9

Learnability of Some Classes of Optimal Categorial Grammars

Jacek Marciniec

Abstract

In this paper we characterize a learnable class of classical categorial grammars, which lies inbetween (learnable) class of rigid grammars and (not learnable) class of optimal grammars. The learning function we provide is based on *guided optimal unification,* introduced in our paper Marciniec (2004).

Keywords CATEGORIAL GRAMMAR, LEARNING, UNIFICATION

9.1 Introduction

Kanazawa investigates in Kanazawa (1998) several classes of classical categorial grammars from the point of view of their learnability. All the learning functions discussed there are based on unification algorithms — the standard one and its *optimal* version, introduced by Buszkowski and Penn (1990). They also involve the process of reconstructing the grammar on the basis of some linguistic data, originated in Buszkowski (1987), van Benthem (1987). Only the learning function for the class of rigid grammars is based solely on unification. The rest is a combination of several operations, unification being just one of them. Kanazawa follows two ways of designing unification based learning procedures. The first one approach involves some preliminary operations *before* standard unification is set to work. It can be, for example, additional partitioning of the search space like in the case of *k-valued* grammars.

The second approach incorporates optimal unification accompanied with some selection mechanism *after* calculating all optimal unifiers. Usually the selection mechanism mentioned above is based on mini-

FG-MoL 2005.
James Rogers (ed.).

mality of some sort. Here Kanazawa's *least cardinality* case may serve as an example.

Both standard and optimal unification posses the feature desirable as far as learnability is concerned, namely compactness — the (optimal) unification image of an infinite set of types can be determined on the basis of its finite subset (cf. Marciniec (1997b,a, 2004)). However, the difference between the two is essential — standard unification algorithm outputs unique (if any) solution, whereas the total number of optimal unifiers usually grows with the increase of the input. Therefore, the only possible operation after completion of the former is to accept (or reject) the solution. The latter is more flexible. The application of a post unification choice function is evident. One can also imagine some activity prior to the unification process, though no such a possibility has been elaborated so far.

In Marciniec (2004), we put forward another solution — incorporating selection mechanism *into* unification engine itself. We described a general framework for optimal unification discovery procedure where the limitation of the number of outputs is achieved by controlling the order in which types are unified, 'decreasing' this way the nondeterminism of the original algorithm.

In this paper we develop the case when types that are alphabetic variants are to be unified first. Consequently, we introduce the class of semi-rigid grammars, which lies inbetween the class of rigid grammars and the class of optimal grammars (introduced in Kanazawa (1998)). We prove learnability of that class.

9.2 Preliminaries

We adopt most of the notation from Buszkowski and Penn (1990). $\mathsf{FS}(V)$ denotes the set of all *functor-argument* structures on the set of *atoms* V. If $A = (A_1, \ldots, A_n)_i$ is a structure, then a *substructure* A_i is the *functor* whereas each *substructure* A_j, for $j \neq i$, is an *argument* of A. For any set T of functor-argument structures, by $\mathsf{SUB}(T)$ we denote the smallest set satisfying the following conditions: $T \subseteq \mathsf{SUB}(T)$, if $(A_1, \ldots, A_n)_i \in \mathsf{SUB}(T)$ then also $A_j \in \mathsf{SUB}(T)$, for $j = 1, \ldots, n$. The functor (of the subfunctor) of a structure A is also its *subfunctor*. The only subfunctor of a structure A that is an atom will be denoted by $\uparrow_f(A)$.

Types are all the elements of the set $\mathsf{Tp} = \mathsf{FS}(\mathsf{Pr})$, where $\mathsf{Pr} = \mathsf{Var} \cup \{\mathsf{S}\}$ and Var is a countable set of *variables* and S is a *designated* primitive type ($\mathsf{S} \notin \mathsf{Var}$). A *substitution* is any homomorphism $\alpha : \mathsf{Tp} \mapsto \mathsf{Tp}$, such that $\alpha(\mathsf{S}) = \mathsf{S}$. Two types t_1 and t_2 are *alphabetic variants*

