\mathit{hide}}(L) \textit{ is } \hat{\gamma}_L\textit{-ambiguous.}$$

Proof. By Theorem 3.15 the language $\sigma_{\mathit{hide}}(L)$ is $\hat{r}_{\mathit{hide},L}$-ambiguous. We get $\hat{r}_{\mathit{hide},L}(n) = \max\{r_{\mathit{hide},L}(w) \mid w \in A^{\leq n}\} = \max\{|\mathit{hide}^{-1}(w) \cap L| \mid w \in A^{\leq n}\} = \max\left\{|A^{|w|} \cap L| \,\middle|\, w \in A^{\leq n}\right\} = \max\left\{|A^j \cap L| \,\middle|\, j \leq n\right\} = \max\left\{\gamma_L(j) \,\middle|\, j \leq n\right\} = \hat{\gamma}_L(n)$. □

Corollary 4.3. *There is an unambiguous context-free language L such that L^+ is exponentially ambiguous and L^k is $\Theta(n^{k-1})$-ambiguous for each $k \in \mathbb{N} \setminus \{0\}$.*

Proof. Let $\{a,b\}$ be an alphabet. We observe that $\hat{\gamma}_{(a+b)^*b}(n) = \lfloor 2^{n-1} \rfloor$ and $\hat{\gamma}_{(a^*b)^k}(n) = \binom{n-1}{k-1}$ for each $k \in \mathbb{N} \setminus \{0\}$. Using Theorem 4.2 we obtain that $\sigma_{\mathit{hide}}((a+b)^*b)$ is $\lfloor 2^{n-1} \rfloor$-ambiguous and $\sigma_{\mathit{hide}}((a^*b)^k)$ is $\binom{n-1}{k-1}$-ambiguous. Thus $\sigma_{\mathit{hide}}((a^*b)^1)$ is unambiguous. Finally since σ_{hide} is a homomorphism we immediately get $\sigma_{\mathit{hide}}((a+b)^*b) = \sigma_{\mathit{hide}}((a^*b)^+) = (\sigma_{\mathit{hide}}(a^*b))^+$ and $\sigma_{\mathit{hide}}((a^*b)^k) = (\sigma_{\mathit{hide}}(a^*b))^k$. Thus $L := \sigma_{\mathit{hide}}(a^*b)$ is a language with the required properties. □

4.2 Cycle-free context-free Grammars

In this part our aim is to find out when the ambiguity function of a cf grammar is inherent for some cf language.

Definition 4.4. *A cf grammar $G = (N, A, P, S)$ is in* Greibach-normal-form *if* $P \subseteq N \times AN^*$

For cf grammars in Greibach-normal-form the parse of each derivation tree has the same length as its frontier. Moreover the i-th symbol of the frontier is uniquely determined by the i-th symbol of the parse. This implies the following lemma:

Lemma 4.5. *Let $G = (N, A, P, S)$ be a cf grammar in Greibach-normal-form and $h_G : P^* \to A^*$ the length preserving homomorphism defined by $h_G(p) := X_p$ for each $p \in P$ where X_p is the terminal at the beginning of p's right-hand side. Then $\downarrow(\rho) = h_G(parse(\rho))$ for all $\rho \in \Delta_G$.*

Theorem 4.6. *Let $G = (N, A, P, S)$ be a context-free grammar in Greibach-normal-form, and h_G defined as in Lemma 4.5. Then the context-free language $\sigma_{h_G}(parse_G(\Delta_G))$ is $\hat{d}_G$-ambiguous.*

Proof. Let $L := parse_G(\Delta_G)$ and $h := h_G$.

$$\begin{aligned}
\hat{r}_{h,L}(n) &= \max\left\{ |h^{-1}(w) \cap L| \;\middle|\; w \in A^{\leq n} \right\} \\
&= \max\left\{ \left|\{\rho \in \Delta_G \mid \downarrow(\rho) = w\}\right| \;\middle|\; w \in A^{\leq n} \right\} \\
&= \max\left\{ d_G(w) \;\middle|\; w \in A^{\leq n} \right\} \qquad\qquad = \hat{d}_G(n).
\end{aligned}$$

By Lemma 2.2 the cf language $parse_G(\Delta_G)$ is unambiguous. Moreover h_G is length preserving. Thus the claim follows by Theorem 3.15 □

