Graph Structure and Monadic Second-order Logic: Language Theoretical Aspects*

Bruno Courcelle

Université Bordeaux-1, LaBRI, CNRS
Institut Universitaire de France
351, Cours de la Libération
33405, Talence cedex, France
courcell@labri.fr

Abstract. *Graph structure* is a flexible concept covering many different types of graph properties. Hierarchical decompositions yielding the notions of *tree-width and clique-width*, expressed by terms written with appropriate graph operations and associated with *Monadic Second-order Logic* are important tools for the construction of Fixed-Parameter Tractable algorithms and also for the extension of methods and results of Formal Language Theory to the description of sets of finite graphs. This informal overview presents the main definitions, results and open problems and tries to answer some frequently asked questions.

Tree-width and *monadic second-order (MS) logic* are well-known tools for constructing *fixed-parameter tractable (FPT)* algorithms taking tree-width as parameter. *Clique-width* is, like tree-width, a complexity measure of graphs from which FPT algorithms can be built, in particular for problems specified in MS logic. These notions are thus essential for constructing (at least theoretically) tractable algorithms but also in the following three research fields:

- the study of the structure of graphs excluding induced subgraphs, minors or *vertex-minors* (a notion related to clique-width, see [48] or [18]);
- the extension of *language theoretical notions* in order to describe and to transform sets of finite and even countable graphs;
- the investigation of classes of finite and countable graphs on which MS logic is decidable.

Although these four research fields have been initially developed independently, they are now more and more related. In particular, new structural results for graph classes have consequences for algorithmic applications (see [7]).

This overview deals only with *finite* graphs, trees and relational structures. There is a rich theory of *countable graphs described by logical formulas, logically defined transformations, equation systems and finite automata*. The survey [1] is a good approach of this theory.

Graph structure and logic

* Supported by the GRAAL project of "Agence Nationale pour la Recherche".

Graph structure is a flexible concept covering many different cases. *Hierarchical decompositions* form an important type of structuring. Those yielding the notions of *tree-width* and *clique-width* can be expressed by terms written with graph operations defined below that generalize the concatenation of words. There exist other types of hierarchical structurings that are useful for establishing results or for algorithmic purposes. Examples are the *modular decomposition* defined by Gallai ([20], [38]), the *split decomposition* (also called *join decomposition*) defined by Cunnigham ([25], [33]), the decomposition in *3-connected components* defined by Tutte ([11], [21]), the *clique-sum decomposition* ([52], [7]).

The existence of an embedding in a fixed surface, or of a homomorphism into a fixed graph (a proper vertex coloring with p colors of a loop-free graph can be defined as a homomorphism of this graph into the clique K_p) is also a type of structure ([8], [43]). Finally, the non-existence in a graph of particular induced subgraphs, minors or vertex-minors is also an important type of structural property. (See [48] for vertex-minors).

There exist nontrivial relations between these different types of structures: graphs without a fixed planar graph P as a minor have tree-width at most $f(size(P))$ for some function f ([50]); graphs embeddable in a fixed surface are characterized by finitely many excluded minors ([51]); forbidding certain induced subgraphs implies bounded clique-width ([15], [16]), just to take a few examples, to which one could add the *restricted duality theorems* of [43]. There are still many open questions concerning comparisons between various types of graph structure.

Monadic second-order logic (MS logic in short) is the extension of first-order logic with quantified variables denoting subsets of the considered relational structures, hence sets of vertices when it is used for graphs, and sets of edges when a graph is represented by its incidence graph. It can express many graph properties: degree constraints, existence of proper colorings with fixed numbers of colors, connectivity, existence of spanning trees with particular properties, absence or existence of particular induced subgraphs or minors. From characterizations by forbidden configurations, one obtains that the sets of cographs, of distance-hereditary graphs, of planar graphs, of graphs embeddable in a fixed surface, of graphs of tree-width bounded by a fixed constant are *MS-definable*, *i.e.*, can be characterized as the finite models of certain MS formulas.

Graph structure notions are related with MS logic in several ways that we can classify under two main titles: *Expressive power of MS logic* and *Construction of algorithms*. We will discuss later the language theoretical aspects.

Expressive power of monadic second-order logic

It is easy to construct an MS formula expressing that a given graph has no minor or no induced subgraph isomorphic to a fixed finite graph. Hence the set of graphs of tree-width at most k is MS-definable because it is characterized by finitely many excluded minors. A set of graphs defined by excluded induced subgraphs forming a set that is infinite but MS-definable is also MS-definable.