$(t_1 \bowtie t_2)$, if $t_2 = \alpha_2(t_1)$ and $t_1 = \alpha_1(t_2)$ for some substitutions α_1 and α_2. A substitution σ unifies a family of sets of types $\mathcal{T} = \{T_1, \ldots, T_n\}$, if each $\sigma[T_j]$ is a singleton. For a substitution α, by $\sim_\alpha$ we denote the relation defined as follows: $t_1 \sim_\alpha t_2$ iff $\alpha(t_1) = \alpha(t_2)$. For an equivalence relation $\sim$ and $T \subseteq \mathsf{Tp}$, by $T/\sim$ we will denote the partition induced on T by $\sim$. For $\mathcal{T} = \{T_1, \ldots, T_n\}$, we define $\mathcal{T}/\sim = T_1/\sim \cup \cdots \cup T_n/\sim$.

By *a (finite) classical categorial grammar* (from now on simply *a (finite) grammar*) we mean any (finite) relation $G \subseteq V \times \mathsf{Tp}$. However, in what follows, it will be convenient to regard a grammar as a family $\{I_G(v) : v \in V_G\}$, where V_G is a *lexicon* of G and the function I_G from V_G to 2^{Tp} is its *initial type assignment* ($I_G(v) = \{t \in \mathsf{Tp} : \langle v, t\rangle \in G\}$). The *terminal type assignment* T_G of G is defined by the following rule: $t_i \in T_G((A_1, \ldots, A_n)_i)$ iff there exist t_j such that $t_j \in T_G(A_j)$ for $j \neq i$, and $(t_1, \ldots, t_n)_i \in T_G(A_i)$. Each grammar G and a type t determine the *category* of type t: $\mathsf{CAT}_G(t) = \{A \in \mathsf{FS}(V_G) : t \in T_G(A)\}$. By $\mathsf{FL}(G) = \mathsf{CAT}_G(\mathsf{S})$ we denote the *(functorial) language* determined by G.

A language $L \subseteq \mathsf{FS}(V)$ is said to be *finitely describable* if there exists a finite grammar G such that $L \subseteq \mathsf{FL}(G)$.

$\mathsf{Tp}(G) = \bigcup_{v \in V_G} \mathsf{SUB}(I_G(v))$. By $\mathsf{Tp}_a(G)$ we denote the subset of $\mathsf{Tp}(G)$, consisting of only argument substructures. For a grammar G and a substitution α, $\alpha[G]$ will denote the grammar $\{\alpha[I_G(v)] : v \in V_G\}$.

A grammar G is *rigid* if $\mathsf{card}(I_G(v)) = 1$ for all $v \in V_G$. Let G be any grammar. Denote $\overline{V} = V \times \mathbb{N}$. The elements of $\overline{V}$ (possible 'copies' of atoms from V) will be denoted by v^i rather then $\langle v, i\rangle$. By *rigid counterpart* of G we mean any rigid grammar $\overline{G} \subseteq \overline{V} \times \mathsf{Tp}$, fulfilling the condition: $I_G(v) = \bigcup_{i=1}^{n} I_{\overline{G}}(v^i)$, where $n = \mathsf{card}(I_G(v))$. By $(\cdot)^{\urcorner}$ we denote a homomorphism from $\mathsf{FS}(\overline{V})$ to $\mathsf{FS}(V)$ fulfilling the condition: $\left(v^i\right)^{\urcorner} = v$, for each i. For any grammar G, we have $\mathsf{FL}(G) = \left(\mathsf{FL}(\overline{G})\right)^{\urcorner}$.

A type $t \in \mathsf{Tp}(G)$ is *useless* if $\mathsf{SUB}(\mathsf{FL}(\overline{G})) \cap \mathsf{CAT}_{\overline{G}}(t) = \emptyset$ for any rigid counterpart $\overline{G}$ of G.

A grammar G is said to be *optimal* if it has no useless type and for all $v \in V_G$, if $t_1, t_2 \in I_G(v)$ and $t_1 \neq t_2$ then $\{t_1, t_2\}$ is not unifiable.

Throughout this paper *learnability* means *learnability from structures* in the sense of Gold's identification in the limit (cf. Gold (1967), Kanazawa (1998), Jain et al. (1999), Osherson et al. (1997)).

9.3 Unification and Infinity

Below we recapitulate some results from Marciniec (2004):

Definition 17 A substitution σ partially unifies a family of possibly infinite sets of types $\mathcal{T} = \{T_1, \ldots, T_n\}$, if $\mathsf{card}(T_j) < \aleph_0$ for each j.

