

# Recognizable sets of graphs, hypergraphs and relational structures : a survey

Bruno Courcelle

August 21, 2004

## Abstract

New results on the recognizability of sets of finite graphs, hypergraphs and relational structures are presented. The general framework of this research which associates tightly algebraic notions (equational and recognizable sets) and Monadic Second-Order logic (for defining sets and transformations of graphs, hypergraphs and relational structures) is reviewed. The lecture [3] is based on two submitted but nevertheless available articles [1,4] ; the present text is an informal overview. The numerous definitions and results can be found in the two articles.

## 1 Introduction

The description of sets of finite words (called *languages*) and of their transformations (called *transductions*) was the original goal of the Theory of Formal Languages. This theory now extends its scope to infinite words, to finite and infinite trees (modelling finite and infinite algebraic terms), and more recently, to finite and infinite graphs, hypergraphs and related structures like partial orders and traces. Unless otherwise specified, we will use “graph” as a generic term covering directed and undirected, labelled and unlabelled graphs and hypergraphs. All these objects are conveniently handled as *relational structures*, *i.e.*, as logical structures with no function symbol except possibly nullary ones.

In addition to classical tools like grammars, automata and transducers, First-Order and Monadic Second-Order logic have proved to be useful to describe sets of words and trees. For dealing with graphs, logic is also essential, not only for defining sets of graphs but also for defining graph transformations. However, the variety of types of graphs, and consequently of operations on them (that generalize the concatenation of words), makes it necessary to use also some unifying concepts provided by Universal Algebra.

The basic notion of a context-free language can be characterized in terms of least solutions of equation systems (equivalent to context-free grammars). That of a regular language can be characterized in terms of congruences with finitely many classes (equivalent to finite deterministic automata). These algebraic definitions are interesting in that they apply to every algebra. They are thus appropriate for dealing with various types of sets of graphs, but they are useful for other reasons. Context-free sets of graphs can be defined as *equational sets*, i.e. as components of least solutions of systems of recursive set equations, in a much easier way than in terms of graph rewriting sequences. Since there is no good notion of graph automaton except in very particular cases, the notion of a finite congruence is the only way to obtain a workable generalization of regularity to sets of graphs : a *recognizable set* is a set which is saturated for a congruence with finitely many classes of each sort (we use many-sorted algebras).

An equational set of graphs can be specified in a readable way by an equation system. But specifying a recognizable set of graphs by a congruence is no more convenient than specifying in this way a regular language, because even in simple cases, congruences tend to have many classes. *Monadic Second-Order logic* is here especially useful as a specification language. Since a graph is nothing but a relational structure, every closed formula, either first-order or second-order specifies a set of graphs, namely the set of its finite models. (We only consider finite objects in this survey). Furthermore most graph properties can be expressed easily by logical formulas. And every set of graphs characterized by a Monadic Second-Order formula, i.e. a formula where quantified variables denote individual elements (typically vertices, but also edges) or sets thereof is recognizable. In particular, basic classes of graphs like trees, connected graphs, planar graphs are monadic second-order definable.

Hence Monadic Second-Order logic (MS logic in short) is an appropriate language for specifying sets of graphs. It can be seen as an alternative to the non-existing notion of graph automaton. Furthermore, it can be used to specify graph transformations, called *MS transductions* that are as useful for studying sets of graphs as are rational transductions for languages. Applications to the construction of linear algorithms for hard (NP complete) problems restricted to certain equational sets of graphs substantiate the claim of usefulness of Monadic Second-Order logic.

We hope to convince the reader that the algebraic notions of a recognizable and of an equational set of graphs on the one hand, and the logical notions based on MS logic on the other build a coherent and robust framework for extending to graphs the notions and results of Formal Language Theory. The following table summarizes the main results. (MST means MS transduction).

