



Barbara Dziemidowicz-Gryz

University of Warmia and Mazury

ON THE LEARNABILITY OF CLASSES OF MINIMAL GRAMMARS WITH RESPECT TO SOME LINEAR PREORDERING RELATIONS

Abstract. In this paper we prove the learnability of the classes of minimal grammars with respect to some linear preorderings on the set of categorial grammars. We give some examples of the linear preordering relations on the set of categorial grammars satisfying necessary conditions.

Keywords: categorial grammar, unification, learning function, learning algorithm.

1. Introduction

Buszkowski and Penn presented in (Buszkowski & Penn, 1990) a unification discovery procedure (learning algorithm) for finding categorial grammars compatible with a given sample of expressions of a language. Kanazawa in (Kanazawa, 1998) studied Gold-style learning functions based on these algorithms.

In this paper we present a learning function for classes of minimal grammars with respect to some linear preorderings on the set of categorial grammars. These results are related to Kanazawa's open problem of whether the class of minimal grammars is learnable.

In sections 2, 3 and 4 we present basic definitions and fundamental theorems of learnability theory for categorial grammars. In section 5 we define some linear preordering relations on a set of categorial grammars. We prove the learnability of classes of minimal grammars with respect to these preordering relations.

2. Learnability Theory

We present basic notions of learnability theory, after (Gold, 1967), (Kanazawa, 1996), (Kanazawa, 1998), (Osheron, de Jongh, Martin & Weinstein, 1997).

A triple (Ω, S, L) such that Ω is a class of grammars, S is a class of sentences (a certain recursive subset of Σ^* for some fixed finite alphabet Σ) and L is a function that maps elements of Ω to subsets of S is called a *grammar system*. We assume that the universal membership problem $s \in L(G)$, for $s \in S$, $G \in \Omega$, is decidable.

Let (Ω, S, L) be a grammar system. A *learning function* is a partial function φ that maps non-empty finite sequences of sentences to grammars ($\varphi : \bigcup_{k \geq 1} S^k \rightarrow \Omega$). Usually, we are interested in learning functions that are effectively computable. The term *learning algorithm* means an algorithm that computes a learning function.

Let $\langle s_i \rangle_{i \in N} = \langle s_0, s_1, \dots \rangle$ be an infinite sequence of sentences from S . Given $\langle s_i \rangle_{i \in N}$, a learning function φ determines a grammar $G_i = \varphi(\langle s_0, \dots, s_i \rangle)$ for each $i \in N$ such that φ is defined on $\langle s_0, \dots, s_i \rangle$. We say that φ *converges to G* on $\langle s_i \rangle_{i \in N}$ if $G_i = \varphi(\langle s_0, \dots, s_i \rangle)$ is defined and equal to G for all but finitely many $i \in N$; equivalently, if there exists an $n \in N$ such that for all $i \geq n$, G_i is defined and equal to G .

Let \mathcal{G} be a class of grammars in Ω . \mathcal{G} determines the corresponding class of languages, $L(\mathcal{G}) = \{L(G) \mid G \in \mathcal{G}\}$.

Let a grammar system (Ω, S, L) be given, and let $\mathcal{G} \subseteq \Omega$. A learning function φ is said *to learn \mathcal{G}* if the following condition holds: for every language L in $L(\mathcal{G})$, for every infinite sequence $\langle s_i \rangle_{i \in N}$ that enumerates the elements of L ($\{s_i \mid i \in N\} = L$), there exists some G in \mathcal{G} such that $L(G) = L$ and φ converges to G on $\langle s_i \rangle_{i \in N}$.

A class \mathcal{G} of grammars is said to be *learnable* if there is a computable learning function that learns \mathcal{G} .

A learning function φ is said to be *conservative* if, for any finite sequence $\langle s_0, \dots, s_i \rangle$ of sentences and for any sentence s_{i+1} , whenever $\varphi(\langle s_0, \dots, s_i \rangle)$ is defined and $s_{i+1} \in L(\varphi(\langle s_0, \dots, s_i \rangle))$, $\varphi(\langle s_0, \dots, s_i, s_{i+1} \rangle)$ is also defined and $\varphi(\langle s_0, \dots, s_i, s_{i+1} \rangle) = \varphi(\langle s_0, \dots, s_i \rangle)$.

