# Identification of Hyperedge-Replacement Graph Grammars

**C. Costa Florêncio** CHRIS.COSTAFLORENCIO@CD.KULEUVEN.BE
Department of Computer Science, K.U. Leuven, Celestijnenlaan 200A-2402, 3001 Heverlee, Belgium

**Keywords**: Hypergraph-replacement grammars, learning from structured data, identification in the limit

## Abstract

This paper is intended as a step towards a theoretically sound approach to learning from graphs, by applying ideas from the field of grammar induction. We present results concerning the learning of hypergraph grammars. We consider identification in the limit from both derivation trees and graphs.

We show that, by restricting the complexity of grammars, a learnable class of derivation languages is obtained. By exploiting a known invariance result this class is extended to a class that is learnable from hypergraphs.

## 1. Introduction

This paper presents a way of applying ideas from grammatical inference to learning graph languages. We show that, by imposing bounds on the number of occurrences of terminals in graph grammars, a class of derivation languages is obtained that has finite elasticity, which implies identifiability in the limit. We then use an existing invariance result for finite elasticity to prove that an extended class, as well as the corresponding class of hypergraph languages is learnable.

### 1.1. Graph Grammars

The study of graph grammars goes back to the late sixties. It was motivated by the desire to extend the theory of formal languages from strings and trees to graphs, as well as by the potential of applications in pattern recognition. Nowadays, graph grammars are used in numerous fields such as chemical compound analysis, CAD, and computational linguistics.

In (Costa Florêncio, 2008), the learnability of Node Label Controlled (NLC) graph grammars was studied. It was shown that a learnable class is obtained if a bound on the number of occurrences of each label is imposed on Boundary NLC graph grammars.

In some sense, this paper generalizes this work, by considering hypergraphs instead of graphs. However, in an important sense it is incomparable, since it is based on *hyperedge* replacement instead of node replacement. This essential difference yields entirely different learnable classes. No other previous work comparable to the present paper seems to exist. Kukluk et al., 2008 proposes a learning algorithm for *edge*-replacement grammars. However, the algorithm is heuristic and learnability issues are not addressed.

Jeltsch & Kreowski, 1991 give an algorithm that generates the set of HR grammars consistent with a given set of hypergraphs. This kind of approach is an interesting step towards a learning algorithm. However, without combining it with a selection strategy, and proving convergence for the resulting algorithm, this cannot be regarded as a learning algorithm in itself.

## 2. HR Grammars and HR Languages

Given the space constraints, we will proceed in a somewhat informal manner. For the formal definitions the reader is referred to (Habel, 1992).

A hyperedge is a generalization of an edge with *ordered* collections of incoming- and outgoing tentacles. These are attached to nodes in a manner specified by a source- and target function, respectively. A set of nodes together with a set of hyperedges such that each tentacle is connected to a node forms a hypergraph.

In HR grammars, during derivation, non-terminal labeled hyperedges are replaced by hypergraphs. Since hypergraphs lack information about how they are to be attached to source- and target nodes, objects are needed that are annotated with this information. These are called *multi-pointed hypergraphs*. In the re-

Presented at *ILP-MLG-SRL*, Leuven, Belgium, 2009.

mainder, we will just call these hypergraphs when this does not lead to confusion. A set of multi-pointed hypergraphs is called a *hypergraph language* over $C$ if it is *closed under isomorphism*. Hypergraphs that represent strings are called *string graphs*.

### 2.1. Derivations

Habel, 1992 defines derivation trees in terms of parallel derivations. This is not suited for our purposes because the trees contain production rules from the generating grammar as edge-labelings. In the context of learnability, such a notion of derivation is unrealistic. We therefore use our own notion, that rids derivation languages of the rules and non-terminals as they appear in the generating grammar. Informally, the nodes of a derivation tree represent the application of rules from a grammar, and are labeled with copies of the righthand-sides(rhses) of these rules. The labels of the leaves contain just terminal hyperedges. We assume an order over the non-terminal hyperedges in every label. This can be any lexicographical order.

Derivations trees can be decomposed, such decompositions are crucial to our proof of learnability. Intuitively, they 'break down' derivation trees into the productions they are defined over. The decomposition of a leaf is simply a singleton containing a tuple of the hyperedge it replaces and its label, the decomposition of an internal node is a similar singleton union the decomposition of its daughters.

## 3. Learnability

We will apply the notion of learnability known as identification in the limit (Gold, 1967). A class of languages is considered learnable just if a computable function over sequences of input data exists that converges on a correct hypothesis after a finite number of presentations. It is assumed that all data is presented eventually.

