



**HAL**  
open science

# Learnable Classes of General Combinatory Grammars

Erwan Moreau

► **To cite this version:**

Erwan Moreau. Learnable Classes of General Combinatory Grammars. 5th International Conference on Logical Aspects of Computational Linguistics, LACL 2005, Apr 2005, Bordeaux, France. pp.189–204. hal-00487064

**HAL Id: hal-00487064**

**<https://hal.science/hal-00487064>**

Submitted on 27 May 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Learnable Classes of General Combinatory Grammars

Erwan Moreau

LINA - FRE CNRS 2729 - Université de Nantes  
2 rue de la Houssinière - BP 92208 - 44322 Nantes cedex 3  
`Erwan.Moreau@univ-nantes.fr`

**Abstract.** Kanazawa has shown that  $k$ -valued classical categorial grammars have the property of finite elasticity [1], which is a sufficient condition for learnability. He has also partially extended his result to general combinatory grammars, but left open the question of whether some subsets of general combinatory grammars have finite elasticity. We propose a new sufficient condition which implies learnability of some classes of  $k$ -valued general combinatory grammars, focusing on the way languages are expressed through a grammatical formalism rather than the classes of languages themselves.

## 1 Introduction

The problem of grammatical inference refers to the process of learning grammars and/or languages from data. Applied to natural languages, this problem consists in guessing, from a set of data corresponding to a (natural) language, “something which represents this language”: a grammar. But what kind of grammar? In this article we propose to study learnability of the formalisms used to represent languages rather than the languages themselves. This question is especially important for natural languages, since a lot of formalisms exist to represent them. As a consequence, the fact that a class of natural languages is learnable is only interesting if there is a way to represent it in a usable and linguistically appropriate way.

Gold’s model of identification in the limit is one of the most important formalizations of the learning process [2]. In this model, the learner must be able to guess the right language after a finite number of examples, from an infinite set of sentences belonging to this language. Several positive results have been obtained in Gold’s model, in particular with categorial grammars: using Buszkowski’s learning algorithm [3] for classical categorial grammars (also called AB grammars), Kanazawa [1] has shown that  $k$ -valued AB grammars<sup>1</sup> are learnable from strings. More precisely, Kanazawa has shown that the class of structure

---

<sup>1</sup> A grammar is *k-valued* if each word is defined by at most  $k$  different types in the lexicon. In the special case where  $k = 1$ , the grammar is said to be *rigid*. This latter case is a strong restriction over the expressive power of the corresponding language: Kanazawa has shown that the class of  $k$ -valued AB grammars languages is strictly included in the class of  $(k + 1)$ -valued languages. Furthermore, there are words in

languages generated by rigid AB grammars has the property of finite elasticity ([4], [5]), which implies learnability and also that learnability can be extended to  $k$ -valued grammars and to string languages. Kanazawa has also generalized his learnability result to *general combinatory grammars*: in this framework it is possible to consider any set of operators and any set of universal rules instead of the usual AB grammars operators and rules. However Kanazawa shows only that rigid structure languages are learnable, because finite elasticity does not hold in this more general case. As a consequence, Kanazawa does not provide any positive learnability result about  $k$ -valued general combinatory grammars, which is a strong limitation to the usefulness of this result in the viewpoint of natural languages.

The question of whether some subsets of general combinatory grammars have finite elasticity was left open by Kanazawa. Costa Florêncio has given in [6] a sufficient condition for finite elasticity in the form of restrictions over rules. Here we propose another sufficient condition which does not include Costa Florêncio's one, but which relates more to grammatical formalisms than to technical constraints (a brief comparison between the two results is given in section 4.2).

The linguistic interest for general combinatory grammars lies in the fact that it allows to express various formalisms through the set of universal rules: one can see the set of rules (and operators) as a parameter of the class of languages which is studied. This point is particularly interesting in the framework of grammatical inference, because it permits to test rather easily whether a given grammatical formalism enjoys some learnability properties: if the formalism can be expressed using rules in the form of general combinatory grammars and if these rules fulfill the conditions described in section 3.2, then learnability is proven. Several examples of such formalisms which are (at least partially) learnable are given in section 4.

## 2 General Combinatory Grammars

The name “general combinatory grammars” is used by Kanazawa to define any class of grammars using a certain set of operators and universal rules (expressed as the rewriting of a sequence of terms containing variables into another term). It refers to *combinatory categorial grammars*, defined by Steedman [7], who proposed to add several rules to AB grammars, in order to give a better syntactic description of natural languages. This means that AB grammars, as well as combinatory categorial grammars, are instances of general combinatory grammars.

**Definition 1 (Terms).** *Given a set  $S$  of operators and a set  $\mathcal{V}$  of variables, the set of  $S$ -terms over  $\mathcal{V}$  is the smallest set such that*

- any  $v \in \mathcal{V}$  is an  $S$ -term over  $\mathcal{V}$ ,

---

natural languages that require several definitions: for example, the grammatical word “to” should not have the same type when it is used with an infinitive or used as a preposition.

- for any operator  $f \in \mathcal{S}$  with  $\text{arity}(f) = n$ , if  $t_1, \dots, t_n$  are  $\mathcal{S}$ -terms over  $\mathcal{V}$  then  $f(t_1, \dots, t_n)$  is an  $\mathcal{S}$ -term over  $\mathcal{V}$ .<sup>2</sup>

The size  $\|t\|$  and height  $h(t)$  of a term  $t$  are defined in the usual way: if  $t$  is a variable then  $\|t\| = 1$  and  $h(t) = 0$ . Otherwise if  $t = f(t_1, \dots, t_n)$  then  $\|t\| = \sum_{1 \leq i \leq n} \|t_i\| + 1$  and  $h(t) = \max(\{h(t_i) \mid 1 \leq i \leq n\}) + 1$ .

$\#_u(t)$  denotes the number of occurrences of a term  $u$  in a term  $t$ :

- if  $u = t$  then  $\#_u(t) = 1$
- otherwise  $u \neq t$ :
  - if  $t$  is a variable, then  $\#_u(t) = 0$
  - if  $t = f(t_1, \dots, t_n)$  then  $\#_u(t) = \sum_{1 \leq i \leq n} \#_u(t_i)$

**Definition 2 (Universal rule).** Let  $\mathcal{S}$  be a set of operators. Given a set of variables  $\text{Var}(R)$ , a universal rule  $R$  over  $\mathcal{S}$  is any expression of the form  $A_1, \dots, A_n \rightarrow A_0$ , where each  $A_i$  is an  $\mathcal{S}$ -type over  $\text{Var}(R)$ .