A class \mathcal{L} of languages is said *to have infinite elasticity* if there exists an infinite sequence $\langle s_n \rangle_{n \in N}$ of sentences and an infinite sequence $\langle L_n \rangle_{n \in N}$ of languages in \mathcal{L} such that, for all $n \in N$, $s_n \notin L_n$ and $\{s_0, \dots, s_n\} \subseteq L_{n+1}$.

A class \mathcal{L} of languages is said *to have finite elasticity* if it does not have infinite elasticity.

Lemma 2.1. (Kapur & Shyam, 1991)

Let \mathcal{L} be a language class with finite elasticity. Then for each language $L \in \mathcal{L}$, there is a finite set $D_L \subseteq L$ such that L is the smallest element of $\{L' \in \mathcal{L} \mid D_L \subseteq L'\}$.

3. Unification

Let L be a first-order language. By $TER(L)$ we denote the set of all terms of L . VAR denotes the set of individual variables, and $CON(L)$ denotes the set of all individual constants of L . A *substitution* is a mapping $\alpha : VAR \rightarrow TER(L)$. Each substitution α is uniquely extended to a homomorphism $\alpha : TER(L) \rightarrow TER(L)$ by setting: (i) $\alpha(a) = a$, for $a \in CON(L)$, (ii) $\alpha(f(t_1, \dots, t_n)) = f(\alpha(t_1), \dots, \alpha(t_n))$. $\alpha \circ \beta$ denotes the composition of α and β ; $(\alpha \circ \beta)(t) = \alpha(\beta(t))$, for any term t . Terms s, t are called *variants*, if there are substitutions α, β such that $\alpha(s) = t$ and $\beta(t) = s$ (then, s is equal to t up to renaming variables).

The set of all substitutions $\alpha : VAR \rightarrow TER(L)$ we denote by SUB . A set $X \subseteq SUB$ is said to be *hereditary*, if, for all $\alpha, \beta \in SUB$ if $\alpha \circ \beta \in X$, then $\beta \in X$.

A set $T \subseteq TER(L)$ is said to be *unifiable*, if there is a substitution α such that $\alpha(s) = \alpha(t)$, for all $s, t \in T$; α is called a *unifier of T* . A substitution α is called a *most general unifier (m.g.u.) of T* , if α is a unifier of T and, for every unifier β of T , there is a substitution γ such that $\beta = \gamma \circ \alpha$.

The well-known *unification algorithm* (Doets, 1994), (Lloyd, 1987) takes an arbitrary finite set of terms as an input and returns an m.g.u. of T , if T is unifiable; otherwise, the algorithm replies negatively.

For our purposes, it is more expedient to consider unifiers of relations. Let R be a finite binary relation on $TER(L)$. A substitution α is called a *unifier of R* , if $\alpha(s) = \alpha(t)$, for any pair $(s, t) \in R$.

The relation R is said to be *unifiable*, if there exists a unifier of R . The notion of an m.g.u. of R is defined as above. It is easy to adopt the unification algorithm to binary relations as inputs. Let $R = \{(s_1, t_1), \dots, (s_n, t_n)\}$. Take a new n-ary function symbol F and consider the set T consisting of two terms: $F(s_1, \dots, s_n)$ and $F(t_1, \dots, t_n)$. Then, unifiers of R are precisely unifiers of T (not using F), and consequently, the unification algorithm applied to T provides an m.g.u. of R , if R is unifiable, and replies negatively, otherwise.

Let \mathfrak{S} be a finite family of finite sets of terms. A substitution α is called a *unifier of \mathfrak{S}* , if α is a unifier of every set $T \in \mathfrak{S}$ (but not necessarily of the

join of \mathfrak{S}). For a family \mathfrak{S} , we define a relation $R(\mathfrak{S})$ as follows: $(s, t) \in R(\mathfrak{S})$ iff, for some $T \in \mathfrak{S}$, both $s \in T$ and $t \in T$. Then, unifiers of \mathfrak{S} are precisely unifiers of $R(\mathfrak{S})$, and similarly for other notions, defined in terms of unifiers.

For a substitution α , the relation $Ker(\alpha)$ (*the kernel of α*) is defined as follows: for $s, t \in TER(L)$, $(s, t) \in Ker(\alpha)$ iff $\alpha(s) = \alpha(t)$. For a relation R between terms, we set $Ker_R(\alpha) = R \cap Ker(\alpha)$. Clearly, α is a unifier of $Ker_R(\alpha)$ and $Ker_R(\alpha) \subseteq Ker_R(\beta \circ \alpha)$, for all R , α and β .