### 3.1. Finite elasticity

Learnability is largely determined by topological properties of the class under consideration. One such property is the existence of an *infinite ascending chain* of languages. This means that there exists an infinite sequence $\langle L_n \rangle_{n\in\mathbb{N}}$ of languages in that class such that $L_0 \subset L_1 \ldots$. Having an infinite ascending chain is a necessary condition for having a *limit point*:

**Definition 1.** ***Existence of a limit point***
*A class $\mathcal{L}$ of languages is said to* have a limit point *if and only if it has an infinite ascending chain $\langle L_n \rangle_{n\in\mathbb{N}}$ and there exists another language $L \in \mathcal{L}$ such that $L = \bigcup_{n\in\mathbb{N}} L_n$. The language $L$ is called a* limit point *of $\mathcal{L}$.*

Having a limit point is a sufficient condition for not being identifiable in the limit, not even non-effectively.

Having an infinite ascending chain implies the weaker property known as *finite elasticity*:

**Definition 2.** *(Wright, 1989; Motoki et al., 1991)*
*A class $\mathcal{L}$ has* infinite elasticity *if there exists an infinite sequence $\langle s_n \rangle_{n\in\mathbb{N}}$ of sentences and an infinite sequence $\langle L_n \rangle_{n\in\mathbb{N}}$ of languages in $\mathcal{L}$ such that for all $n \in \mathbb{N}$, $s_n \notin L_n$, and $\{s_0, \ldots, s_n\} \subseteq L_{n+1}$.*

*A class $\mathcal{L}$ has* finite elasticity *if it does not have infinite elasticity.*

It has been shown that finite elasticity is a sufficient condition for learnability under two conditions:

**Theorem 1.** *(Wright, 1989) Let $\mathcal{G}$ be a class of grammars for a class of recursive languages, where $G \in \mathcal{G}$ is at least semi-decidable. If $\mathrm{L}(\mathcal{G})$ has finite elasticity, then $\mathcal{G}$ is identifiable in the limit.*

One way of proving learnability of a class is demonstrating it has finite elasticity, which also provides a simple way of extending the class by exploiting a closure property. Given two alphabets $\Sigma$ and $\Upsilon$, a relation $R \subseteq \Sigma^* \times \Upsilon^*$ is said to be *finite-valued* iff for every $s \in \Sigma^*$, there are at most finitely many $u \in \Upsilon^*$ such that $Rsu$. If $M$ is a language over $\Upsilon$, define a language $R^{-1}[M]$ over $\Sigma$ by $R^{-1}[M] = \{s \mid \exists u(Rsu \wedge u \in M)\}$.

**Theorem 2.** *(Kanazawa, 1994) Let $\mathcal{M}$ be a class of languages over $\Upsilon$ that has finite elasticity, and let $R \subseteq \Sigma^* \times \Upsilon^*$ be a finite-valued relation. Then $\mathcal{L} = \{R^{-1}[M] \mid M \in \mathcal{M}\}$ also has finite elasticity.*

This theorem is very useful for dealing with formalisms for which a clear and precise notion of derivation is defined: generally the relation between language element and possible derivation is finite-valued, and it is generally easier to prove finite elasticity of a class of derivation languages.

## 4. Learnable classes

The class of hyperedge-replacement grammars $\mathcal{G}_{HR}$ is obviously not identifiable in the limit, from sentential forms nor from derivations. It is easy to construct a chain of grammars $G_1, \ldots$ such that each $G_i$ in this chain generates exactly all string graphs of lengths 1 to $i$, thus the corresponding languages form an infinite ascending chain. Constructing a grammar that generates such string graphs without any bound on their length is trivial, and the language it generates is thus

a limit point for the class. Since we are interested in learnable subclasses of $\mathcal{G}_{HR}$, restrictions have to be imposed on the grammars. Let $k$ be an upper bound on the total number of occurrences of any given edge label from $T$ in the rhses, and let $\mathcal{G}_{k\text{-}HR}$ be the class of $\mathcal{G}_{HR}$ grammars that have $k$ as such a bound. If we restrict the class so that rules have at least one terminal hyperedge in the rhs,a bound on the number of rules is implied (recall that $T$ is finite). A bound on the number of rules implies a bound on the number of distinct non-terminal hyperedge labels in the grammar, since these occur at least once in the lhs and at least once in the rhs of rules of grammars in the class.

Assume that some class $\mathcal{G}$ that is a subclass of $DL(\mathcal{G}_{k\text{-}HR})$, for some fixed $k$, has infinite elasticity with infinite sequences of trees $T = t_1, \ldots$ and derivation languages $D = D_1, \ldots$, with corresponding grammars $G_1, \ldots$. We define the sequence of *sets of smallest grammars consistent with* $T$ $\mathbf{G}_1, \ldots$, such that $\mathbf{G}_{i+1} = \cup\{G | G = \sigma[\text{decomp}(t_1) \cup \ldots \text{decomp}(t_i)] \wedge G \in \mathcal{G}_{k\text{-}HR}\}$, with $\sigma$ an mgu. 'Smallest' is defined in terms of the number of productions of a grammar.