**Definition 3 ( $\mathcal{R}$ -grammar).** Let  $\mathcal{S}$  be a set of operators and  $\mathcal{R}$  a set of universal rules over  $\mathcal{S}$ . An  $\mathcal{R}$ -grammar is a system  $G = \langle \Sigma, Pr, s, \triangleright \rangle$  where

- $\Sigma$  is the vocabulary,
- $Pr$  is a finite set of variables, called primitive types. The set of types  $Tp$  is then defined as the set of  $\mathcal{S}$ -terms over  $Pr$ .
- $s$  is an  $\mathcal{S}$ -term over  $\emptyset$ : this is the special type for valid sentences (see definition 8).<sup>3</sup>
- $\triangleright$  is a binary relation assigning one or several types to each word in the vocabulary:  $\triangleright \subseteq \Sigma \times Tp$ . Each couple  $w \triangleright t$  in this relation is called a lexical rule.

*Remark:* In the framework of categorial grammars, the special type  $s$  is traditionally defined as one of the primitive types. On the contrary, we can benefit here of a more general definition of the set of operators  $\mathcal{S}$ , which permits to define  $s$  as a type using only these operators. This way it is really considered different from the other types, which is particularly relevant in the framework of grammatical inference. Furthermore, this definition permits to take into account new interesting formalisms (see example in 4.3).

$\text{Lex}(G)$  is defined as the set of types used in the lexicon:  $\text{Lex}(G) = \{t \in Tp \mid \text{there exists } w \text{ such that } w \triangleright t\}$ .

**Definition 4 (One-step derivation).** Let  $\mathcal{S}$  be a set of operators,  $\mathcal{R}$  a set of rules over  $\mathcal{S}$  and  $G$  an  $\mathcal{R}$ -grammar. For every rule  $R \in \mathcal{R}$ ,  $R = A_1, \dots, A_n \rightarrow A_0$ , the relation  $\rightarrow_R \subseteq Tp^+ \times Tp$  is defined as:  $t_1, \dots, t_n \rightarrow_R t_0$  if and only if there exists a substitution  $\sigma : \text{Var}(R) \mapsto Tp$  such that  $\sigma(A_i) = t_i$  for all  $i$ .

The relation  $\rightarrow$  is defined as  $t_1, \dots, t_n \rightarrow t_0$  if there exists a rule  $R \in \mathcal{R}$  such that  $t_1, \dots, t_n \rightarrow_R t_0$ .

<sup>2</sup> The special case where  $\text{arity}(f) = 0$  is included in this definition: if  $f$  is such an operator, then it is an  $\mathcal{S}$ -term over  $\mathcal{V}$ .

<sup>3</sup> *Remark:*  $s$  is an  $\mathcal{S}$ -term over the empty set, which implies that there is at least one operator  $f$  in  $\mathcal{S}$  such that  $\text{arity}(f) = 0$ .

From this definition it is possible to define the string derivation relation in the usual following way: let  $\Rightarrow$  be the relation defined as  $\alpha \beta \gamma \Rightarrow \alpha t_0 \gamma$  if and only if  $\beta \rightarrow t_0$ , with  $\alpha, \gamma \in Tp^*$  and  $\beta \in Tp^+$ . The relation  $\Rightarrow^*$  is defined as the reflexive and transitive closure of  $\Rightarrow$ . A string  $w_1, \dots, w_n$  is valid for the  $\mathcal{R}$ -grammar  $G$  if there exists a sequence of types  $t_1, \dots, t_n$  such that for all  $i$   $w_i \triangleright t_i$  and  $t_1, \dots, t_n \Rightarrow^* s$ .

Nevertheless, we will rather use the below definition that links derivation to the existence of a “structure”, because we are also interested in structure languages.

**Definition 5 ( $\mathcal{R}$ -structure).** Let  $\mathcal{S}$  be a set of operators and  $\mathcal{R}$  a set of rules over  $\mathcal{S}$ ,  $\mathcal{R} = \{R_1, \dots, R_m\}$ . Given a vocabulary  $\Sigma$ , the set of  $\mathcal{R}$ -structures  $\mathcal{SL}_{\mathcal{R}}$  is the smallest set such that

- any word  $w \in \Sigma$  belongs to  $\mathcal{SL}_{\mathcal{R}}$ ,
- for any rule  $R_i = A_1, \dots, A_n \rightarrow A_0$ ,  
if  $T_1, \dots, T_n \in \mathcal{SL}_{\mathcal{R}}$  then  $[R_i](T_1, \dots, T_n) \in \mathcal{SL}_{\mathcal{R}}$ .

**Definition 6 (Yield of an  $\mathcal{R}$ -structure).** Let  $\mathcal{S}$  be a set of operators and  $\mathcal{R}$  a set of rules over  $\mathcal{S}$ . The yield of an  $\mathcal{R}$ -structure  $T$ , denoted  $yield(T)$ , is the sequence of words occurring at it leaves. Formally:

- if  $T = w$  with  $w \in \Sigma$ , then  $yield(T) = w$
- if  $T = [R_i](T_1, \dots, T_n)$ , then  $yield(T) = yield(T_1), \dots, yield(T_n)$ .

**Definition 7 (Instance of an  $\mathcal{R}$ -structure).** Let  $\mathcal{S}$  be a set of operators and  $\mathcal{R}$  a set of rules over  $\mathcal{S}$  and  $G$  an  $\mathcal{R}$ -grammar. Given an  $\mathcal{R}$ -structure  $T$ , a couple  $\langle \alpha, t_0 \rangle$ , where  $\alpha \in Tp^+$  and  $t_0 \in Tp$ , is an instance of  $T$  for  $G$  if the following condition holds:

- if  $T = w$  with  $w \in \Sigma$ , then  $\alpha = t_0$  and  $w \triangleright t_0$ .
- if  $T = [R_i](T_1, \dots, T_n)$ , then there exist  $n$  couples  $\langle \alpha_1, t_1 \rangle, \dots, \langle \alpha_n, t_n \rangle$  such that  $\langle \alpha_i, t_i \rangle$  is an instance of  $T_i$  for all  $i$ ,  $\alpha = \alpha_1 \bullet \dots \bullet \alpha_n$  and  $t_1, \dots, t_n \rightarrow_{R_i} t_0$ .

**Definition 8 (Languages of a grammar).** Let  $\mathcal{S}$  be a set of operators,  $\mathcal{R}$  a set of rules over  $\mathcal{S}$  and  $G = \langle \Sigma, Pr, s, \triangleright \rangle$  an  $\mathcal{R}$ -grammar.

- An  $\mathcal{R}$ -structure  $T$  belongs to the structure language defined by  $G$ , denoted  $T \in SL(G)$ , if there exists an instance  $\langle \alpha, s \rangle$  of  $T$  for  $G$ .
- A string (sequence of words)  $w_1 \dots w_n$  belongs to the string language defined by  $G$ , denoted  $w_1 \dots w_n \in L(G)$  if there exists an  $\mathcal{R}$ -structure  $T \in SL(G)$  such that  $w_1 \dots w_n \in yield(T)$ .