4. Classical Categorical Grammars

The notion of a categorical grammar has been introduced by Ajdukiewicz in (Ajdukiewicz, 1935). In a categorical grammar, each symbol in the alphabet is associated with a finite number of types. Pr denotes the set of primitive types. One member t of Pr is singled out as *the distinguished type*. The members of Pr other than t are called *variables*. The set of variables is denoted VAR . *Types* are constructed from primitive types by two type-forming operators, $/$ and \backslash . Tp (the set of all types) is defined as the smallest set satisfying the following conditions:

- (T1) $Pr \subseteq Tp$,
- (T2) if $A \in Tp$ and $B \in Tp$, then $A \backslash B \in Tp$,
- (T3) if $A \in Tp$ and $B \in Tp$, then $A/B \in Tp$.

One assumes the following reduction rules:

$$A, A \backslash B \rightarrow B$$

$$A/B, B \rightarrow A.$$

A *classical categorical grammar* over Σ is a finite relation G between Σ and Tp . For $c \in \Sigma$ and $A \in Tp$, if $(c, A) \in G$, then we say that G assigns A to c , and write $G : c \rightarrow A$. We say that G assigns type A to string $c_1 \dots c_n$ if there exist $(c_i, A_i) \in G, i = 1, \dots, n$, such that A_1, \dots, A_n reduces to A by a successive application of the above rules.

Types can be identified with terms of a first-order language with function symbols $\backslash, /$ and the constant t .

Let σ be a substitution. $\sigma(G)$ denotes a grammar obtained by applying σ to the type assignments of G , that is: $\sigma(G) = \{(c, \sigma(A)) : (c, A) \in G\}$.

We define the set Σ^F (of functor-argument structures over alphabet Σ) as follows:

- (i) $c \in \Sigma^F$ for all $c \in \Sigma$,
- (ii) if $X_1, X_2 \in \Sigma^F$, then also $(X_1, X_2)_i \in \Sigma^F, i = 1, 2$.

X_i is called *the functor* and $X_j, j \neq i$ is called *an argument*.

Let $v \in \Sigma$. By $I_G(v)$ we denote the set $\{A \in Tp : G : v \rightarrow A\}$. For $T_1, T_2 \subseteq Tp$, we set: $f_1(T_1, T_2) = \{B \in Tp : \exists A(B/A \in T_1 \wedge A \in T_2)\}$, $f_2(T_1, T_2) = \{B \in Tp : \exists A(A \in T_1 \wedge A \setminus B \in T_2)\}$. Let T_G be a mapping from Σ^F into the subsets of Tp defined as follows: $T_G(v) = I_G(v)$, for $v \in \Sigma$, $T_G((X, Y)_i) = f_i(T_G(X), T_G(Y))$. We define *the structure language of G* by $FL(G) = \{X \in \Sigma^F : t \in T_G(X)\}$.

Let D be a finite set of functor-argument structures. A grammar $GF(D)$ (the general form determined by D) is constructed in the following way:

- (i) we assign the distinguished type t to all functor-argument structures $X \in D$,
- (ii) we assign distinct variables to all argument-substructures of all $X \in D$,
- (iii) we assign types to all functor-substructures according to the following rules:
 - if $Y \rightarrow A$, $(X, Y)_1 \rightarrow B$, then $X \rightarrow B/A$,
 - if $X \rightarrow A$, $(X, Y)_2 \rightarrow B$, then $Y \rightarrow A \setminus B$.
- (iv) we collect the types assigned to all symbols from the alphabet into the grammar $GF(D)$.

We define: $dom(G) = \{c \in \Sigma : \exists A \in Tp(G : c \rightarrow A)\}$.

Let \sqsubseteq be a binary relation on grammars such that $G_1 \sqsubseteq G_2$ if and only if there exists a substitution σ with the following properties:

- (i) for all $c \in dom(G)$, if $G_1 : c \rightarrow A$, $G_1 : c \rightarrow B$ and $A \neq B$, then $\sigma(A) \neq \sigma(B)$,
- (ii) $\sigma(G_1) \subseteq G_2$.