Perfect graphs and *comparability graphs* are of this type. Their definitions are not directly translatable into MS formulas ([17], [38], [24]).

Monadic second-order logic can also be used to specify graph transformations. By analogy with the transformations of words and terms called *transductions* in language theory, I call *monadic second-order (MS) transductions* certain transformations of relational structures (hence of trees, graphs and hypergraphs) that can be specified by MS formulas. They generalize the notion of *interpretation* used in model theory and the *rational transductions* (transforming words into sets of words) such that the image of every word is finite. (Due to space limitations, definitions are not given formally. They can be found in the given references and in my book in preparation [3].)

In many situations concerning graph structure, one needs more than a yes or no answer. For an example, that a graph does not contain K_5 or $K_{3,3}$ as a minor implies that it is planar, but this fact does not describe any planar embedding. In other words, we are not only interested in checking that a given graph "has some structure", *e.g.* a tree-decomposition or a planar embedding, but also in having an MS transduction that constructs from the given graph some tree-decomposition or some planar embedding. Such transductions may be difficult to construct. Sometimes, they use edge set quantifications (decomposition in 3-connected components, [21]), and/or auxiliary linear orderings of the input structures. Constructions of planar embeddings, of the modular and split decompositions, of the *chord diagram defining a circle graph*, respectively are considered in this perspective in [22], [20], [25], [26].

Independently of these graph theoretical applications, MS transductions are useful tools for building MS formulas because the *inverse image of an MS-definable set* of graphs or relational structures under an MS transduction is MS-definable.

Construction of algorithms

Books by Downey and Fellows [4] and by Flum and Grohe [6], survey articles by Grohe [7], and by Makowsky [9], and many other articles have popularized the facts that MS expressible graph problems have FPT algorithms for tree-width and clique-width taken as parameters. We will refer by CMS to the extension of MS logic allowing set predicates $Card_p(X)$ expressing that the cardinality of a set X is a multiple of p, by MS_2 to the extension allowing *edge set quantifications* (also called *guarded second-order logic* in [39]) and by CMS_2 to the combination of both extensions.

Theorem 1 (Fixed-Parameter Tractability Theorem) : *Every CMS_2 expressible graph problem has a fixed-parameter linear algorithm for tree-width. Every CMS expressible graph problem has a fixed-parameter cubic algorithm for clique-width. These results extend to the counting and optimization problems specified in these extensions of MS logic.*

This result makes particularly interesting the expression of graph properties in CMS or CMS_2 logic. However, monadic second-order logic yields no polynomial algorithm for graphs of unbounded tree-width or clique-width : each level

of the polynomial hierarchy contains complete problems expressible in MS logic ([45]).

Algebraic characterizations of tree-width and clique-width.

Tree-width and clique-width are based on graph decompositions that can be expressed with *graph operations*. Such operations generalize the concatenation of words. For defining tree-decompositions, we use graphs with distinguished vertices called *sources* (or *boundary vertices* in [4]) specified by labels. Each label designates a single vertex. The corresponding graph operations are the *parallel-composition* that glues two graphs at their sources with same labels and unary operations that remove or modify source labels. The basic graphs are isolated vertices and graphs with a single edge. Tree-decompositions correspond closely to terms built with these operations and the tree-width of a graph is the minimum number of labels to be used to construct it with these operations.

Clique-width is similar but it is defined with different operations. These operations use also labels but a label may be attached to several vertices. The relevant operations are disjoint union (denoted by $\oplus$), unary operations $add_{a,b}$ that add edges between every vertex labeled by a and every vertex labeled by b, and unary operations that modify labels. The basic graphs are isolated vertices. The clique-width of a graph is the minimum number of labels to be used to construct it with these operations. Whereas words are generated from letters by a single binary operation, operations that use countably many labels (they form a countable set) are needed for generating all graphs,

Is Theorem 1 best possible ?

No. That a graph is Hamiltonian can be decided in polynomial time on graphs of bounded clique-width, although this property is MS_2 but provably not CMS [55]. (The algorithm is not FPT).

However, some converse results do exist : *if every existential monadic second-order property* (3-vertex colorability is an example of such a property) is *decidable in polynomial time on all graphs of a set $\mathcal{C}$ that is closed under taking minors or topological minors then $\mathcal{C}$ has bounded tree-width.* The closure conditions under taking minors cannot be replaced by closure under taking subgraphs. These results are proved in [44].