*Example 1 (AB grammars).*

Let define the set of operators  $\mathcal{S}_{AB}$  as  $\mathcal{S}_{AB} = \{/(2), \backslash(2), s(0)\}$  and the set of rules  $\mathcal{R}_{AB}$  as the set containing only the two following rules:

$$\begin{array}{lll} FA : & A/B \ B \rightarrow A & Var(FA) = \{A, B\} \\ BA : & B \ B \backslash A \rightarrow A & Var(BA) = \{A, B\} \end{array}$$

The class of  $\mathcal{R}_{AB}$ -grammars corresponds exactly to the class of AB grammars.

Let  $G$  be the  $\mathcal{R}_{AB}$ -grammar defined with the lexicon  $\{ Peter \triangleright n ; Mary \triangleright n ; loves \triangleright (n \setminus s)/n \}$ . the  $\mathcal{R}_{AB}$ -structure  $T = BA(Peter, FA(loves, Mary))$  belongs to  $SL(G)$ , because  $\langle (n, (n \setminus s)/n), s \rangle$  is an instance of  $T$  for  $G$ . Since  $yield(T) = \{ (Peter \text{ loves } Mary) \}$ , the sentence “*Peter loves Mary*” belongs to  $L(G)$ .

**Definition 9 (#G).**

We denote the number of elements in a set  $E$  by  $\#E$ . Applied to a grammar  $G = \langle \Sigma, Pr, s, \triangleright \rangle$ , the set considered is the set of the lexical rules in  $G$ :

$$\#G = \#\{ (w, t) \text{ with } w \in \Sigma \text{ and } t \in Tp \mid w \triangleright t \}$$

As a consequence of this definition, a grammar  $G$  is included in a grammar  $G'$ , denoted  $G \subseteq G'$ , if all lexical rules in  $G$  are also defined in  $G'$ .

**Proposition 1.** Let  $G$  and  $G'$  be two  $\mathcal{R}$ -grammars. If  $G \subseteq G'$  then  $SL(G) \subseteq SL(G')$ .

*Proof.* Let  $T \in SL(G)$ : there exists an instance  $\langle t_1 \dots t_n, s \rangle$  of  $T$  for  $G$ .  $G \subseteq G'$  implies that if  $w_i \triangleright_G t_i$  then  $w_i \triangleright_{G'} t_i$ , so  $\langle t_1 \dots t_n, s \rangle$  is also an instance of  $T$  for  $G'$ , thus  $T \in SL(G')$ .  $\square$

**Definition 10 ( $k$ -valued grammar).**  $G = \langle \Sigma, Pr, s, \triangleright \rangle$  is  $k$ -valued if for any word  $w \in \Sigma$ :  $\#\{ t \in Tp \mid w \triangleright t \} \leq k$ .

**Proposition 2.** For any  $k \geq 0$  there exists  $k' \geq 0$  such that

$$\{ G \mid G \text{ is } k\text{-valued} \} \subseteq \{ G \mid \#G \leq k' \}$$

*Proof.* Clearly any  $k$ -valued grammar can not have more than  $k' = k \times \#\Sigma$  rules.  $\square$

**Definition 11 (Equivalent grammars).** Two  $\mathcal{R}$ -grammars  $G$  and  $G'$  are said to be equivalent if they differ only by a renaming of their primitive types.

**Proposition 3.** If  $G$  and  $G'$  are equivalent, then  $SL(G) = SL(G')$  (and as a consequence  $L(G) = L(G')$ ).

### 3 $k$ -valued Flat Grammars Are Learnable from Structures

The distinction between *functors* and *arguments* is important in the first learning algorithm for AB grammars defined by Buszkowski [3], and also in the extensions given by Kanazawa [1]. However it is not the main point used by Kanazawa to show finite elasticity of  $k$ -valued AB grammars. Here we propose to focus on this distinction and its consequences in the learnability viewpoint, and see how this

particular property of AB grammars can be generalized. Thus we obtain a sufficient condition for learnability of grammars through the criterion of “flatness”.

To show that such languages are learnable we will use Shinohara’s criterion of *bounded finite thickness*. The general framework proposed by Shinohara has the advantage to take into account the way languages are represented in the formalization of learning. Actually this is not the case in the basic definition of Gold’s identification in the limit, nor in Wright’s criterion of finite elasticity, where the fact that languages must have a grammatical representation is only implicit. But in the natural languages viewpoint this point is essential. In particular, it is probably more interesting to be able to test whether a given grammatical formalism has some learnability properties (due to the formalism itself) than to know if it contains a learnable subclass of languages.

### 3.1 Gold’s Model of Identification in the Limit

In the following we use the abstract word *objects* to refer to the elements of a language. These objects may be strings (that is simple sequences of words), but also structures which may be more or less complex.

Gold’s model of identification in the limit is a formal model of learning [2]. In this model, the learner has to guess the right language from an infinite sequence of objects belonging to this language (positive examples). Formally, let  $\phi$  be a learning function, and  $L$  a language. Let  $\langle a_i \rangle_{i \in \mathbb{N}}$  be any infinite sequence of objects such that  $a \in \langle a_i \rangle_{i \in \mathbb{N}}$  if and only if  $a \in L$ .  $\phi$  *converges* to  $L$  if there exists  $n \in \mathbb{N}$  such that  $\phi(\langle a_1, a_2, \dots, a_n \rangle) = L$ , and for all  $i > n$   $\phi(\langle a_1, a_2, \dots, a_i \rangle) = L$ . A class of languages  $\mathcal{L}$  is learnable if there exists a learning function  $\phi$  such that for all  $L \in \mathcal{L}$ ,  $\phi$  converges to  $L$  for any enumeration of  $L$ .

Wright has proposed in [4], [5] a sufficient condition for learnability, called *finite elasticity*. A class  $\mathcal{L}$  has infinite elasticity if there exist two infinite sequences  $a_0, a_1, \dots$  of objects and  $L_1, L_2, \dots$  of languages such that for any  $k \geq 1$   $\{a_0, a_1, \dots, a_{k-1}\} \subseteq L_k$  but  $a_k \notin L_k$ . A class  $\mathcal{L}$  has finite elasticity if  $\mathcal{L}$  does not have infinite elasticity. Kanazawa has shown an important theorem about finite elasticity in [1]: if a class  $\mathcal{L}_1$  has finite elasticity and there exists a *finite-valued*<sup>4</sup> relation between  $\mathcal{L}_1$  and  $\mathcal{L}_2$ , then  $\mathcal{L}_2$  has also finite elasticity.