Algebraic notions	Algebraic characterizations	Logical characterizations	Closure properties
EQ	Equation systems, val(REC(Terms))	MST(Trees)	Union, $\cap$ Rec, Homomorphisms, MST
REC	Finite Congruences	MS-definable ( $\subset$ REC)	Boolean operations, Inverse homomorphisms, Inverse MST

## 2 Notions from Universal Algebra

For dealing with graphs one needs to use *many-sorted algebras*, with countably many sorts. A sort is here a finite set of labels (which is a subset of a fixed countable set of labels). These labels are used to specify in a nonambiguous way “graph concatenation” operations. Since the combinatorial structure of graphs is much more complicated than that of words, one cannot limit oneself to concatenation operations (or graph construction operations) based on a uniformly bounded number of labels. (One can actually generate all finite graphs from 6 operations with 2 sorts in a somewhat artificial way but the interesting algorithmic results discussed below do not work for these operations. See [4].)

In a many-sorted algebra, an *equational set* is a set of elements of the same sort that is a component of the least solution of a (finite) system of recursive set equations. An example of such a system is :

$$\{ X = f(X, Y) \cup \{a\} ; Y = g(X, Y, Y) \cup f(Y, Y) \cup \{b\} \},$$

where  $X$  and  $Y$  denote sets and  $f$  and  $g$  denote the set extensions of functions belonging to the signature. A *recognizable set* is a set of elements of the same sort, that is a union of classes for a finite congruence, i.e., a congruence such that any two equivalent elements are of the same sort and which has finitely many classes of each sort. (See Courcelle [2]). These two notions depend on the signature. We refer to  $F$ -equational and to  $F$ -recognizable sets, forming the classes  $EQ(F)$  and  $REC(F)$  respectively, when we need to specify the signature  $F$ .

In some cases, the notion of a recognizable set somehow degenerates. If the signature is “poor” (for example if it consists only of constants and a finite set of unary operations closed under composition), then every set is recognizable.

If on the opposite it is “too rich” then the only recognizable sets are the empty set and the set of all elements of a same sort. This is the case of the set of positive integers equipped with the successor and the predecessor function.

In every algebra, even having an infinite signature, the following properties hold :

**Property 1 :** The class of recognizable sets is closed under union, intersection, difference and inverse homomorphisms. In particular, for every finite subsignature  $F$  of the considered signature, the set of (finite)  $F$ -terms, the value of which belongs to a recognizable set is recognizable, hence is definable by a finite deterministic tree-automaton.

**Property 2 :** The class of equational sets is closed under union, homomorphisms, and intersection with recognizable sets ; it is closed under the operations of the signature.

**Property 3 :** One can decide the emptiness of an equational set given by a system of equations.

**Property 4 :** A set is equational iff it is the set of values of a recognizable set of terms over a finite subset of the signature.

For two signatures  $F$  and  $K$  on a same set  $M$ , such that  $F$  is a subsignature of  $K$ , we get immediately from the definitions

**Property 5 :** Every  $F$ -equational set is  $K$ -equational. Every  $K$ -recognizable set is  $F$ -recognizable.

We will say that the signatures  $F$  and  $K$  are *equivalent* if the corresponding classes of equational and recognizable sets are the same. If  $K$  is  $F$  enriched with operations that are defined by finite  $F$ -terms, then  $F$  and  $K$  are equivalent. But there are examples of equivalent signatures not of this type. Consider the set of words over a finite alphabet, where  $F$  consists of concatenation, empty word and letters, and  $K$  is the same together with the mirror image operation. Then the signatures  $F$  and  $K$  are equivalent, but the mirror image is not expressible as a composition of operations of  $F$ .

We will be interested by the comparison of various signatures of graph operations. We will see that no more than three signatures have to be considered, each of them having many equivalent variants. This indicates the robustness of this algebraic approach.