Is Theorem 1 practical usable ?

Not directly, for several reasons. First, because the algorithms need appropriate hierarchical decompositions of the input graphs. A tree-decomposition of width at most k can be found in linear time if there exists one, but the constant depends exponentially on k, and the algorithm is not implementable [13]. For clique-width, the situation is similar: one can construct in cubic time a clique-width expression of width at most $2^{k+1} - 1$ if the given graph has clique-width at most k by an algorithm derived from [41] and [49], but this algorithm is too complex to be implemented. Deciding if the clique-width of a graph G is at most k for given (G, k) is NP-complete [35]. This problem is polynomial for $k < 4$ and open for $k = 4$. A second difficulty comes from the translation of MS formulas into the finite automata on terms, on which the FPT algorithms are based. Since

short formulas can express complicated properties, these automata have in the worst cases non-elementary sizes in terms of the considered formulas ([37]).

Is the situation hopeless ?

Fortunately not! There exist implementable algorithms producing non optimal but usable tree-decompositions [14]. For clique-width, there exist algorithms based on modular decomposition that can produce in linear time optimal clique-width expressions for graphs from particular classes ([16], [30]). Another possibility consists in inputting graphs that have "natural" tree-decompositions (just because of the nature of the problems they formalize) or graphs produced by context-free grammars with their derivation trees, because derivation trees are hierarchical decompositions of the appropriate types. The second difficulty appears in the general statement intended to cover all formulas, but concrete problems may yield automata of reasonable sizes. Softwares like MONA [42] may be used for graphs defined by terms written with the operations described above, as well as directly for words and terms [54].

What about first-order formulas ?

There are FPT algorithms for model-checking of first-order formulas on certain classes of graphs of unbounded tree-width or clique-width, for instance, on those that have *locally bounded tree-width* or that exclude a fixed graph as a minor ([36], [7]). The structural description of the latter types of graphs used in [52] for proving the *Graph Minor Theorem* finds here unexpected applications. Nešetřil and Ossona de Mendez also apply structure theorems to the verification of first-order formulas expressing graph inclusions [47].

Could one use terms describing graphs written with other operations than those defining tree-width and clique-width, and that would have good "compatibility" with MS logic ?

One could use the *unfolding* operation that transforms a directed graph into the tree of finite paths issued from a specified vertex, because the inverse image under it of a CMS definable set of trees is CMS definable [32]. This operation is used together with MS transductions in order to construct countable graphs having decidable MS theories. These graphs form a hierarchy defined by Caucal (see the survey [1]). This operation has not yet been used to my knowledge to describe sets of finite graphs in a similar way. By using it, one could obtain compact representations of large graphs. (Trees with 2^n nodes can be described by terms of size n that encode directed acyclic graphs.)

Language theoretical concepts extended to sets of finite graphs

Two sets of graph operations have been defined above to characterize algebraically tree-width and clique-width. They define two algebraic structures on finite labelled graphs and two types of descriptions of these graphs by terms. Algorithmic applications are based on that, but so are also the language theoretical notions of context-free graph grammars and recognizability.

Context-free grammars as equation systems.

Context-free grammars are usually defined as sets of rewriting rules, however, a classical theorem characterizes the context-free languages as the components

of the least solutions of equation systems that are easily built from context-free grammars. These systems define languages in a recursive way in terms of set union, the extension to sets of the concatenation of words, letters and a constant denoting the empty word. Context-free languages are thus the *equational subsets* of the free monoid. This characterization extends to arbitrary algebras (Mezei and Wright [46]), even to those with infinite sets of operations, and in particular to our graph algebras. We obtain thus *context-free graph grammars,* defined formally as *equation systems*, without having to consider derivation sequences, permutation of derivation steps, derivation trees, because these notions are useless or are given for free in the algebraic setting.

Closure under union and under the operations of the algebra, as well as decidability results (emptiness, finiteness) can be established once and for all at the algebraic level. Since systems are finite, algorithms on them make sense although the global set of operations may be infinite.

Two robust classes of context-free graph grammars.