Shinohara proposed in [8] a framework in which languages are defined through a set of expressions, called a formal system. His definition of formal systems corresponds to a general definition of grammars, except that these grammars must consist of a set of elements (called expressions), that correspond usually to rules: a *concept defining framework* is a triple  $\langle \mathcal{U}, \mathcal{E}, \mathcal{M} \rangle$  of a universe  $\mathcal{U}$  of objects, a set  $\mathcal{E}$  of expressions and a semantic mapping  $\mathcal{M}$  that maps finite subsets of  $\mathcal{E}$  (grammars) to subsets of  $\mathcal{U}$  (languages). A semantic mapping  $\mathcal{M}$  is monotonic if  $G \subseteq G'$  implies  $\mathcal{M}(G) \subseteq \mathcal{M}(G')$ .

*Example 2.* Let  $\mathcal{S}_{AB}$  and  $\mathcal{R}_{AB}$  be the sets of AB grammars operators and rules, as defined in example 1. Given a vocabulary  $\Sigma$ , let  $\mathcal{E}_{\mathcal{R}}$  be the set of all possible

<sup>4</sup> A relation  $R \subseteq \mathcal{U}_1 \times \mathcal{U}_2$  is *finite-valued* iff for every  $a \in \mathcal{U}_1$  there are at most finitely many  $b \in \mathcal{U}_2$  such that  $Rab$ .

lexical rules  $w \triangleright t$ , with  $w \in \Sigma$  and  $t$  an  $\mathcal{S}_{AB}$ -term. Then the concept defining framework  $\langle \mathcal{S}\mathcal{L}_{\mathcal{R}}, \mathcal{E}_{\mathcal{R}}, \mathcal{S}\mathcal{L} \rangle$  describes the grammatical system of the structure language of AB grammars, and the concept defining framework  $\langle \Sigma, \mathcal{E}_{\mathcal{R}}, L \rangle$  describes the grammatical system of the string language of AB grammars.

Given a concept defining framework  $\langle \mathcal{U}, \mathcal{E}, \mathcal{M} \rangle$ , a grammar  $G \subseteq \mathcal{E}$  is *reduced* with respect to a finite set of objects  $D \subseteq \mathcal{U}$  if  $D \subseteq \mathcal{M}(G)$  but  $D \not\subseteq \mathcal{M}(G')$  for any grammar  $G' \subset G$ .

**Definition 12 (Bounded finite thickness).** *A concept defining framework  $\langle \mathcal{U}, \mathcal{E}, \mathcal{M} \rangle$  has bounded finite thickness if  $\mathcal{M}$  is monotonic and the set*

$$\{ \mathcal{M}(G) \mid G \text{ is reduced with respect to } D \text{ and } \#G \leq n \}$$

*is finite for any  $D \subseteq \mathcal{U}$  and any  $n \geq 0$ .*

Shinohara has shown that if a concept defining framework  $\langle \mathcal{U}, \mathcal{E}, \mathcal{M} \rangle$  has bounded finite thickness, then for any  $n \geq 0$  the class  $\mathcal{L}_n = \{ \mathcal{M}(G) \mid G \subseteq \mathcal{E} \text{ and } \#G \leq n \}$  has finite elasticity (and then is learnable).

General combinatory grammars do not enjoy (in the general case) the same learnability properties as AB grammars. This point is shown for example in [9], where Costa-Florêncio gives an example of a class of rigid general combinatory grammars which is not learnable.

### 3.2 Flat Grammars

The sufficient condition for learnability of general combinatory grammars that we propose below lies in the restriction to *flat* grammars. Informally, this restriction is based on the distinction between *principal* and *argument* types of a complex type: each position in an operator is defined as principal or argument position, and in the latter case any type built with this operator must verify that only an atomic (or primitive) type is allowed in this position. This way the height of any type in argument position is bounded, without bounding the height (nor the size) of a type in general. We will show that this condition together with suitable restrictions on the form of the rules (see definition 14) allow that the class of languages has finite elasticity.

**Definition 13 (Flat types).** *Let  $\mathcal{S}$  be a set of operators, and for each operator  $f$  in  $\mathcal{S}$  let  $arg_f$  be a function from  $\{1, \dots, arity(f)\}$  to  $\{0, 1\}$ . Given a set of variables  $\mathcal{V}$ , the set of flat  $\mathcal{S}$ -types over  $\mathcal{V}$ , denoted  $FT(\mathcal{V})$ , is defined as the smallest set such that*

- $\mathcal{V} \subseteq FT(\mathcal{V})$ ,
- for any  $f \in \mathcal{S}$  with  $arity(f) = n$ , if  $t_1, \dots, t_n$  are flat  $\mathcal{S}$ -types over  $\mathcal{V}$  then  $f(t_1, \dots, t_n) \in FT(\mathcal{V})$  if  $t_i \in \mathcal{V}$  for any  $i$  such that  $arg_f(i) = 1$ .



A subtype  $u$  in a type  $t$  is said to be in *argument position* in  $t$  if there is a subtype  $w$  in  $t$  such that  $w = f(t_1, \dots, t_i, \dots, t_n)$ ,  $t_i = u$  and  $\text{arg}_f(i) = 1$ .<sup>5</sup> Clearly, if  $t$  is a flat  $\mathcal{S}$ -type over  $\mathcal{V}$  then any subtype  $u$  which is in argument position in  $t$  must be in  $\mathcal{V}$ .

**Definition 14 (Flat universal rule).** *Let  $R = A_1, \dots, A_n \rightarrow A_0$  be a universal rule over a set  $\mathcal{S}$  of operators.  $R$  is a flat rule if the following conditions hold:*

- for any  $v \in \text{Var}(R)$ , if  $v$  is a subtype of  $A_0$  then there exists  $A_i$  with  $i \geq 1$  such that  $v$  is also a subtype of  $A_i$ .
- For each  $i$ ,  $A_i \in \text{FT}(\text{Var}(R))$ .
- For each variable  $v \in \text{Var}(R)$  such that  $v$  is not in argument position in the left hand side of  $R$  (that is there is no  $A_i$  with  $i \geq 1$  in which  $v$  is in argument position):
  - $v$  is not in argument position in  $A_0$ ,
  - and for each  $i \geq 1$ :  $\#_v(A_i) \leq \#_v(A_0)$ .

Given a set  $\mathcal{R}$  of flat universal rules,  $G = \langle \Sigma, Pr, s, \triangleright \rangle$  is a flat  $\mathcal{R}$ -grammar if every type  $t \in \text{Lex}(G)$  is flat.

*Example 3.* the argument positions of AB grammars operators are defined in the following way:

- $\text{arg}_f(1) = \text{arg}_f(2) = 0$
- $\text{arg}_g(2) = \text{arg}_g(1) = 1$

This means that in  $A/B$  (as well as in  $B \setminus A$ )  $B$  is the only argument type. One can see that AB grammars rules (defined in example 1) verify the conditions of flat universal rules: In particular there is only one variable in each rule (namely  $A$ ) which does not occur in argument position in the left hand side, and the condition that  $A$  must not occur in argument position in the right hand side is fulfilled.