Let us consider the ambiguity function of an arbitrary cycle-free cf grammar $G = (N, A, P, S)$. If $A = \emptyset$ then $L(G) \subseteq \{\varepsilon\}$ and G has a constantly bounded ambiguity function, which is an inherent ambiguity function. If $A \neq \emptyset$ we can find an ε-free cycle-free cf grammar G' such that $L(G') = (L(G) \cup \{a\}) \setminus \{\varepsilon\}$ for some $a \in A$, such that $\hat{d}_{G'}(n) = \hat{d}_G(n)$ for all $n \in \mathbb{N} \setminus \{0\}$. It can be shown that G' can be transformed into an equivalent cf grammar G'' in Greibach-normal-form which has the same characteristic ambiguity power series as G'. Then by Theorem 4.6 the language $\sigma_{h_{G''}}(parse_{G''}(\Delta_{G''}))$ is $\hat{d}_{G'}$-ambiguous. By inspection of Definition 2.1 we can easily verify that this also implies $\hat{d}_G$-ambiguity for $\sigma_{h_{G''}}(parse_{G''}(\Delta_{G''}))$. Therefore $\hat{d}_G$ is an inherent ambiguity function.

Now we consider an arbitrary inherent ambiguity function f. Then there is an f-ambiguous cf language L. Since symbols which cannot occur in any derivation tree do not contribute to the ambiguity function there is a reduced cf grammar G such that $\hat{d}_G = f$ and $L = L(G)$. We assume G is cyclic. Then there is a word w which has infinitely many derivation trees. Thus $f(|w|) = \infty$. But each cf language can be generated by a cycle-free cf grammar. Since they do not have an infinite number of derivation trees for any word f cannot be inherent for L, which is a contradiction. Hence G is a cycle-free cf grammar. Therefore f is the ambiguity function of a cycle-free cf grammar. This finally implies:

Theorem 4.7. *The set of ambiguity functions for cycle-free context-free grammars and the set of inherent ambiguity functions coincide.*

5 Conclusion

It has been shown that the loss of information induced by a length preserving homomorphism can be turned into inherent ambiguity. The result has been used to prove that:

- The least monotone function $\hat{\gamma}$ larger than or equal to the census function of an arbitrary unambiguous context-free language is an inherent ambiguity function.
- The set of ambiguity functions for cycle-free context-free grammars and the set of inherent ambiguity functions coincide.

The latter result is particularly useful for future research. As was mentioned in the introduction some examples of languages with sublinear ambiguity have recently been discovered. But a characterisation of the obtainable ambiguities is still missing. The author conjectures that context-free languages with infinite sublogarithmic ambiguity can be found. To prove that a given function f is an inherent ambiguity function, with the result of this paper, it is sufficient to find an f-ambiguous cycle-free context-free grammar. Finally we can raise the question whether each context-free language is f-ambiguous for some function f.

Acknowledgements: I would like to thank Friedrich Otto for proofreading and valuable discussions.

References

1. J. Berstel. *Transductions and context-free languages.* Teubner Studienbücher, Stuttgart, 1979.
2. J. Crestin. Un langage non ambigu dont le carré est d'ambiguité non bornée. In M. Nivat, editor, *Automata, Languages and Programming*, pages 377–390. Amsterdam, North-Holland, 1973.
3. H. Maurer. The existence of context-free languages which are inherently ambiguous of any degree. Research series, Department of Mathematics, University of Calgary, 1968.
4. M. Naji. Grad der Mehrdeutigkeit kontextfreier Grammatiken und Sprachen, 1998. Diplomarbeit, FB Informatik, Johann–Wolfgang–Goethe–Universität Frankfurt/M.
5. R. J. Parikh. Language–generating devices. In *Quarterly Progress Report*, volume 60, pages 199–212. Research Laboratory of Electronics, M.I.T, 1961.
6. K. Wich. Exponential ambiguity of context-free grammars. In G. Rozenberg and W. Thomas, editors, *Proceedings of the 4th International Conference on Developments in Language Theory, July 1999*, pages 125–138. World Scientific, Singapore, 2000.
7. K. Wich. Sublinear ambiguity. In M. Nielsen and B. Rovan, editors, *Proceedings of the MFCS 2000*, number 1893 in Lecture Notes in Computer Science, pages 690–698, Berlin-Heidelberg-New York, 2000. Springer.