There are many possible graph algebras, and each of them yields a notion of equational set. However, two of them have emerged as particularly robust and interesting. These are the *HR algebra* whose operations are those characterizing tree-width, and the *VR algebra* related similarily to clique-width. The acronym HR stands for *Hyperedge Replacement* and refers to an algebra of graphs, the equational sets of which are exactly those defined independently by *hyperedge replacement (hypergraph) grammars*. Its operations are those from which tree-width can be defined. In particular, the set of graphs of tree-width at most k is HR-equational. Similarly, VR stands for *Vertex Replacement* and refers to other graph grammars that actually generate the VR-equational sets. Its operations have been designed in [28] so that the sets defined by certain grammars be the corresponding equational sets. The set of graphs of clique-width at most k is VR-equational.

In which sense are these classes robust ?

Our robustness criterium is stability under MS transductions, generalizing the fact that the families of context-free and of regular languages are closed under rational transductions. Furthermore, context-free languages are generated by rational transductions from particular context-free languages that describe trees. (Rational transductions are compositions of homomorphisms, inverse homomorphisms, intersections with regular languages. They are closed under composition and under inverse. Monadic second-order transductions are closed under composition, but not under inverse.)

The family of HR equational sets of graphs is characterized as the set of images of binary trees under MS_2 transductions (those that transform graphs *via* their incidence graphs), hence is closed under these transductions ([27]). It follows also that MS_2 transductions *preserve bounded tree-width*. The family of VR equational sets is the set of images of binary trees under MS transductions, hence is closed under these transductions ([34]) which consequently *preserve bounded clique-width*. These facts give characterizations independent of the chosen alge-

bras. Furthermore, they help to prove that "small variations" on the signatures preserve the corresponding classes of equational sets ([12], [29]).

What do we get from definitions of sets of graphs by equation systems ?

We get relatively *compact descriptions*. By using *derived operations*, one can make them (hopefully) more readable. For example, series-composition of graphs with two sources, denoted by $\bullet$, is not a basic operation of the HR algebra, but it is defined by a term over the basic operations, and $\bullet$ can replace this term in an equation system. The equation defining series-parallel graphs is then $S = S//S \cup S \bullet S \cup e$ where $//$ and e that denote respectively *parallel-composition* and a single edge, are basic operations. Every generated graph has at least one derivation tree, which is a term over the operations of types HR and VR and their extensions with derived operations. This term can be used for *storing the graph as a string of symbols*, and as *input to algorithms*. Extensions of the *Semi-Linearity Theorem* for context-free languages (Parikh's theorem) make it possible to extract numerical informations, like the possible numbers of vertices and/or edges in a generated graph. Incorrect graphs can thus be detected by using this result as a preliminary test. *Filtering theorems* (see below) make it possible to transform equation systems. *Parsing* is more difficult for context-free graph grammars than for context-free word grammars. It is NP-complete for certain particular grammars and polynomial for others. (See the first two chapters of [10]). Unambiguous graph grammars would be interesting for counting purposes, like are unambiguous context-free grammars, (see the book by Flajolet and Sedgewick [5]). *Ambiguity* for a graph grammar is actually not that obvious to define, in particular because of associative and commutative operations like $//$ in the above definition of series-parallel graphs.

Recognizability

Two of the various equivalent characterizations of regular languages are interesting for dealing with graphs. First, their characterization in terms of *finite congruences*, because it applies to every algebra (Mezei and Wright in [46]) and second their characterization as the set of MS-definable languages. MS-definability is only meaningful for logical structures or for objects represented by such structures, and this is the case for graphs. In the case of languages, MS-definability is equivalent to recognizability, but for graphs one has only one implication : MS-definability implies recognizability. (It is open to find restrictions on the congruences used in the definition of HR- (or VR-) recognizability so as to make it equivalent to CMS_2- (or CMS-) definability.)

Theorem 2 (Recognizability Theorem): *Every CMS_2-definable set of graphs is recognizable in the HR algebra. Every CMS-definable set of graphs is recognizable in the VR algebra.*

The opposite implications cannot hold since there are uncountably many HR- and VR- recognizable sets of graphs. This uncountability result is linked to the infiniteness of the signatures of the HR and VR algebras. However a result stated in [40] says that a set of graphs of bounded tree-width is CMS_2-definable if and only if it is HR-recognizable. No similar result is known for

VR-recognizability. An important consequence of the *Recognizability Theorem* is the *Filtering Theorem*.

Theorem 3 (Filtering Theorem): *The intersection of an HR-equational set of graphs with a CMS_2-definable one is effectively HR-equational. The intersection of a VR-equational set with a CMS-definable one is effectively VR-equational.*

Direct constructions for particular properties like planarity (which is MS definable) would be particularly long and technical.