**Lemma 1.** *For every* $\mathbf{G}_i$ *there is a grammar* $G' \in \mathbf{G}_i$ *such that* $\exists\sigma.\sigma[G'] \subseteq G_i$*, thus for any tree* $t \notin DL(G_i)$*,* $t \notin DL(G')$*. Given that* $\exists G' \in \mathbf{G}_{i+1}.t_i \in DL(G')$*, and assuming that* $t_i$ *does not introduce new terminal labels, it follows that* $\forall G'' \in \mathbf{G}_{i+1}$*,* $\exists\sigma, G' \in \mathbf{G}_i.\sigma[G'] = G''$*, with* $\sigma$ *non-trivial.*

In other words, for such a tree, all grammars in $\mathbf{G}_{i+1}$ can be derived from $\mathbf{G}_i$ by applying substitutions. Note that any $\mathbf{G}_i$ is a set of finite cardinality.

**Proposition 1.** *For* $k = 1$*,* $DL(\mathcal{G}_{k\text{-}HR})$ *has finite elasticity.*

*Proof.* (Sketch) Assume that this class has infinite elasticity. Then $G_{i+1}$ must include just one rule for each such hypergraph $H_t$ corresponding to a node in $t_i$, and by the definition of finite elasticity, the same is true for each $G_j, j \geq i + 1$.

Given Lemma 1, for any tree $t_i$ in $T$, all grammars in $\mathbf{G}_{i+1}$ can be derived from grammars from the previous timestep by applying substitutions. Only a finite number of such substitutions exist, so from this it follows that the subsequence of trees $t_i, \ldots t_{i+j}$, where none of the trees introduce new terminals, is of finite length.

Assume that $t_i$ does introduce new terminals. There are only finitely many such trees in the sequence, and since the sequences inbetween them are finite, after some point $p$, all grammars in the sequence $G_p, \ldots$ can be obtained from a grammar $G' \in \mathbf{G}_p$ by applying a substitution. Just a finite number of such substitutions exist for each $G'$, so after $p$ only a finite number of different grammars occur. Each of these grammars can only occur a finite number of times in the sequence. Thus, the whole sequence $G_1, \ldots$, and thus the whole sequence $D_1, \ldots$, must be of finite length. □

For any grammar $G \in \mathcal{G}_{k\text{-}HR}$, let $R$ be the relation between the $x$th occurrence of terminal $l$ and some terminal $l_x$, i.e., we define an alphabet that contains $k$ copies of each terminal symbol. This way, we can obtain a grammar in $G \in \mathcal{G}_{k\text{-}HR}$, $k = 1$ for any grammar in $G \in \mathcal{G}_{k\text{-}HR}$ for any $k$. This relation is finite-valued, so Theorem 2 applies and finite elasticity for $k = 1$ generalizes to $k \geq 1$.

The relation between derivation trees and their yields is finite-valued, given the definition of derivation trees and the restriction that every rhs must contain at least one terminal hyperedge, so by Theorem 2 we obtain:

**Corollary 1.** *For any* $k$*,* $L(\mathcal{G}_{k\text{-}HR})$ *has finite elasticity and is thus learnable from positive data (hypergraphs) by a consistent and conservative learner.*

## References


Costa Florêncio, C. (2008). Learning node label controlled graph grammars: Extended abstract. *Proceedings of ICGI'08, 9th International Colloquium on Grammatical Inference, St Malo, Britanny, France* (pp. 286–288). Springer Verlag.

Gold, E. M. (1967). Language identification in the limit. *Information and Control*, *10*, 447–474.

Habel, A. (1992). *Hyperedge replacement: Grammars and languages.* Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Jeltsch, E., & Kreowski, H.-J. (1991). Grammatical inference based on hyperedge replacement. *Proc. International Workshop on Graph Grammars and Their Application to Computer Science* (pp. 461–474). Springer.

Kanazawa, M. (1994). *A note on language classes with finite elasticity* (Technical Report CS-R9471). CWI, Amsterdam.

Kukluk, J. P., Holder, L. B., & Cook, D. J. (2008). Inference of edge replacement graph grammars. *International Journal on Artificial Intelligence Tools*, *17*, 539–554.

Motoki, T., Shinohara, T., & Wright, K. (1991). The correct definition of finite elasticity: Corrigendum to identification of unions. *The Fourth Workshop on Computational Learning Theory.* San Mateo, Calif.: Morgan Kaufmann.

Wright, K. (1989). Identification of unions of languages drawn from an identifiable class. *The 1989 Workshop on Computational Learning Theory* (pp. 328–333). San Mateo, Calif.: Morgan Kaufmann.