A grammar G is said to be *in reduced form* if there is no G' such that $G' \sqsubset G$ and $FL(G) = FL(G')$. $G' \sqsubset G$ means: $G' \sqsubseteq G$ and $G \not\sqsubseteq G'$.

Proposition 4.1 (Kanazawa, 1998)

For every grammar G , one can effectively find a grammar G' in reduced form such that $G' \sqsubseteq G$ and $FL(G) = FL(G')$.

By $range(G)$ we denote the set $\{A : (c, A) \in G \text{ for some } c \in \Sigma\}$.

Let $Tp(G) = \{A : A \text{ is a subtype of some } B \in range(G)\}$. A type $A \in Tp(G)$ is useless if and only if there is no string $c_1 \dots c_n$ such that there exist $(c_i, A_i) \in G, i = 1, \dots, n$, such that A is a subtype of some A_i and A_1, \dots, A_n reduces to t .

Proposition 4.2 (Kanazawa, 1998)

If a grammar G is in reduced form, then G has no useless type.

5. Classes of minimal grammars with respect to some linear preordering relations

We prove the learnability of the classes of minimal grammars with respect to some linear preorderings on the set of categorial grammars.

Recall that a *preordering relation on a set X* is a reflexive and transitive relation on X . If \leq is a preordering relation on X , then we write $a \sim b$ iff $a \leq b$ and $b \leq a$; and $a < b$ if $a \leq b$ and $b \not\leq a$, for $a, b \in X$.

Let $X \subseteq SUB$ be hereditary. Let R be a finite relation between types. A preordering relation \leq on X is said *to be sound for R* , if it satisfies the following conditions:

(C1) if $Ker_R(\alpha) \subset Ker_R(\beta)$, then $\beta < \alpha$,

(C2) if $Ker_R(\alpha) = Ker_R(\beta)$, then $\alpha \sim \beta$,

for all $\alpha, \beta \in X$.

A *linear preordering on a set X* is a preordering relation on X satisfying the following condition: $a \leq b$ or $b \leq a$, for all $a, b \in X$.

Let D be a set of functor-argument structures. Then $\mathcal{A}(D) = \{\{A : GF(D) : c \rightarrow A\} : c \in dom(GF(D))\}$. We set $R_D = R_{\mathcal{A}(D)}$.

Let \preceq be a linear preordering on a set of categorial grammars satisfying the following conditions:

(G1) $G_1 \sqsubseteq G_2 \Rightarrow G_1 \preceq G_2$,

(G2) the preordering \leq on a set of substitutions defined by: $\alpha \leq \beta$ iff $\alpha(GF(D)) \preceq \beta(GF(D))$ is sound for R_D , for all finite sets of functor-argument structures D ,

(G3) the class of structure languages $\{FL(G') : G' \preceq G\}$ has finite elasticity for any categorial grammar G .

We define: $G_1 \simeq G_2$ iff $G_1 \preceq G_2$ and $G_2 \preceq G_1$; $G_1 \prec G_2$ iff $G_1 \preceq G_2$ and $G_2 \not\preceq G_1$.

Let $L \subseteq \Sigma^F$. A grammar G is said *to be minimal with respect to (L, \preceq)* if $L \subseteq FL(G)$ and there is no grammar G' such that $G' \prec G$ and $L \subseteq FL(G')$.

A grammar G is said *to be minimal with respect to \preceq* iff G is minimal with respect to $(FL(G), \preceq)$. The class of minimal grammars with respect to \preceq is denoted by $\mathcal{G}_{min, \preceq}$. By $\mathcal{FL}_{min, \preceq}$ we denote the set $\{FL(G) : G \in \mathcal{G}_{min, \preceq}\}$.

A substitution α is called *an optimal unifier of a relation R* , if it satisfies the following conditions (Buszkowski & Penn, 1990), (Buszkowski & Dziemidowicz, 2000):

(OU.1) for every substitution β , $Ker_R(\alpha) = Ker_R(\beta \circ \alpha)$,

(OU.2) α is an m.g.u. of $Ker_R(\alpha)$.

Let R be a finite relation between types. We define:

$$OU(R) = \{\sigma : \sigma \text{ is optimal unifier of } R\}.$$

We provide an Algorithm OG based on Buszkowski's algorithm (Buszkowski & Penn, 1990):

Algorithm OG

Input: a finite set D of functor-argument structures

Output: a finite set of grammars G such that $D \subseteq FL(G)$.