How can one prove the Recognizability Theorem ?

One proof can be sketched as follows : let Θ_k be the set of monadic second-order sentences (closed formulas) of quantifier height at most k over a fixed relational signature, and written in a certain normal form. This set has large cardinality but is finite. For a structure S we let $M_k(S)$ be the *level k theory of* S, *i.e.*, the set of sentences in Θ_k that are true in S. The main lemma states that $M_k(S \oplus T)$ can be computed from $M_k(S)$ and $M_k(T)$ by a function that depends only on k ($\oplus$ is disjoint union). A second (straightforward) lemma states that if t is a transformation of structures that can be expressed by *quantifier free (QF) formulas*, (this is the case of a relabelling, of the edge creation operation $add_{a,b}$ and of the edge complement to take typical examples) then $M_k(t(S))$ can be computed from $M_k(S)$ by a function that depends only on k and t. Since all VR and HR operations are expressible in terms of $\oplus$ and operations definable by QF formulas, the proof can be completed as follows: one defines a congruence on relational structures by $S \equiv T$ iff $M_k(S) = M_k(T)$. Technical details are omitted but everything is on the table. The lemma about $\oplus$ has generalizations involving combinations of infinite families of structures presented in [9]. The case of a mapping t such that $M_k(t(S))$ can be computed from $M_{f(k)}(S)$ for some fixed function f by a function that depends only on k and t, is also interesting. This is the case of unfolding ; such operations are discussed in [9].

Recognizability is a difficult notion to handle (the emptiness of an HR-recognizable set of graphs of unbounded tree-width is undecidable) but MS logic provides a handy language for specifying recognizable sets. In many cases the expression of a graph property by a monadic second-order formula is straightforward. Finite automata and regular expressions make it easy to specify recognizable languages, and monadic second-order formulas play a similar role for recognizable sets of graphs. They replace finite automata. No existing notion of graph automaton gives an equivalence with monadic second-order logic. If graph automata of some type would be effectively equivalent to monadic second-order formulas for all graphs, their emptiness problem would be undecidable because so is the corresponding problem for MS logic.

Back to equational sets.

The *Filtering Theorem* stated above is a direct application of the Recognizability Theorem and of the following fact : *in every algebra the intersection of an equational set and an effectively given recognizable set is effectively equational.* This generalizes the fact that the intersection of a context-free language and a regular one is context-free. Together with the decidability of emptiness, we get

that MS logic is decidable on VR-equational sets, and that CMS_2 logic is decidable on HR-equational sets. This means that one can test if every graph of the considered equational set satisfies the considered formula.

Could one prove or disprove mechanically (at least theoretically) some conjectures of graph theory by using this theorem ?

Only those of the form : every graph in a particular VR- or HR-equational set, or of clique-width or tree-width at most some given k, satisfies some property expressible in MS logic. But most graph theoretic conjectures (like *Hadwiger's Conjecture*) concern all graphs, and are not restricted to equational sets. Another difficulty with this idea is that graph properties involving comparisons of cardinalities are not MS expressible in general.

On which classes can one decide MS logic ?

There is a structural necessary (but not sufficient) condition.

Theorem 4 (Structural preresquisite for MS decidability) : *(1) Every set of graphs having a decidable MS_2 theory has bounded tree-width, hence is a subset of some HR-equational set. (2) Every set of graphs having a decidable C_2MS theory has bounded clique-width, hence is a subset of some VR-equational set.*

The first statement is proved in [53], and the second one in [18]. The hypothesis that the C_2MS theory (C_2MS = MS with the even cardinality set predicate $Card_2(X)$) is decidable is stronger than requiring that the MS theory is decidable. These proofs use the result of [50] that excluding a planar graph as a minor implies bounded tree-width, and an extension of it to matroids. Hence in the case of sets of graphs, there are two obstacles to the decidability of CMS (or CMS_2) logic: unbounded clique-width (or tree-width), and, for sets of bounded clique-width or tree-width, the "internal complexity" of the considered sets. Sets of words can be complex enough so as to forbid the decidability of MS logic, although words are, considered as graphs, of tree-width 1.

By using the unfolding operation applied to directed acyclic graphs and MS transductions, one can construct classes of graphs that are not VR equational, that have decidable CMS-theories but (necessarily by Theorem 4) have bounded clique-width. That such graphs have bounded clique-width follows more directly from the fact that unfolding makes a graph into a tree. The use of unfolding increases the "internal complexity" of the described graphs, while keeping decidability of MS logic.