It is important to notice that all AB grammars types are not flat: for example,  $(a/b)/c$  is flat whereas  $a/(b/c)$  is not. The relationship between flat AB grammars and unrestricted AB grammars is discussed in section 4.1.

The following proposition shows that “flatness” of types is closed under derivation with flat rules. This means that it is not necessary to add a restriction to each derivation step so that it outputs only flat types, which would be counter-intuitive: if all types are flat at the beginning and only flat rules are used, then only flat types can appear.

**Proposition 4.** *Let  $R$  be a flat rule,  $Pr$  a finite set of variables and  $t_1, \dots, t_n$  flat  $\mathcal{S}$ -types over  $Pr$ . If  $t_1, \dots, t_n \rightarrow_R t_0$ , then  $t_0$  is also a flat  $\mathcal{S}$ -type.*

<sup>5</sup> *Remark:* in the case where there are several occurrences of  $u$  in  $t$ , it is sufficient that one occurrence verifies the condition.

*Proof.* Let  $R = A_1, \dots, A_n \rightarrow A_0$ .  $t_1, \dots, t_n \rightarrow_R t_0$  implies that there is a substitution  $\sigma$  such that  $\sigma(A_i) = t_i$  for all  $i$ .

Suppose  $t_0 = \sigma(A_0)$  is not a flat type. Then there must be a subtype  $u$  in  $\sigma(A_0)$  such that  $u = f(u_1, \dots, u_m)$ , and there is a  $k$ ,  $1 \leq k \leq m$ , such that  $\text{arg}_f(k) = 1$  and  $u_k \notin Pr$ . For any variable  $v_j \in \text{Var}(R)$  (used in any  $t_i$ ) we have  $\sigma(v_j) \in FT(Pr)$ , because  $\sigma(v_j)$  is a subtype of at least one  $t_i$ ,  $i \geq 1$ , and  $t_i \in FT(Pr)$ . Therefore there is no  $v_j$  such that  $u$  is a subtype of  $\sigma(v_j)$ . As a consequence,  $A_0$  contains  $f(a_1, \dots, a_m)$  as a subtype, with  $\sigma(a_k) \notin Pr$ . But  $A_0 \in FT(\text{Var}(R))$  and  $\text{arg}_f(k) = 1$ , so  $a_k \in \text{Var}(R)$ . Since  $R$  is a flat rule and  $a_k$  is in argument position in  $A_0$ ,  $a_k$  must appear in argument position in some  $A_i$ ,  $i \geq 1$ . Thus  $\sigma(A_i)$  contains  $\sigma(a_k)$  in argument position whereas  $\sigma(a_k) \notin Pr$ : this contradicts the hypothesis that  $\sigma(A_i) = t_i$  is a flat type.  $\square$

The following propositions are used to show that flat general combinatory grammars have bounded finite thickness. The proof, which is similar to Shinozaki's one in [8], consists in bounding the size of the possible reduced grammars.

**Proposition 5.** *Let  $R$  be a flat rule,  $Pr$  a finite set of variables and  $t_1, \dots, t_n$  flat  $\mathcal{S}$ -types over  $Pr$ . If  $t_1, \dots, t_n \rightarrow_R t_0$ , then for all  $i \geq 1$*

$$\|t_i\| \leq \|t_0\| + M_R, \text{ where } M_R = \max(\{\|A_i\| \mid 1 \leq i \leq n\}) - \|A_0\|$$

*Proof.* Let  $R = A_1, \dots, A_n \rightarrow A_0$  and  $\text{Var}(R) = \{v_1, \dots, v_m\}$ .  $t_1, \dots, t_n \rightarrow_R t_0$  implies that there is a substitution  $\sigma$  such that  $\sigma(A_i) = t_i$  for all  $i$ .

For any  $0 \leq i \leq n$  we have

$$\|t_i\| = \|\sigma(A_i)\| = \|A_i\| + \sum_{j=1}^m (\#_{v_j}(A_i) \times (\|\sigma(v_j)\| - 1)),$$

$\text{Var}(R)$  is partitioned into two subsets  $\mathcal{V}_{arg}$  and  $\mathcal{V}_{pr}$ :

$$\mathcal{V}_{arg} = \{v \in \text{Var}(R) \mid v \text{ is in argument position in the left hand side of } R\},$$

and  $\mathcal{V}_{pr} = \text{Var}(R) - \mathcal{V}_{arg}$ . Let  $\mathcal{V}_{arg} = \{v'_1, \dots, v'_a\}$  and  $\mathcal{V}_{pr} = \{v''_1, \dots, v''_p\}$ . For any  $0 \leq i \leq n$

$$\|t_i\| = \|A_i\| + \sum_{j=1}^a (\#_{v'_j}(A_i) \times (\|\sigma(v'_j)\| - 1)) + \sum_{j=1}^p (\#_{v''_j}(A_i) \times (\|\sigma(v''_j)\| - 1))$$

Since  $t_i = \sigma(A_i)$  is a flat type, for all  $j$ ,  $1 \leq j \leq a$ ,  $\sigma(v'_j) \in Pr$ . Therefore  $\|\sigma(v'_j)\| = 1$ , which gives

$$\sum_{j=1}^a (\#_{v'_j}(A_i) \times (\|\sigma(v'_j)\| - 1)) = 0,$$

then  $\|t_i\| = \|A_i\| + \sum_{j=1}^p (\#_{v''_j}(A_i) \times (\|\sigma(v''_j)\| - 1))$

Since any  $v_j'' \in \mathcal{V}_{pr}$  is not in argument position in the left hand side of  $R$  and  $R$  is a flat rule,  $v_j''$  verifies the condition  $\#_{v_j''}(A_i) \leq \#_{v_j''}(A_0)$  for all  $1 \leq i \leq n$ . Thus for all  $i \geq 1$ :

$$\begin{aligned} \|t_i\| &= \|A_i\| + \sum_{j=1}^p (\#_{v_j''}(A_i) \times (\|\sigma(v_j'')\| - 1)) \\ &\leq \|A_i\| + \sum_{j=1}^p (\#_{v_j''}(A_0) \times (\|\sigma(v_j'')\| - 1)) \\ &\leq M_R + \|A_0\| + \sum_{j=1}^p (\#_{v_j''}(A_0) \times (\|\sigma(v_j'')\| - 1)) \\ &\leq M_R + \|t_0\| \end{aligned}$$

□

**Proposition 6.** *Let  $G$  be a flat  $\mathcal{R}$ -grammar and  $T$  an  $\mathcal{R}$ -structure. If  $\langle t_1 \dots t_n, t_0 \rangle$  is an instance of  $T$  for  $G$ , then for all  $i$ ,  $1 \leq i \leq n$ :*

$$\|t_i\| \leq h(T) \times M_{\mathcal{R}} + \|t_0\|, \text{ where } M_{\mathcal{R}} = \max(\{ M_R \mid R \in \mathcal{R} \}).$$