*Algorithmic applications* of recognizability can be described from the above algebraic properties, using the fact that an MS definable set of graphs is recognizable. Let us consider an equational set  $L$  given by an equation system. Let

$K$  be a recognizable set which is “effectively given”, for instance by means of a homomorphism into an algebra each domain of which is finite, or by an MS formula  $\varphi$ . By Property 2, the set  $L \cap K$  is an equational set for which one can construct an equation system. It follows from Property 3 that one can decide the emptiness of  $L \cap K$ . By applying this to the case of an equational set of graphs  $L$  and a set  $K$  defined as the set of finite models of a closed MS formula  $\varphi$ , one obtains that one can decide whether there exists in  $L$  a graph satisfying  $\varphi$ . This decision problem, called the *Monadic Second-Order satisfiability problem for  $L$*  is non trivial : it is undecidable  $L$  is the (non-equational) set of all finite graphs, and even for first-order formulas  $\varphi$ .

By using the second assertion of Property 1 one can decide in linear time if an  $F$ -term (where  $F$  is finite subsignature of the considered global signature) has for value an object belonging to the recognizable set  $K$  defined by an MS formula  $\varphi$ . It follows that if a graph  $G$  in  $L$  is given by a term over the finitely many operations that occur in the defining system for  $L$  (say a derivation tree of  $G$  relative to the context-free graph grammar represented by the equation system) then one can decide in time proportional to the size of this term whether  $G$  satisfies the MS formula  $\varphi$ . This applies to *NP complete problems* expressible by MS formulas, like 3-vertex colorability.

### 3 Graph operations

The use of algebraic notions is based the definition of the operations on graphs that form the signature. There are actually two main (non-equivalent) signatures of interest, that we call HR and VR because the corresponding equational sets have been defined independently and previousy by “Hyperedge Replacement” Context-free Graph Grammars and, respectively, by context-free graph grammars based on “Vertex Replacement”. (See the book edited by G. Rozenberg on graph grammars [7]). They are robust in the sense that many variants of the definitions yield the same equational sets. Furthermore, the corresponding classes of equational sets are closed under MS transductions. This is analogue to the closure of the class of context-free languages under rational transductions.

The HR operations deal with graphs and hypergraphs having distinguished vertices called *sources* designated by *labels*. (There is only one source for each label.) The HR operations are the *parallel composition* of two graphs or hypergraphs (one takes the disjoint union of the two and one fuses the sources with same labels), operations that *change the labels* of sources, and operations that “*forget*” sources (forgetting the  $a$ -source means that the vertex designated by  $a$  is no longer distinguished, but is made “ordinary”). Basic graphs or hypergraphs are those with a single edge or hyperedge (possibly with loops), and isolated vertices.

The VR operations deal with graphs (not with hypergraphs) with labelled vertices. Each vertex has one and only one label but several vertices may have the same label. The VR operations are the *disjoint union* of two labelled graphs, some operations that *modify labels* in a uniform way (every vertex labelled by  $p$  is relabelled by  $q$ ) and operations that *add edges* between every vertex labelled by  $p$  and every vertex labelled by  $q$ , for fixed vertex labels  $p$  and  $q$ . (These last operations produce graphs having complete bipartite subgraphs  $K_{n,m}$  (with  $n + m$  vertices) where  $n$  is the number of  $p$ -labelled vertices and  $m$  is the number of  $q$ -labelled vertices. Since  $n$  and  $m$  are not bounded, such graphs have unbounded tree-width and the VR operations cannot be replaced by compositions of HR-operations.) The basic graphs are those with a single vertex.

In both cases, every graph (or hypergraph in the case of HR) can be generated by these operations by using one label for each vertex. By bounding the allowed number of labels, one obtains particular classes of graphs and hypergraphs forming two infinite hierarchies. We obtain also the families of *HR-equational*, *VR-equational*, *HR-recognizable* and *VR-recognizable sets* of graphs and hypergraphs. They correspond to the families of context-free and regular languages, but we have two notions in the case of graphs.