Are there fragments of MS logic that have decidable theories on classes of graphs of unbounded clique-width ?

An example is first-order logic, which is decidable on square grids (but not on all planar graphs). The satisfiability problem for existential monadic second-order logic (sentences of the form $\exists X_1, \ldots, X_n \varphi$ with φ first-order) is undecidable on square grids. However one might look for improvements of Theorem 4 where the hypotheses are the decidabilities of $\mathcal{L}$-theories for fragments $\mathcal{L}$ of MS logic.

Does the decidability of a fragment $\mathcal{L}$ of MS logic on a set of graphs $\mathcal{C}$ imply the existence of a polynomial time algorithm for each $\mathcal{L}$-expressible property over the graphs in $\mathcal{C}$?

For CMS logic and clique-width, and for CMS_2 logic and tree-width, the *Recognizability Theorem* implies simultaneously Theorem 1 and the decidabilities of CMS and CMS_2 logics over graphs of bounded clique-width and tree-width respectively, but, the only implication uses the detour through Theorem 4.

Couldn't we extend the languages CMS and CMS_2 while keeping decidability of the corresponding theories on graphs of bounded clique-width and tree-width ?

No such extension exists to my knowledge. The extension of CMS logic by an equal cardinality predicate $Eq(X, Y)$ expressing that sets X and Y have equal cardinality is undecidable on the set of words over a single letter. Another possibility could be with a cardinality oracle. That is we fix a recursive set of integers A and we let $Card_A(X)$ mean : X has cardinality in A. If A is the set of numbers that are either a power of 2 or a power of 3, then the corresponding extension of MS is *undecidable* on words. If A is the set of prime numbers the decidability is unknown. (This extension is stronger than the extension of the linear order of natural numbers with a predicate $P_A(x)$ expressing that $x \in A$ because bijections are not definable by monadic second-order formulas).

Can't one define all graphs with a finite set of operations ?

Yes, one can define all linearly ordered loop-free undirected graphs from four unary operations : addition of a new isolated vertex as new last vertex, addition of a new edge between the first two vertices, exchange of the first two vertices and circular shift (the first vertex becomes the last one, the second one becomes the first, etc...). As single constant, one can use the empty graph. With these operations and the one that forgets the ordering, one can define all graphs by terms, but these terms that are nothing but lists of vertices and edges. No interesting hierarchical structure is obtained like with the HR and VR operations. MS logic is undecidable on the corresponding equational sets. Although "small" and "powerful" this set of graph operations is uninteresting.

Relational structures.

The results of Theorems 1,2,3 have been stated for graphs, but their extensions to *relational structures over finite relational signatures* are straightforward, because most results are proved either at the Universal Algebra level or are valid for relational structures (cf. the proof sketch of the *Recognizability Theorem*). What are the relevant algebras ? One of them, generalizing the VR algebra uses disjoint union and QF operations. Another one can be defined that extends the HR algebra. More manageable sets of operations that generate the same equational sets and the same recognizable sets have been considered in [19], [31], [12]. In most cases, proofs in terms of relational structures are no more difficult than for classes of graphs and give more general statements.

A challenging open problem : *Is it true that if a set of relational structures has a decidable CMS theory, then it is the image of a set of binary trees under an MS transduction ?* If true, this would generalize Theorem 4 (2) but the tools used for its proof do not extend obviously. I consider this question as the main one in the area of relationships between MS logic and graph structure. The corresponding extension of Theorem 4 (1) is not difficult.

Here is a last result that indicates how nicely recognizability and MS logic fit together.

Theorem 5 ([12]) : The inverse image of a recognizable set of relational structures under a CMS transduction is recognizable.

In this statement, recognizability is understood with respect the algebra based on disjoint union and QF operations. This result generalizes the fact that the inverse image of a CMS definable set of relational structures under a CMS transduction is CMS definable.

Acknowledgement : I thank A. Blumensath, M. Fellows and I. Walukiewicz for many useful comments on a first draft of this overview.

The first 11 references are books and survey articles.