*Proof.* We show by induction on  $h = h(T)$  that  $t_0 \in FT(Pr)$  and  $\|t_i\| \leq h \times M_{\mathcal{R}} + \|t_0\|$ :

- $h = 0$ .  $T = w \in \Sigma$ , therefore  $n = 1$  and  $t_0 = t_1 \in FT(Pr)$  because  $w \triangleright t_1$  and  $G$  is a flat grammar.
- $h > 0$ . Suppose the property holds for any  $h' < h$ . Let  $T = [R](T_1, \dots, T_m)$ , and let  $\langle \alpha_1, u_1 \rangle, \dots, \langle \alpha_m, u_m \rangle$  be instances of  $T_1, \dots, T_m$  such that  $t_1 \dots t_n = \alpha_1 \bullet \dots \bullet \alpha_m$  and  $u_1, \dots, u_m \rightarrow_R t_0$ . By induction hypothesis, the property holds for any  $T_i$ :  $u_i \in FT(Pr)$  and for all  $t_i \in \alpha_j$  we have  $\|t_i\| \leq h_j \times M_{\mathcal{R}} + \|u_j\|$ , with  $h_j \leq h'$ . From proposition 4  $t_0 \in FT(Pr)$ , and from proposition 5  $\|u_j\| \leq \|t_0\| + M_R$ , then  $\|t_i\| \leq h_j \times M_{\mathcal{R}} + \|t_0\| + M_R$ . Since  $h_j \leq h - 1$  and  $M_R \leq M_{\mathcal{R}}$ , we obtain  $\|t_i\| \leq (h - 1) \times M_{\mathcal{R}} + \|t_0\| + M_{\mathcal{R}}$ , that is  $\|t_i\| \leq h \times M_{\mathcal{R}} + \|t_0\|$ .

□

**Corollary 1.** *Let  $D$  be a finite set of  $\mathcal{R}$ -structures and  $G$  a flat  $\mathcal{R}$ -grammar. If  $G$  is reduced with respect to  $D$  then every type  $t \in Lex(G)$  verifies  $\|t\| \leq H_D \times M_{\mathcal{R}} + \|s\|$ , where  $H_D = \max(\{ h(T) \mid T \in D \})$ .*

*Proof.*  $w \triangleright_G t$  and  $G$  is reduced with respect to  $D$ , so there exists an  $\mathcal{R}$ -structure  $T \in D$  and an instance  $\langle \alpha, s \rangle$  of  $T$  for  $G$  such that  $t \in \alpha$  (otherwise it would be possible to remove the rule  $w \triangleright_G t$  from  $G$ , and  $G$  would not be reduced). By proposition 6,  $\|t\| \leq h(T) \times M_{\mathcal{R}} + \|s\| \leq H_D \times M_{\mathcal{R}} + \|s\|$ . □

**Proposition 7.** *Let  $\mathcal{R}$  be a finite set of flat rules, and  $\mathcal{E}_{\mathcal{R}}$  the set of all flat  $\mathcal{R}$ -grammars lexical rules. The concept defining framework  $\langle S\mathcal{L}_{\mathcal{R}}, \mathcal{E}_{\mathcal{R}}, SL \rangle$  has bounded finite thickness.*

*Proof.*  $G \subseteq G'$  implies that  $SL(G) \subseteq SL(G')$  from proposition 1. Let  $G \subseteq \mathcal{E}_{\mathcal{R}}$  be a grammar reduced with respect to a finite set  $D \subseteq \mathcal{SL}_{\mathcal{R}}$ , with  $\#G \leq n$ . Corollary 1 shows that the size of each type in  $Lex(G)$  is bounded by a constant  $H_D \times M_{\mathcal{R}} + \|s\|$ . There must be only finitely many pairwise inequivalent such grammars, because the number of rules in  $G$  is also bounded. Since two equivalent grammars have the same language (from proposition 3), we obtain that the set  $\{ SL(G) \mid G \subseteq \mathcal{E}_{\mathcal{R}} \text{ is reduced with respect to } D \text{ and } \#G \leq n \}$  is finite for any set  $D$  and any  $n \geq 0$ .  $\square$

*Remark:* contrary to the case of simple AB grammars, where any functor-argument structure is compatible with at least one grammar, there may be no  $\mathcal{R}$ -grammar corresponding to a given  $\mathcal{R}$ -structure in  $\mathcal{SL}_{\mathcal{R}}$ . Thus given a set  $D$  of  $\mathcal{R}$ -structures it is possible that there is no grammar reduced with respect to  $D$ . Of course this does not contradict the previous result, since an empty set is clearly finite.

Since Shinohara has shown that bounded finite thickness implies that the class of languages definable by grammars with at most  $k$  rules has finite elasticity, the following corollaries are obtained easily with proposition 2:

**Corollary 2.** *Given any set  $\mathcal{R}$  of flat rules and any  $k \geq 0$ , the class of  $\mathcal{R}$ -structure languages definable by  $k$ -valued flat  $\mathcal{R}$ -grammars is learnable.*

**Corollary 3.** *Let  $\mathcal{R}$  be a set of flat rules such that any rule  $R = A_1, \dots, A_n \rightarrow A_0$  in  $\mathcal{R}$  verifies  $n \geq 2$ . For any  $k \geq 0$ , the class of string languages definable by  $k$ -valued flat  $\mathcal{R}$ -grammars is learnable.*

*Proof.* Suppose every rule  $A_1, \dots, A_n \rightarrow A_0$  in  $\mathcal{R}$  verify  $n \geq 2$ . We show that there is a finite-valued relation between the classes of  $\mathcal{R}$ -structure languages and of string languages definable by  $k$ -valued flat  $\mathcal{R}$ -grammars. Let  $G$  be a  $k$ -valued flat grammar. For every  $w \in L(G)$  the height  $h$  of the corresponding  $\mathcal{R}$ -structure must verify  $h < |w|$ , because  $n \geq 2$  for every rule in  $\mathcal{R}$ . So there is only a finite number of  $\mathcal{R}$ -structures corresponding to a string  $w$ . Using the property (shown by Kanazawa in [1]) that finite elasticity can be extended through a finite-valued relation, the class of string languages definable by  $k$ -valued flat grammars has also finite elasticity.  $\square$

## 4 Application to some Classes of $\mathcal{R}$ -grammars

In this section we propose to apply these learnability results to several formalisms, which are more or less close to the general framework of categorial grammars. Some of these positive results have already been proved separately. Even so, the fact that they can be deduced directly in our framework is interesting because it emphasizes the fact that their learnability is due to some common properties. Thus learnability of flat general combinatory grammars is also to some extent a generalization of different ways to prove learnability.