There are other significant differences with the case of words. The set of all words on a fixed finite alphabet is context-free, whereas the set of all finite graphs (say, of simple undirected unlabelled graphs to get a precise statement) is neither HR-equational nor VR-equational. This is due to the necessity of using infinitely many operations to generate all graphs (whereas the unique operation of concatenation suffices to generate all words). Since an equation system is by definition finite, none can define all graphs. From this observation, it follows that a set of graphs may be non-HR-equational for two reasons : either because it has unbounded tree-width (this the case of the set of finite planar graphs) or because it has an “irregular” internal structure (this is the case of the set of strings, the length of which is a prime number). Note however that classical non-context-free languages like  $a^n b^n c^n$  are equational as sets of vertex labelled graphs.

Classes of recognizable sets of graphs (here we discuss labelled, directed or undirected graphs, not hypergraphs) are associated with the signatures HR and VR. Again, due to the infiniteness of the signatures, some properties of recognizable sets of words do not extend to graphs. In particular there are uncountably many recognizable sets (every set of square grids is HR-recognizable as well as VR-recognizable ; see [4]).

There exist two complexity measures on graphs, called *tree-width* and *clique-width* defined respectively as the minimum number of labels necessary to construct the considered graphs or hypergraphs with HR and VR operations. A set of graphs has bounded tree-width (resp. bounded clique-width) iff it is a subset of an HR-equational (resp. of a VR-equational) set. Tree-width has been

introduced independently of graph grammars by Robertson and Seymour, and this parameter is essential in the *theory of parametrized complexity* developed by Downey and Fellows [5]. Clique-width is also useful for constructing polynomial algorithms for hard problems for particular classes of graphs.

Up to now we have only presented two signatures HR and VR. The first one concerns graphs *stricto sensu* as well as hypergraphs, whereas the second concerns only graphs or slightly more generally, *binary relational structures* (i.e., structures with relations of arity 1 and 2). We will now consider general relational structure, which correspond exactly to *directed, ranked, hyperedge-labelled hypergraphs*, simply called *hypergraphs* in the sequel. The *arity* of a hypergraph is the maximal arity of the (relation) symbols labelling its hyperedges.

For dealing with them, we introduce a many-sorted signature STR. We fix a set of relation symbols with countably many relations of each arity. Every finite subset  $\Sigma$  of this set is a sort and the corresponding domain is the set  $STR(\Sigma)$  of finite  $\Sigma$ -structures. The operations are the disjoint union and all the unary operations that transform a structure into another one by means of quantifier-free conditions. The transformations performed by these operations can delete elements (for example, in a graph one may want to remove all vertices incident to no edge or loop), and/or redefine relations (for example, for defining the edge-complement of a graph). The unary VR operations are of this latter form. The HR operations also, if we denote “sources” by nullary function symbols. Technical details are omitted here.

Although there exist infinitely many quantifier-free formulas written with finitely many variables and relation symbols, there are only finitely many quantifier-free formula up to logical equivalence, and this equivalence is decidable. It follows that there are only finitely many quantifier-free operations  $STR(\Sigma) \longrightarrow STR(\Gamma)$  where  $\Sigma$  and  $\Gamma$  are finite sets of relations. We obtain the inclusions of signatures :

$$\text{HRg} \subseteq \text{VR} \subseteq \text{STR}$$

where HRg denotes the restriction of HR to graphs obtained by taking graphs and not hypergraphs as basic objects. Then follow the inclusions :

$$\text{EQ}(\text{HRg}) \subseteq \text{EQ}(\text{VR}) \subseteq \text{EQ}(\text{STR})$$

and the reverse inclusions for the corresponding classes of recognizable sets of graphs by Property 5.