References

1. A. Blumensath, T. Colcombet, C. Löding, Logical Theories and Compatible Operations, in *Logic and automata: History and Perspectives*, J. Flum, E. Grädel, T. Wilke, eds., Amsterdam University Press, 2008, pp. 73-106.
2. B. Courcelle, The Expression of Graph Properties and Graph Transformations, in Monadic Second-Order Logic, in *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*. G. Rozenberg ed., World Scientific, 1997, pp. 313-400
3. B. Courcelle, *Graph Structure and Monadic Second-order Logic*, book in preparation, to be published by Cambridge University Press, readable on http://www.labri.fr/perso/courcell /ActSci.html
4. R. Downey, M. Fellows, *Parameterized Complexity*, Springer-Verlag, 1999.
5. P. Flajolet, R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press, to appear.
6. J. Flum, M. Grohe, *Parameterized Complexity Theory*, Springer 2006
7. M. Grohe, Logic, Graphs, and Algorithms, in *Logic and automata: History and Perspectives*, J. Flum, E. Grädel, T. Wilke, eds., Amsterdam University Press, 2008, pp. 357-422.
8. P. Hell, J. Nešetřil, *Graphs and homomorphisms*, Oxford University Press, 2004
9. J. Makowsky, Algorithmic uses of the Feferman-Vaught Theorem, *Ann. Pure Appl. Logic* **126** (2004) 159-213
10. G. Rozenberg ed., *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*, World Scientific, 1997.
11. W. Tutte, *Graph Theory*, Addison–Wesley, 1984.
12. A. Blumensath, B. Courcelle, Recognizability, Hypergraph Operations, and Logical Types, *Inf.Comput.* **204** (2006) 853-919
13. H. Bodlaender, A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth, *SIAM J. Comput.* **25** (1996) 1305-1317
14. H. Bodlaender, Treewidth: Characterizations, Applications, and Computations, in *Proceedings of WG 2006, Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science **4271** (2006) 1-14
15. A. Brandstädt, F. Dragan, H. Le, R. Mosca, New Graph Classes of Bounded Clique-Width, *Theory Comput. Syst.* **38** (2005) 623-645

16. A. Brandstädt, J. Engelfriet, H. Le, V. Lozin, Clique-Width for 4-Vertex Forbidden Subgraphs, *Theory Comput. Syst.* **39** (2006) 561-590
17. M. Chudnovsky, N. Robertson, P. Seymour, R. Thomas, Progress on Perfect Graphs, *Mathematical programming, Ser. B* **97** (2003) 405-422
18. B. Courcelle, S. Oum, Vertex-minors, Monadic Second-order logic, and a Conjecture by Seese, *J. Comb. Theory, Ser. B* **97** (2007) 91-126
19. B. Courcelle, The Monadic Second-Order Logic of Graphs VII: Graphs as Relational Structures, *Theor. Comput. Sci.* **101** (1992) 3-33
20. B. Courcelle, The Monadic Second-Order Logic of Graphs X: Linear Orderings, *Theor. Comput. Sci.* **160** (1996) 87-143
21. B. Courcelle, The Monadic Second-Order Logic of Graphs XI: Hierarchical Decompositions of Connected Graphs, *Theor. Comput. Sci.* **224** (1999) 35-58
22. B. Courcelle, The Monadic Second-Order Logic of Graphs XII: Planar Graphs and Planar Maps, *Theor. Comput. Sci.* **237** (2000) 1-32
23. B. Courcelle, The Monadic Second-Order Logic of Graphs XIV: Uniformly Sparse Graphs and Edge Set Quantifications, *Theor. Comput. Sci.* **299** (2003) 1-36
24. B. Courcelle, The Monadic Second-Order Logic of Graphs XV: On a conjecture by D. Seese, *J. Applied Logic* **4** (2006) 79-114
25. B. Courcelle, The Monadic Second-Order Logic of Graphs XVI : Canonical graph decompositions, *Logical Methods in Computer Science* **2** (2006)
26. B. Courcelle, Circle Graphs and Monadic Second-order logic, *Journal of Applied Logic*, in press.
27. B. Courcelle, J. Engelfriet, A Logical Characterization of the Sets of Hypergraphs Defined by Hyperedge Replacement Grammars, *Mathematical Systems Theory* **28** (1995) 515-552
28. B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-Rewriting Hypergraph Grammars, *J. Comput. Syst. Sci.* **46** (1993) 218-270
29. B. Courcelle, J. Makowsky, Fusion in Relational Structures and the Verification of Monadic Second-Order Properties, *Mathematical Structures in Computer Science* **12** (2002) 203-235
30. B. Courcelle, J. Makowsky, U. Rotics, Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width, *Theory Comput. Syst.* **33** (2000) 125-150
31. B. Courcelle, P. Weil, The Recognizability of Sets of Graphs is a Robust Property, *Theor. Comput. Sci.* **342** (2005) 173-228
32. B. Courcelle, I. Walukiewicz, Monadic Second-Order Logic, Graph Coverings and Unfoldings of Transition Systems, *Ann. Pure Appl. Logic* **92** (1998) 35-62
33. W. Cunnigham, Decomposition of Directed Graphs, *SIAM Algor. Discrete Meth.* **3** (1982) 214-228
34. J. Engelfriet, V. van Oostrom, Logical Description of Contex-Free Graph Languages, *J. Comput. Syst. Sci.* **55** (1997) 489-503
35. M. Fellows, F. Rosamond, U. Rotics, S. Szeider, Clique-width Minimization is NP-hard, *38th Annual ACM Symposium on Theory of Computing*, 2006, pp. 354-362
36. M. Frick, Generalized Model-Checking over Locally Tree-Decomposable Classes, *Theor. Comput. Sci.* **37** (2004) 157-191
37. M. Frick, M. Grohe, The Complexity of First-order and Monadic second-order Logic Revisited, *Ann. Pure Appl. Logic* **130** (2004) 3-31
38. T. Gallai, Transitiv Orientierbare Graphen, *Acta Math. Acad. Sci. Hungar.* **18** (1967) 25-66; translation in English by F. Maffray and M. Preissmann, in: J.L. Ramirez Alfonsin, B.A. Reed (Eds.), *Perfect Graphs*, Wiley, New York, 2001, pp. 25-66.