Definition 18 Let $\mathcal{T} = \{T_1, \ldots, T_n\}$ be a family of possibly infinite sets of types. An *optimal unifier* for $\mathcal{T}$ is a substitution σ fulfilling the following conditions:

- $\mathcal{T}/\sim_\sigma$ is finite,
- σ is the most general unifier of $\mathcal{T}/\sim_\sigma$,
- for $i \in \{1, \ldots, n\}$, $a, b \in T_i$, if $\sigma(a) \neq \sigma(b)$, then $\{\sigma(a), \sigma(b)\}$ is not unifiable.

Proposition 18 *The number of optimal unifiers of infinite set of types does not have to be finite.*

Theorem 19 (Marciniec (2004)) *Let $\mathcal{T} = \{T_1, \ldots, T_n\}$ be partially unifiable. For each optimal unifier η for $\mathcal{T}$ there exist a family $\mathcal{U} = \{U_1, \ldots, U_n\}$ and an optimal unifier σ for $\mathcal{U}$, such that for each $i \in \{1, \ldots, n\}$, $U_i \subseteq T_i$, U_i is finite and $\eta[T_i] = \sigma[U_i]$.*

Definition 19 Let $\approx$ be an equivalence relation on Tp. A substitution α respects $\approx$ on $T \subseteq \mathsf{Tp}$, if the following condition holds:

$$(\forall t_1, t_2 \in T)(t_1 \approx t_2) \rightarrow \alpha(t_1) = \alpha(t_2)).$$

A substitution α respects $\approx$ on $\mathcal{T} = \{T_1, \ldots, T_n\}$, if it respects $\approx$ on each T_i, for $i \in \{1, \ldots, n\}$.

Proposition 20 *For any family $\mathcal{T} = \{T_1, \ldots, T_n\}$, such that each T_i is bounded in length, there are only finitely many optimal unifiers respecting the relation $\bowtie$.*

Proof $\mathcal{T}/\bowtie$ is finite and unifiable and any substitution respecting $\bowtie$ unifies $\mathcal{T}/\bowtie$. □

Theorem 21 (Marciniec (2004)) *Let $\mathcal{T} = \{T_1, \ldots, T_n\}$ and let each T_i be finite. The following algorithm (guided optimal unification algorithm) outputs precisely the optimal unifiers for $\mathcal{T}$, respecting $\bowtie$:*

- *Compute mgu η for $\mathcal{T}/\bowtie$*
- *Compute all optimal unifiers for $\{\eta(T_1), \ldots, \eta(T_n)\}$.*

Theorem 22 (Marciniec (2004)) *let $\mathcal{T} = \{T_1, \ldots, T_n\}$ be partially unifiable family of nonempty, possibly infinite sets of types. There exists $\mathcal{U} = \{U_1, \ldots, U_n\}$, such that the set of all optimal unifiers for $\mathcal{U}$ respecting $\bowtie$ is the same as for any $\mathcal{V} = \{V_1, \ldots, V_n\}$, where each V_i is finite and $U_i \subseteq V_i \subseteq T_i$.*

9.4 Semi-rigid Grammars

Definition 20 Let $A \in \mathsf{FS}(V)$. We will describe a construction of a grammar $\mathsf{GF}(A)$ — *general form* determined by A. At first we choose any $\overline{A} \in \mathsf{SUB}(\overline{V})$, such that $\left(\overline{A}\right)^{\urcorner} = A$ and each atom from $\overline{V}$ occurs in $\overline{A}$ at most once (compare the definition of a rigid counterpart). Now, by induction, we define a mapping $\mapsto$ from $\mathsf{SUB}(\overline{A})$ to Tp:

- $\overline{A} \mapsto \mathsf{S}$,
- if $(\overline{A}_1, \ldots, \overline{A}_n)_i \mapsto t$ then for each $j \neq i$ we set $\overline{A}_j \mapsto x_j$, where x_j is a 'new' variable, and $\overline{A}_i \mapsto (x_1, \ldots, x_{i-1}, t, x_{i+1}, \ldots, x_n)_i$.