1. Construct $GF(D)$.
2. Compute $OU(R_D)$.
3. Let $OG(D) = \{\sigma(GF(D)) : \sigma \in OU(R_D)\}$.

A grammar G is said to be optimal if the following two conditions hold:

- (i) G has no useless types,
- (ii) if $G : c \rightarrow A$ and $G : c \rightarrow B$ and $A \neq B$, then $R = \{(A, B)\}$ is not unifiable.

The class of optimal grammars is denoted by $\mathcal{G}_{optimal}$.

Lemma 5.1. (Kanazawa, 1998)

$$\{G \in \mathcal{G}_{optimal} : G \text{ is in reduced form}\} \subset \bigcup \text{range}(OG).$$

Lemma 5.2. (Kanazawa, 1998)

If $G \in OG(D)$ and $T \in FL(G)$, then $G \in OG(D \cup \{T\})$.

A categorial grammar assigning at most k types to each symbol in the alphabet is called a *k-valued grammar*. The set of k -valued grammars is denoted by \mathcal{G}_k .

Let $\mathcal{A} = \{T_1, \dots, T_n\}$ be a family of sets of types. If $\mathcal{P}_i = \{P_{i,1}, \dots, P_{i,l_i}\}$ is a partition of T_i for $1 \leq i \leq n$, then the family $\mathcal{P} = \bigcup \{\mathcal{P}_i : 1 \leq i \leq n\}$ is called a *partition of \mathcal{A}* . \mathcal{P} is called a *k-partition of \mathcal{A}* if $l_i \leq k$ for each i .

Let $PU_k(\mathcal{A}(D)) = \{m.g.u.(\mathcal{P}) : \mathcal{P} \text{ is a } k\text{-partition of } \mathcal{A}(D)\}$.

We provide an Algorithm VG_k :

Algorithm VG_k

Input: a finite set D of functor-argument structures

Output: a finite set \mathcal{G} of k -valued grammars such that for each $G \in \mathcal{G}$, $D \subseteq FL(G)$

1. Construct $GF(D)$.
2. Compute $PU_k(\mathcal{A}(D))$.
3. Let $VG_k(D) = \{\sigma(GF(D)) : \sigma \in PU_k(\mathcal{A}(D))\}$.

Lemma 5.3. (Kanazawa, 1998)

For $k \geq 2$, $\{G \in \mathcal{G}_k : G \text{ is in reduced form}\} \subset \bigcup \text{range}(VG_k)$.

Lemma 5.4.

Let G be a grammar in reduced form. Then there is a substitution σ such that $G = \sigma(GF(D))$ for some set of functor-argument structures D .

Proof. Let G be a grammar in reduced form. Find k , such that $G \in \mathcal{G}_k$. By lemma 5.3, we have $G \in \bigcup \text{range}(VG_k)$, so there are a substitution σ and a set of functor-argument structures D such that $G = \sigma(GF(D))$. \square

Let D be a finite set of functor-argument structures. We define:

$$MG_{\preceq}(D) = \{G \in OG(D) : \neg \exists_{G' \in OG(D)}(G' \prec G)\}.$$

Lemma 5.5.

The following are equivalent:

- (i) $G \in \mathcal{G}_{min, \preceq}$,
- (ii) there is a finite subset $D \subseteq FL(G)$ such that G is minimal with respect to (D, \preceq) .

Proof. (ii) \Rightarrow (i) is obvious. To prove (i) \Rightarrow (ii) assume $G \in \mathcal{G}_{min, \preceq}$. Let $\langle T_i \rangle_{i \in \mathbb{N}}$ be an infinite sequence enumerating $FL(G)$ ($\{T_i : i \in \mathbb{N}\} = FL(G)$). For any $G' \prec G$, $FL(G) \not\subseteq FL(G')$. Let $i_{G'}$ be the least i such that $T_i \notin FL(G')$. Then $\{T_0, \dots, T_{i_{G'}-1}\} \subseteq FL(G')$. Since $\{FL(H) : H \prec G\}$ has finite elasticity by (G3), this implies that there is $m \in \mathbb{N}$ such that $i_{G'} \leq m$ for all $G' \in \{H : H \prec G\}$. Then G must be minimal with respect to (D, \preceq) , where $D = \{T_0, \dots, T_m\}$. \square

Lemma 5.6.