The inclusion  $\text{EQ}(\text{HRg}) \subseteq \text{EQ}(\text{VR})$  is proper : the set of all cliques is VR-equational (easy to see ; cliques have clique-width 2) but not HR-equational because cliques have unbounded tree-width. If a set of graphs is VR-equational but is “without large complete bipartite subgraphs” which means that for some large enough  $n$ , no graph in  $L$  has a subgraph isomorphic to  $K_{n,n}$  then it is

HR-equational. The intuition about this result is the following : the operation in the VR signature which is not expressible in terms of HR operations is that which adds to a graph edges forming a complete bipartite subgraph. If  $L$  has no graph containing large  $K_{n,n}$ 's, this means that this operation is not used in a crucial way hence that each of its occurrence can be replaced by a composition of HR-operations.

A result proved in [4] establishes a similar result for recognizable sets of graphs. First the inclusion  $\text{REC}(\text{VR}) \subseteq \text{REC}(\text{HR})$  is proper because every set of cliques is HR-recognizable, but the set of cliques of size  $n$  such that  $n$  belongs to a set of positive integers which is not recognizable (like the set of prime numbers), is not VR-recognizable. Second, we have :

**Theorem 1** [4] : *If set of graphs without large complete bipartite subgraphs is HR-recognizable, then it is VR-recognizable.*

Hence the same combinatorial condition collapses simultaneously the two proper inclusion of  $\text{EQ}(\text{HR})$  in  $\text{EQ}(\text{VR})$  and of  $\text{REC}(\text{VR})$  in  $\text{REC}(\text{HR})$ . However, the proofs of the two results are different.

We now discuss the inclusion of VR in STR. We have the following :

**Theorem 2** [4] : *A set of graphs is VR-equational iff it is STR-equational, and it is VR-recognizable iff it is STR-recognizable.*

The first result is a direct consequence of characterizations of VR-equational and STR-equational sets in terms MS transductions in Theorem 7 below. The second one is proved in [4]. We only explain here the meanings of these results.

To generate graphs by means of VR-operations, one uses only structures over the following relation symbols : a single binary relation *edg* representing edges linking vertices, and an unbounded number of unary auxiliary relations for representing vertex labels. These auxiliary relations need not occur in the generated graphs. They are only useful at intermediate stages of the generation process to establish edges. Among the quantifier-free unary operations used in STR, are operations that may delete relations. This means that if one uses the operations of STR to generate graphs, some intermediate generated objects may be hypergraphs (represented by non-binary relational structures). One might think that because of this richer signature, the family  $\text{EQ}(\text{STR})$  would contain sets of graphs not in  $\text{EQ}(\text{VR})$ , but this not the case. The signature VR is “strong enough” to yield the same equational sets as the apparently more powerful signature STR, which uses domains (sets of hypergraphs) not in the VR algebra of graphs. Hence in order to generate graphs represented by relational structures with one relation of arity 2, the auxiliary relations may be limited



to arity 1. A similar statement can be given for recognizable sets : to establish that a set of graphs is STR-recognizable, it is enough to produce a congruence for the VR-operations, without having to extend it to a congruence on all the domains  $STR(\Sigma)$  for all  $\Sigma$ 's.

The above discussion showing that for generating graphs, one need not use auxiliary relations of arity more than 1 can be repeated for each maximal arity : for generating hypergraphs with hyperedges with at most  $n$  distinct vertices, one need not auxiliary hyperedges of arity more than  $n - 1$ . This is formally defined and proved in [1]. Letting  $STR_n$  denote the restriction of  $STR$  to relations (and the corresponding domains and operations) of arity at most  $n$  we get the following statement :

**Theorem 3** [1] : *A set of hypergraphs of arity at most  $n$  is  $STR_n$ -equational iff it is STR-equational. It is  $STR_n$ -recognizable iff it is STR-recognizable.*

Hence, with respect to sets of hypergraphs of maximal arity  $n$ , the full signature  $STR$  is equivalent to  $STR_n$ , its restriction to structures of arity at most  $n$ . Stronger formulations and their proofs can be found in [1].