$\mathsf{GF}(A) = \{\langle v, t\rangle \in V \times \mathsf{Tp} : \exists i \in \mathbb{N}(v^i \mapsto t)\}$. Finally, for $L \subseteq \mathsf{FS}(V)$, assuming that $\mathsf{Tp}(\mathsf{GF}(A)) \cap \mathsf{Tp}(\mathsf{GF}(B)) = \emptyset$ when $A \neq B$, we define $\mathsf{GF}(L) = \bigcup_{A \in L} \mathsf{GF}(A)$. After Marciniec (1997b), we admit the case of L being infinite.

Proposition 23 *Let $G = \alpha[\mathsf{GF}(L)]$ for some substitution α. For any $x \in \mathsf{Var}$, if $x \in \mathsf{Tp}(\mathsf{GF}(L))$ then $\alpha(x) \in \mathsf{Tp}_a(G)$.*

Proposition 24 *For any rigidly describable L, $\mathsf{GF}(L)$ is bounded in length.*

Definition 21 Let L be finitely describable. By $OG_{\bowtie}(L)$ we will denote the set of all grammars G, such that $G = \eta[\mathsf{GF}(L)]$ for some optimal unifier η for $\mathsf{GF}(L)$, respecting $\bowtie$ on $\mathsf{GF}(L)$.

From Propositions 20 and 24, we have:

Corollary 25 *For any finitely describable L, the set $OG_{\bowtie}(L)$ is finite.*

Proposition 26 *Let G be a grammar. There exists a finite set $D \subseteq \mathsf{FL}(G)$ such that for each set E, if $D \subseteq E \subseteq \mathsf{FL}(G)$ then $OG_{\bowtie}(E) = OG_{\bowtie}(\mathsf{FL}(G))$.*

Proposition 27 *Let L be finitely describable. Then*

$$L \subseteq \bigcap_{G \in OG_{\bowtie}(L)} \mathsf{FL}(G).$$

Definition 22 Let $u, v \in \mathsf{Tp}$. We define the relation $\underset{\{u,v\}}{\leftrightsquigarrow} \subseteq \mathsf{SUB}(u) \times \mathsf{SUB}(v)$:

$$u \underset{\{u,v\}}{\leftrightsquigarrow} v \tag{9.21}$$

$$\text{if } (u_1, \ldots, u_n)_i \underset{\{u,v\}}{\leftrightsquigarrow} (v_1, \ldots, v_n)_i \text{ then } u_i \underset{\{u,v\}}{\leftrightsquigarrow} v_i \tag{9.22}$$

Proposition 28 *If $t'_1 \underset{\{t_1,t_2\}}{\leftrightsquigarrow} t'_2$ then $\alpha(t'_1) \underset{\{\alpha(t_1),\alpha(t_2)\}}{\leftrightsquigarrow} \alpha(t'_2)$, for any substitution α.*

Definition 23 We define the relation $\Lsh\!\Rsh \subseteq \mathsf{Tp}\times\mathsf{Tp}$:
$t_1 \Lsh\!\Rsh t_2$ iff

$$\left[\uparrow_f(t_1) \underset{\{t_1,t_2\}}{\leftrightsquigarrow} \uparrow_f(t_2)\right] \wedge [\uparrow_f(t_1) = \uparrow_f(t_2) \vee \{\uparrow_f(t_1), \uparrow_f(t_2)\} \subseteq \mathsf{Var}]\,.$$

Example 1 To see that the two following types are related in the sense of $\Lsh\!\Rsh$:

$$(x_1, ((x_2, x_3)_2, x_7)_1, x_2, (x_3, x_4)_2)_2,$$
$$((x_7, x_2)_1, (((x_3, x_2)_1, x_1)_2, (x_1, x_2)_2)_1, x_4, (x_1, x_5, x_4)_1)_2,$$

we simply disregard all the argument subtypes:

$$(\circ, ((\circ, x_3)_2, \circ)_1, \circ, \circ)_2,$$
$$(\circ, ((\circ, x_1)_2, \circ)_1, \circ, \circ)_2.$$

Proposition 29 *For any substitution α and a type t, if $\uparrow_f(t) = \mathsf{S}$ then $t \Lsh\!\Rsh \alpha(t)$.*