If $G \in OG(D) \cap \mathcal{G}_{min, \preceq}$, then there exists a finite set D' such that $G \in MG_{\preceq}(D')$.

Proof. Assume $G \in OG(D)$ and $G \in \mathcal{G}_{min, \preceq}$. By lemma 5.5 we can find a finite subset $D_0 \subseteq FL(G)$ such that G is minimal with respect to (D_0, \preceq) . Then $D \cup D_0 \subseteq FL(G)$ and G is minimal with respect to $(D \cup D_0, \preceq)$. By lemma 5.2, $G \in OG(D \cup D_0)$. Then $G \in MG_{\preceq}(D \cup D_0)$. \square

Lemma 5.7.

For each $G \in \mathcal{G}_{min, \preceq}$, there exists $G_0 \in \mathcal{G}_{min, \preceq}$ such that G_0 is in reduced form, $FL(G_0) = FL(G)$ and $G_0 \in MG_{\preceq}(D)$ for some finite set $D \subseteq FL(G)$.

Proof. Assume $G \in \mathcal{G}_{min, \preceq}$. There exists G_0 such that G_0 is in reduced form and $FL(G_0) = FL(G)$. We show that $G_0 \in \mathcal{G}_{min, \preceq}$. Assume not. Then, there is a grammar G' such that $G' \prec G_0$ and $FL(G) \subseteq FL(G')$. We have $G_0 \preceq G$, by $G_0 \sqsubseteq G$ and the condition (G1). We obtain $G' \prec G$, which contradicts $G \in \mathcal{G}_{min, \preceq}$. We have to show that there exists some finite set $D \subseteq FLG$ such that $G_0 \in MG_{\preceq}(D)$. By lemma 5.6 and lemma 5.1, it is enough to show that $G_0 \in \mathcal{G}_{optimal}$. Suppose not. Since G_0 is in reduced form, it has no useless types, so by the definition of an optimal grammar there are types A and A' assigned to the same symbol by G_0 such that $A \neq A'$ and the relation $\{(A, A')\}$ has a unifier α . By lemma 5.4, there is a substitution σ such that $G_0 = \sigma(GF(D))$ for some set of functor-argument structures D . Then, by (G2) and (C1) we have $\alpha(\sigma(GF(D))) \prec \sigma(GF(D))$, so $\alpha(G_0) \prec G_0$. $FL(G_0) \subseteq FL(\alpha(G_0))$. It contradicts $G_0 \in \mathcal{G}_{min, \preceq}$. \square

Let μ_{FL} be a computable function that maps a non-empty finite set \mathcal{G} of grammars to a grammar $G \in \mathcal{G}$ such that $FL(G)$ is a minimal element of $\{FL(G) : G \in \mathcal{G}\}$.

We define the learning function $\varphi_{MG_{\preceq}}$ for $\langle \mathcal{G}_{min, \preceq}, \Sigma^F, FL \rangle$ in the following way:

- (i) $\varphi_{MG_{\preceq}}(\langle T_0 \rangle) = \mu_{FL}(MG_{\preceq}(\{T_0\}))$,
- (ii) $\varphi_{MG_{\preceq}}(\langle T_0, \dots, T_{i+1} \rangle) = \varphi_{MG_{\preceq}}(\langle T_0, \dots, T_i \rangle)$ if $\varphi_{MG_{\preceq}}(\langle T_0, \dots, T_i \rangle)$ is defined and $T_{i+1} \in FL(\varphi_{MG_{\preceq}}(\langle T_0, \dots, T_i \rangle))$,
- (iii) $\varphi_{MG_{\preceq}}(\langle T_0, \dots, T_{i+1} \rangle) = \mu_{FL}(MG_{\preceq}(\{T_0, \dots, T_{i+1}\}))$ otherwise.

Lemma 5.8.

$FL(\varphi_{MG_{\preceq}}(\langle T_0, \dots, T_i \rangle))$ is always a minimal element of $\{FL(G) : G \in MG_{\preceq}(\{T_0, \dots, T_i\})\}$

Proof. It is easy to show by definition of the function $\varphi_{MG_{\preceq}}$ and definition of the function μ_{FL} . \square

Theorem 5.9

$\varphi_{MG_{\preceq}}$ learns $\mathcal{G}_{min, \preceq}$.