## 4 Monadic Second-Order logic and graph properties

Sets of graphs can be specified either recursively, in terms of base graphs and application of operations : this is what yields an equation system. Sets of graphs are also frequently specified by *characteristic properties*. Considering a graph as a logical structure with relations representing adjacency or incidence makes it possible to formalize its properties by logical formulas.

*First-Order logic* can only express local properties like bounds on the degrees of vertices, hence is here of limited interest. But *Monadic Second-Order logic*, (*MS logic* in short) *i.e.*, the extension of First-Order logic with variables denoting subsets of the domains of the considered structures is quite powerful. It can express both coloring properties (for instance that a graph is 3-vertex colorable, an NP complete property), and path properties (like connectivity, existence of cycles, planarity via Kuratowski's theorem). But a property based on the existence of bijections, like the existence in a graph of a nontrivial automorphism, is provably not MS expressible.

A fundamental theorem says that :

**Theorem 4** : Every MS definable set of graphs is VR-recognizable. More generally, every MS-definable set of relational structures is STR-recognizable.

The algorithmic applications of recognizability reviewed in the first section are based on this result. One can actually improve Theorem 4 and its algorithmic consequences as follows. If instead of representing a graph by a relational structure, the domain of which is the set of vertices and which has a binary relation representing the edges, we use its incidence graph, *i.e.*, the relational structure the domain of which consists of vertices and edges, and equipped with an incidence relation, then, MS logic becomes more expressive because one can denote sets of edges by set variables. The same can be done for hypergraphs, and we denote by  $MS_2$  the use of MS logic with this representation of graphs or hypergraphs. That a graph has a Hamiltonian circuit is  $MS_2$  expressible but provably not MS expressible. Then we get the following result :

**Theorem 5** : Every  $MS_2$  definable set of graphs or hypergraphs is HR-recognizable.

If we compare Theorems 4 and 5 in the perspective of their algorithmic applications, we can see that Theorem 5 concerns more properties, namely the  $MS_2$  definable properties instead of the more restricted MS definable ones, but less families of graphs, namely those of bounded tree-width (generated by finite subsignatures of HR) instead of bounded clique-width (generated by finite subsignatures of VR).

## 5 Monadic Second-Order Transductions

Transformations of words and trees are based on *finite automata, equipped with output functions (sequential machines, tree-transducers)*, or on *rational expressions* and homomorphisms (*rational transductions* between languages). Since for graphs we have neither automata nor rational expressions, we must base *graph transformations* on another model. Monadic Second-Order logic offers the appropriate alternative.

An *MS transduction* :  $STR(\Sigma) \longrightarrow STR(\Gamma)$  is a partial multivalued function specified by a finite sequence of MS formulas, using (possibly) set variables called *parameters*, and forming its *definition scheme*. A structure  $S$  in  $STR(\Sigma)$  is transformed into a structure  $T$  in  $STR(\Gamma)$  as follows : one select values for the parameters that satisfy a formula, the first one in the definition scheme. Then one builds a structure  $S'$  consisting  $k$  disjoint “marked” copies of  $S$  ( $k$  is fixed in the definition scheme). The output structure  $T$  is defined inside  $S'$  by restricting the domain and by defining its  $\Gamma$ -relations from the  $\Sigma$ -relations in  $S'$  and the “marks”. These restrictions and definitions are done by MS formulas depending on the parameters. The transformation is multivalued because in general several choices of parameters can be made. An MS transduction defined by a parameterless definition scheme is a partial function. An important difference

with rational transductions is the fact that the inverse of an MS transduction is not always an MS transduction whereas that of a rational transduction is a rational transduction. With respect to the equational and recognizable sets, MS transductions behave like homomorphisms as we will see.

The fundamental property of MS transductions is, with the above notation:

**Theorem 6 :** The monadic second-order properties of the output structure  $T$  can be expressed by monadic second-order formulas in the input structure  $S$  in terms of the parameters used to define  $T$  from  $S$ .