#### 4.1 Flat Classical Categorical Grammars and Extensions

Kanazawa has explored learnability of classical categorical grammars in [1]. In particular, he has shown that  $k$ -valued AB grammars have finite elasticity, thus are learnable from strings. We have seen in example 3 that AB grammars rules verify the conditions of flat universal rules, therefore it is possible to show learnability of  $k$ -valued flat AB grammars using corollaries 2 and 3 (from structures as well as for strings).

Since all AB grammars are not flat, this result is included in the learnability result obtained by Kanazawa (in which the definition of types is the usual one, which does not require that types be flat). Nevertheless, any AB grammar can be transformed into a flat AB grammar as it is shown in [10]. In this article, the authors obtained a slightly different learnability result concerning also AB grammars, by using also flat types (called *compact types* in their article). They show that there exists a flat AB grammar for each AB grammar, thus proving that the classes of languages are equivalent.<sup>6</sup>

**Extensions.** The interest in the framework of general combinatory grammars is that it becomes possible to add specific rules to the usual AB grammars system. For example, it is interesting in the dependency grammars viewpoint (see for example [12]) to add the new operators  $/^*$  (iterative types),  $/^+$  (repetitive types),  $/^?$  (optional types). The argument positions are defined in the same way as for AB grammars rules, that is  $arg_{/^*}(1) = arg_{/^+}(1) = arg_{/^?}(1) = 0$  and  $arg_{/^*}(2) = arg_{/^+}(2) = arg_{/^?}(2) = 1$ . These operators are used with the following rules:

$$\begin{array}{lll}
 R_1^* : & A/^*B B \rightarrow A/^*B & Var(R_1^*) = \{A, B\} \\
 R_2^* : & A/^*B \rightarrow A & Var(R_2^*) = \{A, B\} \\
 R_1^+ : & A/^+B B \rightarrow A/^+B & Var(R_1^+) = \{A, B\} \\
 R_2^+ : & A/^+B \rightarrow A & Var(R_2^+) = \{A, B\} \\
 R_1^? : & A/^?B B \rightarrow A & Var(R_1^?) = \{A, B\} \\
 R_2^? : & A/^?B \rightarrow A & Var(R_2^?) = \{A, B\}
 \end{array}$$

*Remark:* Symmetrical rules have to be defined for symmetrical operators on the left:  $\backslash^*$ ,  $\backslash^+$ ,  $\backslash^?$ .

Like AB grammars rules these rules are flat (the conditions of definition 14 are verified), therefore  $k$ -valued flat AB grammars with all these rules are also learnable from structures. However this class is not necessarily learnable from strings, because the rules  $R_2^*$  and  $R_2^?$  do not fulfill the condition defined in corollary 3: there must be at least two types in the left hand side. Actually, among these operators the only one that can be added to AB grammars languages without losing learnability from strings is the repetitive operator  $/^+$  (see [12] for details about this point).

<sup>6</sup> Such a proof can also be achieved using the classical transformation between AB grammars and context-free grammars given in [11]: if the AB grammar is converted into a CFG grammar and re-converted into an AB grammar, then the latter is flat.

## 4.2 Steedman’s Combinatory Categorical Grammars

Costa Florêncio has shown in [6] a sufficient condition for finite elasticity of any class of  $k$ -valued  $\mathcal{R}$ -grammars. The condition that he provides is based on the reduction of the class of languages to the class of  $k$ -valued AB grammars languages (which has finite elasticity: shown in [1]), using the fact that a finite-valued relation exists between the two classes. His criterion differs from ours: the method used implies that it must be possible to transform the set of universal rules into a set of rules which are very similar to AB grammars rules. But his criterion also has the advantage that it does not put any restriction on the form of the types, contrary to our condition (types must be flat). Actually the two methods lead to two different learnability results, each result allowing to learn classes that the other does not allow.

Costa Florêncio illustrates his learnability result with Steedman’s Combinatory Categorical Grammars rules (see for example [7] about CCG): he shows that that the language generated by a subset of CCG rules is learnable. However one of the usual CCG rules, namely the composition rule, can not be included in this subset, whereas it can be used in our framework:<sup>7</sup>

$$\begin{array}{lll}
 >B \text{ (Forward Composition)} & A/B \ B/C \rightarrow A/C & \text{Var}( >B) = \{A, B, C\} \\
 <B \text{ (Backward Composition)} & C \setminus B \ B \setminus A \rightarrow C \setminus A & \text{Var}( <B) = \{A, B, C\} \\
 >S \text{ (Forward Substitution)} & (A/B)/C \ B/C \rightarrow A/C & \text{Var}( >S) = \{A, B, C\} \\
 <S \text{ (Backward Substitution)} & C \setminus B \ C \setminus (B \setminus A) \rightarrow C \setminus A & \text{Var}( <S) = \{A, B, C\}
 \end{array}$$

One can see that these rules verify the conditions of flat universal rules. In particular, it is worth noting that type  $B$  in the composition rule  $>B$  is in argument position in the left hand side (in  $A/B$ ), even if there is also an occurrence of  $B$  in  $B/C$ . Nevertheless, the fact that this rule can be used in a learnable class of grammars within our framework should not hide the fact that it is usable only with flat types.

To provide a complete view of learnability of CCG rules, a word must be said about the type raising rules:

$$\begin{array}{lll}
 >T : & A \rightarrow B/(A \setminus B) & \text{Var}( >T) = \{A, B\} \\
 <T : & A \rightarrow (B/A) \setminus B & \text{Var}( <T) = \{A, B\}
 \end{array}$$

These rules do not fulfill the conditions of “learnable rules” neither in Costa Florêncio’s framework nor in ours (these rules are not flat because  $B$  does not appear in the left hand side). However types that can be used in these rules are restricted to a finite set of categories, so it is possible to “simulate” these rules directly in the lexicon [7]. This solution permits that such class of languages also have finite elasticity.

In a totally different approach, Hockenmaier has explored the acquisition of a combinatory categorial grammars lexicon in a practical viewpoint, in order to build a wide-coverage parser for English [13].

<sup>7</sup> *Remark:* traditionally, types using the  $\setminus$  operator in CCG are written *Functor*  $\setminus$  *Argument*, whereas the notation *Argument*  $\setminus$  *Functor* is used in classical AB grammars. We keep this latter notation here for consistency.

### 4.3 Categorical Link Grammars

Link grammars are defined by Sleator and Temperley in [14]. This is a rather simple formalism which is able to represent in a reliable way natural languages. This can be seen in the modelization that the authors provided for English in this system: their grammar deals with most of the linguistic phenomena in English, as it can be verified using their link grammar parser [15].