Definition 24 A grammar G is said to be *a semi-rigid grammar* if it is optimal and the following conditions hold:
for all $v \in V_G$, $t_1, t_2 \in I_G(v)$ and types t'_1, t'_2:

$$t_1 \neq t_2 \rightarrow \neg\,(t_1 \Lsh\!\Rsh t_2)\,, \tag{9.23}$$
$$t_1 \neq t_2 \wedge t'_1 \underset{\{t_1,t_2\}}{\leftrightsquigarrow} t'_2 \rightarrow t'_1 \notin \mathsf{Tp}_a(G) \vee t'_2 \notin \mathsf{Tp}_a(G). \tag{9.24}$$

Example 2 Below we will denote types traditionally, writing

$$t_1, \ldots, t_{i-1} \backslash t_i / t_{i+1}, \ldots, t_n$$

instead of $(t_1, \ldots, t_n)_i$. Let G_1 denote the following grammar over the lexicon $V = \{a, b\}$:

$$a \mapsto x, x\backslash\mathsf{S}$$
$$b \mapsto x\backslash x, (x\backslash\mathsf{S})\backslash(x\backslash\mathsf{S})$$

G_1 is not semi-rigid because both related types x and $x\backslash\mathsf{S}$ occur as argument subtypes in the type assignment. Our next example, G_2:

$$a \mapsto x, x\backslash\mathsf{S}$$
$$b \mapsto x\backslash x, x\backslash(x\backslash\mathsf{S})$$

is semi-rigid (observe that $\mathsf{FL}(G_1) = \mathsf{FL}(G_2)$). Notice also that semi-rigidness does not impose any restrictions concerning the number of types assigned to lexicon elements. For example, G_2 would admit for b any number of types of the form $x\backslash(x\backslash(\ldots(x\backslash\mathsf{S})\ldots))$.

Theorem 30 *The class of all semi-rigid grammars is learnable.*

Proof Let G be a semi-rigid grammar. At first, we will show:

$$G \in OG_{\bowtie}(\mathsf{FL}(G)). \tag{9.25}$$

Let $\overline{G}$ be any rigid counterpart of G. Denote $L = \mathsf{FL}(G)$ and $\overline{L} = \mathsf{FL}(\overline{G})$. Since $\overline{G}$ has no useless type, $\overline{G} = \eta[\mathsf{GF}(\overline{L})]$ for some mgu of $\mathsf{GF}(\overline{L})$ (cf. Kanazawa (1998)). We may assume that $\mathsf{GF}(L) = \left(\mathsf{GF}(\overline{L})\right)^{\Lsh}$, so $G = \eta[\mathsf{GF}(L)]$. Since $\{I_{\mathsf{GF}(\overline{L})}(v) : v \in V_{\mathsf{GF}(\overline{L})}\} = \{I_{\mathsf{GF}(L)}(v) : v \in V_{\mathsf{GF}(L)}\}/\sim_\eta$ and G is optimal, the substitution η is an optimal unifier of $\mathsf{GF}(L)$. To prove (9.25), we have to show that η respects $\bowtie$.

Let $t_i \bowtie t_j$, where $\{t_i, t_j\} \subseteq I_{\mathsf{GF}(L)}(v)$.

Suppose $\{\uparrow_f(t_i), \uparrow_f(t_j)\} \not\subseteq \mathsf{Var}$. By the definition of $\Lsh\!\Rsh$, we have $\uparrow_f(t_i) = \uparrow_f(t_j) = \mathsf{S}$. Since $\bowtie \subseteq \Lsh\!\Rsh$ and $\Lsh\!\Rsh$ is transitive, by Proposition 29, we get $\eta(t_i) \Lsh\!\Rsh\, \eta(t_j)$ and consequently , by (9.23), $\eta(t_i) = \eta(t_j)$.

Suppose then $\{\uparrow_f(t_i), \uparrow_f(t_j)\} \subseteq \mathsf{Var}$. Denote $x_i = \uparrow_f(t_i)$ and $x_j = \uparrow_f (t_j)$. Since $t_i \Lsh\!\Rsh\, t_j$ from Proposition 28 and the definition of $\Lsh\!\Rsh$ it follows:

$$\eta(x_i) \underset{\{\eta(t_i),\eta(t_j)\}}{\leftrightsquigarrow} \eta(x_j). \tag{9.26}$$

By Proposition 23, both $\eta(x_i)$ and $\eta(x_j)$ are argument substructures of $\mathsf{Tp}(G)$. By (9.24) and (9.26) we get $\eta(t_i) = \eta(t_j)$ again.