It follows that the composition of two MS transductions is an MS transduction (a result much more important than the closure on inverse). It follows also that if a set of structures has a decidable monadic second-order satisfiability problem, then so has its image under an MS transduction.

The mapping from a term in  $T(HR)$  (the set of finite HR-terms),  $T(VR)$  or  $T(STR)$  to the corresponding graph or hypergraph is an MS transduction. It follows that an equational set of graphs is the image under an MS transduction of a recognizable set of terms, equivalently (we omit details) of the set of finite binary trees. This result have a very important converse :

**Theorem 7 :** If a set of graphs (resp. a set of hypergraphs) (resp. the set of incidence graphs of a set  $L$  of hypergraphs) is the image of the set of finite binary trees under an MS transduction, then this set is VR-equational (resp. is STR-equational) (resp. is HR-equational).

This theorem is somewhat similar to the one saying that the context-free languages are the images under rational transductions of the Dyck language, which is actually a coding of trees by words. An important consequence is the following :

**Theorem 8 :** The image of a VR-equational set (resp. of an STR-equational set) (resp. of an HR-equational set) under an MS transduction of appropriate type is VR-equational (resp. STR-equational) (resp. HR-equational).

"Appropriate" means that it produces graphs from graphs in the first case, and that it transforms hypergraphs through their incidence graphs in the third case. Hence, with respect to equational sets, MS transductions behave like homomorphisms. They do the same with respect to recognizable sets since we have :

**Theorem 9 [1] :** *The inverse image of an STR-recognizable set under an MS transduction is STR-recognizable.*

This result is not very surprising, because we know already from Theorem 6 that the inverse image of an MS definable set is MS definable, which yields

Theorem 9 for those particular recognizable sets that are MS definable. However it shows how algebraic and logical notions are tightly linked.

## 6 Open questions and research directions

We only mention a few questions related to the notions discussed in this overview.

1. *Which quantifier-free operations on relational structures preserve recognizability ?*

2. *Can one define a complexity measure on relational structures generalizing clique-width and that is linked to the signature  $STR$  or rather, to a subsignature equivalent to it like  $STR_n$  ?*

3. *How can one enrich the signature  $STR$  into a larger signature equivalent to it ?*

Concerning question 1 : Not all quantifier-free operations preserve recognizability : the operation that deletes a relation  $R$  of arity at least 2 does not. Concerning question 2, clique-width is a complexity measure linked to VR, which is a subsignature of  $STR$  equivalent to it for graphs. Answers to question 3 can be found in [1] but they do not close the question.

## 7 References

More references can be found in [1,4].

[1] A. Blumensath, B. Courcelle : Recognizability and hypergraph operations using local informations, July 2004,  
[http://www.labri.fr/Perso/~courcell/Textes/BC-Blumensath-submitted\(2004\).pdf](http://www.labri.fr/Perso/~courcell/Textes/BC-Blumensath-submitted(2004).pdf)

[2] B. Courcelle : Basic notions of Universal Algebra for Language Theory and Graph Grammars, Theoretical Computer Science 163 (1996) 1-54.

[3] B. Courcelle : Recognizable sets of graphs, hypergraphs and relational structures : a survey,  
<http://www.labri.fr/Perso/~courcell/Conferences/ExpoDLT04.pdf>, (slides).

- [4] B. Courcelle, P. Weil : The recognizability of sets of graphs is a robust property, December 2003,  
[http://www.labri.fr/Person/~courcell/Textes/BC-Weil-submitted\(2004\).pdf](http://www.labri.fr/Person/~courcell/Textes/BC-Weil-submitted(2004).pdf)
- [5] R. Downey, M. Fellows : Parametrized complexity, Springer, 1999.
- [6] J. Mezei, J. B. Wright, Algebraic automata and context-free sets, Information and Control 11 (1967) 3-29.
- [7] G. Rozenberg ed., Handbook of graph grammars and computing by graph transformations, Volume 1, World Scientific, 1997.