Béchet has shown in [16] that  $k$ -valued link grammars are learnable from strings, using also Shinohara's property of bounded finite thickness. It is shown in [17] that basic link grammars rules are equivalent to the following set of  $\mathcal{R}$ -grammars rules:

$$\begin{aligned} R_l : \quad & d(L, \text{cons}(c, R)) \quad d(\text{cons}(c, \text{nil}), \text{nil}) \rightarrow d(L, R) \quad \text{Var}(R_l) = \{c, L, R\} \\ R_r : \quad & d(\text{nil}, \text{cons}(c, \text{nil})) \quad d(\text{cons}(c, L), R) \rightarrow d(L, R) \quad \text{Var}(R_r) = \{c, L, R\} \end{aligned}$$

With these rules, a sequence of words is a correct sentence for the grammar if there is a type for each word such that the sequence of types can be reduced into the special type  $d(\text{nil}, \text{nil})$ .

*Example 4.* Let define a categorical link grammar  $G$  with the following lexicon:<sup>8</sup>

$$\begin{aligned} \text{a, the} & \triangleright d([], [D]) \\ \text{cat, snake} & \triangleright d([D], [S]), d([O, D], []) \\ \text{chased} & \triangleright d([S], [O]) \end{aligned}$$

The following derivation shows that the sentence “*the cat chased a snake*” is correct for  $G$ :

$$\begin{aligned} & \text{the} \quad \text{cat} \quad \text{chased} \quad \text{a} \quad \text{snake} \\ & d([], [D]), d([D], [S]), d([S], [O]), d([], [D]), d([D, O], []) \\ \Rightarrow & \quad d([], [S]), \quad d([S], [O]), d([], [D]), d([D, O], []) \\ \Rightarrow & \quad d([], [S]), \quad d([S], [O]), \quad d([O], []) \\ \Rightarrow & \quad d([], [S]), \quad d([S], []) \\ \Rightarrow & \quad d([], []) \end{aligned}$$

This system is called *Categorical Link Grammars (CLG)* (see [17] for more details). Since the formalism of link grammars is (at first sight) very different from categorical grammars, this equivalence permits to include the learnability result for link grammars obtained by Béchet in the more general framework of general combinatory grammars.

Let consider that the set of operators is  $\{d(2), \text{cons}(2), \text{nil}(0)\}$ , with  $\text{arg}_d(1) = \text{arg}_d(2) = 0$ ,  $\text{arg}_{\text{cons}}(1) = 1$  and  $\text{arg}_{\text{cons}}(2) = 0$ . Clearly rules  $R_l$  and  $R_r$  are flat, because  $L$  and  $R$ , which are the only variables that are not in argument position in the left hand side, do not appear in argument position in the right hand side. Thus it is possible to apply corollaries 2 and 3 to conclude that  $k$ -valued flat categorical link grammars are learnable from structures and from strings. Since the original definition of link grammars includes only flat types, we obtain here the same result as Béchet in [16].

<sup>8</sup> For a better readability, the notation  $[c_1, c_2, \dots, c_n]$  for connectors lists is used here instead of  $\text{cons}(c_1, \text{cons}(c_2, \dots, \text{cons}(c_n, \text{nil}) \dots))$ .

## 5 Conclusion

In this study, we did not show that a new class of languages is learnable. But we have shown that our result includes (partially or totally) several previous learnability results. Actually, the main interest in this result is that it is focused on the way languages are expressed with a grammatical formalism: the examples show that the framework of general combinatory grammars permits to express very different formalisms through the set of universal rules, and that the condition of flat grammars is not so restrictive. In particular the example of link grammars shows that using other operators than the standard binary AB grammars operators is possible and useful.

It should also be emphasized that the criterion of flat grammars, which is a sufficient condition for learnability of  $k$ -valued grammars, is not an ad hoc “technical” condition deduced from the constraints of the learning framework: this criterion is suitable for learning from structures, and it can be easily tested with any class of  $\mathcal{R}$ -grammars. As a future work, it remains to see if this criterion can be extended to more complex types. In particular, flat types means that all types in argument positions must be atoms, and the consequence (which is the main point of the proof) is that it is possible to bound the size of all types. Therefore an interesting question would be to know if it is possible to relax this constraint (for example by bounding the order of the types) without losing this consequence.

## References

1. Kanazawa, M.: Learnable classes of categorial grammars. Cambridge University Press (1998)
2. Gold, E.: Language identification in the limit. *Information and control* **10** (1967) 447–474
3. Buszkowski, W., Penn, G.: Categorial grammars determined from linguistic data by unification. Technical Report TR-89-05, Department of Computer Science, University of Chicago (1989)
4. Wright, K.: Identification of unions of languages drawn from an identifiable class. In: *Proceedings of the Second Annual Workshop on Computational Learning Theory*, Morgan Kaufmann (1989) 328–333
5. Motoki, T., Shinohara, T., Wright, K.: The correct definition of finite elasticity: corrigendum to Identification of unions. In: *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, San Mateo, CA, Morgan Kaufmann (1991) 375
6. Costa Florêncio, C.: Combinatory categorial grammars and finite elasticity. In Hoste, V., Pauw, G.D., eds.: *Proceedings of the Eleventh Belgian-Dutch Conference on Machine Learning*, University of Antwerp (2001) 13–18
7. Steedman, M.: *The Syntactic Process*. The MIT Press, Cambridge, Massachusetts (2000)
8. Shinohara, T.: Inductive inference of monotonic formal systems from positive data. *New Generation Computing* **8** (1991) 371–384
9. Costa Florêncio, C.: *Learning categorial grammars*. PhD thesis, Utrecht University (2003)



10. Besombes, J., Marion, J.Y.: Learning reversible categorial grammars from structures. In: Proceedings of Categorial Grammars 2004, Montpellier, France. (2004) 148–163
11. Bar-Hillel, Y., Gaifman, C., Shamir, E.: On categorial and phrase structure grammars (1960)
12. Béchet, D., Dikovsky, A., Foret, A., Moreau, E.: On learning discontinuous dependencies from positive data. In: Proceedings of the 9th conference on Formal Grammar. (2004)
13. Hockenmaier, J.: Data and models for statistical parsing with Combinatory Categorial Grammar. PhD thesis, School of Informatics, The University of Edinburgh (2003)
14. Sleator, D.D.K., Temperley, D.: Parsing english with a link grammar. Technical Report CMU-CS-TR-91-126, Carnegie Mellon University, Pittsburgh, PA (1991)
15. Temperley, D., Sleator, D., Lafferty, J.: Link grammar. <http://hyper.link.cs.cmu.edu/link/> (1991)
16. Béchet, D.:  $k$ -valued link grammars are learnable from strings. In: Proceedings Formal Grammars 2003. (2003) 9–18
17. Moreau, E.: From link grammars to categorial grammars. In: Proceedings of Categorial Grammars 2004, Montpellier, France. (2004) 31–45