Let μ denote any computable function which, from any finite set of grammars, selects a grammar minimal with respect to the language it determines. Since the problem $\mathsf{FL}(G_1) \subseteq \mathsf{FL}(G_2)$ is decidable, such a function exists. We define a function φ from $\mathsf{FS}(V)^*$ to the set of grammars:

$$\varphi(\langle s_0, \ldots, s_i\rangle) = \mu(OG_\bowtie(\{s_0, \ldots, s_i\}).$$

Let $L = \mathsf{FL}(G)$ for some semi-rigid grammar G and let $L = \{s_i : i \in \mathbb{N}\}$. By Proposition 26, there exists $n \in \mathbb{N}$ such that for all $i \geqslant n$ we have $OG_\bowtie(\{s_0, \ldots, s_i\}) = OG_\bowtie(\mathsf{FL}(G))$. Denote $G' = \varphi(\langle s_0, \ldots, s_n\rangle)$. By Proposition 27, we get $\mathsf{FL}(G) \subseteq \mathsf{FL}(G')$. However, (9.25) and the minimality of G' leads to the conclusion that $\mathsf{FL}(G') = \mathsf{FL}(G) = L$. Hence, φ learns the class of all semi-rigid grammars. It is also easy to observe, that φ is responsive, set-driven and consistent on the class. $\square$

Proposition 31 *The learning function φ, defined in the proof of Theorem 30, is not prudent.*

Proof The grammar:

$$\begin{aligned} a &\mapsto y, y\backslash \mathsf{S} \\ b &\mapsto \mathsf{S}/z \\ c &\mapsto z, y\backslash \mathsf{S} \end{aligned}$$

determines the following language: $L = \{(a,a)_2, (b,c)_1, (a,c)_2\}$. It is easy to check, that φ converges on L to:

$$\begin{aligned} a &\mapsto x, x\backslash \mathsf{S} \\ b &\mapsto \mathsf{S}/(x\backslash \mathsf{S}) \\ c &\mapsto x\backslash \mathsf{S} \end{aligned}$$

that is not semi-rigid. □

References

Buszkowski, Wojciech. 1987. Discovery procedures for categorial grammars. In Klein and van Benthem (1987).

Buszkowski, Wojciech and Gerald Penn. 1990. Categorial grammars determined from linguistic data by unification. *Studia Logica* XLIX(4):431–454.

Gold, E Mark. 1967. Language identification in the limit. *Information and Control* 10:447–474.

Jain, Sanjay, Daniel Osherson, James S. Royer, and Arun Sharma. 1999. *Systems that Learn: An Introduction to Learning Theory*. Cambridge, Massachusetts: MIT Press, 2nd edn.

Kanazawa, Makoto. 1998. *Learnable Classes of Categorial Grammars*. Studies in Logic, Language and Information. Stanford, California: CSLI Publications & FoLLI.

Klein, Evan and Johan van Benthem, eds. 1987. *Categories, Polymorphism and Unification*. Amsterdam: Universiteit van Amsterdam.

Marciniec, Jacek. 1997a. Connected sets of types and categorial consequence. In C. Retoré, ed., *Logical Aspects of Computational Linguistics*, vol. 1328 of *Lecture Notes in Artificial Inteligence*, pages 292–309. Berlin: Springer.

Marciniec, Jacek. 1997b. Infinite set unification with application to categorial grammar. *Studia Logica* LVIII(3):339–355.

Marciniec, Jacek. 2004. Optimal unification of infinite sets of types. *Fundamenta Informaticae* 62(3,4):395–407.

Osherson, Daniel, Dick de Jongh, Eric Martin, and Scott Weinstein. 1997. Formal learning theory. In J. van Benthem and A. ter Meulen, eds., *Handbook of Logic and Language*, pages 737–775. Elsevier Science B. V.

van Benthem, Johan. 1987. Categorial equations. In Klein and van Benthem (1987).