Proof. Let $L \in \mathcal{FL}_{min, \preceq}$ and let $\langle T_i \rangle_{i \in \mathbb{N}}$ be an infinite sequence enumerating L . Let $G_0 \in \mathcal{G}_{min, \preceq}$ be a grammar in reduced form such that $FL(G_0) = L$.

Since $\{FL(H) : H \preceq G_0\}$ has finite elasticity, this implies that there is a set D_L such that L is the unique minimal element of the set $\{FL(H) : H \preceq G_0 \wedge D_L \subseteq FL(H)\}$.

By lemma 5.7, there exists $D_0 \subseteq L$ such that $G_0 \in MG_{\preceq}(D_0)$. By lemma 5.2, for all D' such that $D_0 \subseteq D' \subseteq L$ we have $G_0 \in MG_{\preceq}(D')$. If $D_0 \subseteq \{T_0, \dots, T_i\}$, then for all $H \in MG_{\preceq}(\{T_0, \dots, T_i\})$ we have $H \simeq G_0$, since \preceq is a linear preordering. Consequently, L is the unique minimal element of $\{FL(G) : G \in MG_{\preceq}(\{T_0, \dots, T_i\})\}$, for all i such that $D_L \cup D_0 \subseteq \{T_0, \dots, T_i\}$. Since $\varphi_{MG_{\preceq}}$ is conservative, then by lemma 5.8, this implies that $\varphi_{MG_{\preceq}}$ converges on $\langle T_i \rangle_{i \in \mathbb{N}}$ to a grammar $G_1 \in \mathcal{G}_{min, \preceq}$ such that $FL(G_1) = L$. \square

We give some examples of the linear preordering relations on a set of categorial grammars satisfying the above conditions.

Let $l = |\Sigma|$, and c_1, \dots, c_l be the elements of Σ arranged in a fixed order. For each grammar G over Σ , let $v(G)$ be the vector defined as follows:

$$v(G) = \langle n_1, \dots, n_l \rangle, \text{ where for } 1 \leq j \leq l, n_j = |\{A : G : c_j \rightarrow A\}|.$$

We define a relation $<_1$ on vectors in N^l in the following way:

$$\langle n_1, \dots, n_l \rangle <_1 \langle m_1, \dots, m_l \rangle \Leftrightarrow (n_1 + \dots + n_l < m_1 + \dots + m_l) \vee (n_1 + \dots + n_l = m_1 + \dots + m_l \wedge \exists_{1 \leq k \leq l} [(\forall_{i < k} n_i = m_i) \wedge (n_k < m_k)]).$$

Let \preceq_1, \preceq_2 be the linear preordering relations on a set of categorial grammars defined in the following way:

$$G_1 \preceq_1 G_2 \text{ iff } v(G_1) \leq_1 v(G_2),$$

$$G_1 \preceq_2 G_2 \text{ iff } \text{dom}(G_1) = \text{dom}(G_2) \wedge ((\sum_{v \in \text{dom}(G_1)} \text{card}(I_{G_1}(v))) \leq (\sum_{v \in \text{dom}(G_2)} \text{card}(I_{G_2}(v)))).$$

By theorem 5.9, $\mathcal{G}_{min, \preceq_1}$ and $\mathcal{G}_{min, \preceq_2}$ are learnable from structures.

Example 1

Let D be the following language sample:

$$\left\{ \begin{array}{l} (John, (likes, Helen)_1)_2 \rightarrow t \\ ((Only, Helen)_1, (likes, John)_1)_2 \rightarrow t \\ ((John, likes)_1, (only, Helen)_1)_2 \rightarrow t \end{array} \right.$$

We obtain the grammar $GF(D)$:

$$\begin{array}{l} John \rightarrow x_1, x_5, x_6/x_7 \\ likes \rightarrow x_7, (x_1 \setminus t)/x_2, (x_3 \setminus t)/x_5 \\ only \rightarrow x_3/x_4, (x_6 \setminus t)/x_8 \\ Helen \rightarrow x_2, x_4, x_8 \end{array}$$

It is not unifiable. The following substitutions:

- (1) $(x_1 : x_6/x_7, x_5 : x_6/x_7, x_3 : x_6/x_7, x_2 : x_6/x_7, x_4 : x_6/x_7, x_8 : x_6/x_7)$
- (2) $(x_2 : x_6 \setminus t, x_4 : x_6 \setminus t, x_8 : x_6 \setminus t, x_3 : x_6 \setminus t, x_5 : x_6 \setminus t, x_1 : x_6 \setminus t, x_7 : ((x_6 \setminus t)/(x_6 \setminus t)))$
- (3) $(x_4 : x_2, x_8 : x_2, x_3 : x_6 \setminus t, x_1 : x_6/x_7, x_5 : x_6/x_7)$
- (4) $(x_1 : x_6/x_7, x_5 : x_6/x_7, x_3 : x_6 \setminus t, x_4 : x_8, x_2 : x_8)$ are optimal unifiers of R_D .

We obtain the following grammars:

- (1): $John \rightarrow x_6/x_7$
 $likes \rightarrow x_7, ((x_6/x_7) \setminus t)/(x_6/x_7)$
 $only \rightarrow (x_6/x_7)/(x_6/x_7), (x_6 \setminus t)/(x_6/x_7)$
 $Helen \rightarrow x_6/x_7$
- (2): $John \rightarrow x_6 \setminus t, x_6/(((x_6 \setminus t) \setminus t)/(x_6 \setminus t))$
 $likes \rightarrow ((x_6 \setminus t) \setminus t)/(x_6 \setminus t)$
 $only \rightarrow (x_6 \setminus t)/(x_6 \setminus t)$
 $Helen \rightarrow x_6 \setminus t$
- (3): $John \rightarrow x_6/x_7$
 $likes \rightarrow x_7, ((x_6/x_7) \setminus t)/x_2, ((x_6 \setminus t) \setminus t)/(x_6/x_7)$
 $only \rightarrow (x_6 \setminus t)/x_2$
 $Helen \rightarrow x_2$
- (4): $John \rightarrow x_6/x_7$
 $likes \rightarrow x_7, ((x_6/x_7) \setminus t)/x_8, ((x_6 \setminus t) \setminus t)/(x_6/x_7)$
 $only \rightarrow (x_6 \setminus t)/x_8$
 $Helen \rightarrow x_8$

The grammar (2) is minimal for (D, \preceq_1) and (D, \preceq_2) .

REFERENCES

- Ajdukiewicz, K. (1935). Die syntaktische Konnexität. *Studia Philosophica*, 1, 1-27.
- Buszkowski, W. (1987). Solvable problems for classical categorial grammars. *Bulletin of the Polish Academy of Sciences, Mathematics*, 35, 373-382.
- Buszkowski, W., & Dziemidowicz, B. (2000). Restricted optimal unification. In: Hamamatsu, Y., Kosicki, W., Polkowski, L., Toho, M., & Yonekura, T. (Eds.), *Formal Methods and Intelligent Techniques in Control, Decision Making, Multimedia and Robotics*, 1-8.
- Buszkowski, W., & Penn, G. (1990). Categorial grammars determined from linguistic data by unification. *Studia Logica*, 49, 431-454.
- Doets, K. (1994). *From Logic to Logic Programming*, Cambridge, Mass.: MIT Press.

- Dziemidowicz, B. (2002). Optimal unification and learning algorithms for categorial grammars. *Fundamenta Informaticae*, 49, 297-308.
- Dziemidowicz, B. (2007). On learnability of restricted classes of categorial grammars. *Studia Logica*, 85, 153-169.
- Gold, E. M. (1967). Language identification in the limit. *Information and Control*, 10, 447-474.
- Kanazawa, M. (1996). Identification in the limit of categorial grammars. *Journal of Logic, Language and Information*, 5, 115-155.
- Kanazawa, M. (1998). *Learnable classes of categorial grammars*. Stanford: CSLI Publications FoLLI.
- Kapur, S. (1991). *Computational learning of languages*. Ph. D. disseration, Available as technical raport, 91-1234, Cornel University.
- Lloyd, J. W. (1987). *Foundations of Logic Programming*, Berlin: Springer.
- Marciniec, J. (1994). Learning categorial grammars by unification with negative constraints. *Journal of Applied Non-Classical Logics*, 4, 181-200.
- Marciniec, J. (1997). The method of unification in procedures of determining categorial grammars (in Polish). *Investigationes Linguisticae*, 2, 45-96.
- Osheron, D., de Jongh, D., Martin, E., & Weinstein, S. (1997). Formal learning theory. In: van Benthem, J., & ter Meulen, A. (Eds.), *Handbook of Logic and Language*, 737-775.