39. E. Grädel, C. Hirsch, M. Otto, Back and Forth Between Guarded and Modal Logics, *ACM Trans. Comput. Log.* **3** (2002) 418-463
40. D. Lapoire, Recognizability Equals Monadic Second-Order Definability for Sets of Graphs of Bounded Tree-Width, *STACS 1998*, *Lecture Notes in Computer Science* **1373** (1998) 618-628
41. P. Hlinený, S. Oum: Finding Branch-Decompositions and Rank-Decompositions, *ESA 2007 Lecture Notes in Computer Science* **4698** (2007) 163-174
42. N. Klarlund, Mona & Fido: The Logic-Automaton Connection in Practice, *Computer Science Logic 1997, Lecture Notes in Computer Science* **1414** (1998) 311-326
43. F. Madelaine, Universal Structures and the Logic of Forbidden Patterns, *Computer Science Logic 2006, Lecture Notes in Computer Science* **4207** (2006) 471-485
44. J. Makowsky, J. Mariño, Tree-width and the Monadic Quantifier Hierarchy, *Theor. Comput. Sci.* **303** (2003) 157-170
45. J. Makowsky, Y. Pnueli, Arity and Alternation in Second-Order Logic, *Ann. Pure Appl. Logic* **78** (1996) 189-202; Erratum: *Ann. Pure Appl. Logic* **92** (1998) 215
46. J. Mezei, J. Wright, Algebraic Automata and Context-Free Sets, *Information and Control* **11** (1967) 3-29
47. J. Nešetřil, P. Ossona de Mendez, Linear Time Low Tree-width Partitions and Algorithmic Consequences, *Proc. Symp. Theory of Computation* (2006) 391-400
48. S. Oum, Rank-width and Vertex-minors, *J. Comb. Theory, Ser. B* **95** (2005) 79-100
49. S. Oum, P. Seymour, Approximating Clique-width and Branch-width, *J. Comb. Theory, Ser. B* **96** (2006) 514-528
50. N. Robertson, P. Seymour, Graph Minors. V. Excluding a Planar Graph, *J. Comb. Theory, Ser. B* **41** (1986) 92-114
51. N. Robertson, P. Seymour, Graph minors. VIII. A Kuratowski Theorem for General Surfaces, *J. Comb. Theory, Ser. B* **48** (1990) 255-288
52. N. Robertson, P. Seymour, Graph Minors. XVI. Excluding a Non-planar Graph, *J. Comb. Theory, Ser. B* **89** (2003) 43-76
53. D. Seese, The Structure of Models of Decidable Monadic Theories of Graphs. *Ann. Pure Appl. Logic* **53** (1991) 169-195
54. D. Soguet, Génération Automatique d'Algorithmes Linéaires, *Doctoral dissertation*, Paris-Sud University, France, July 2008.
55. E. Wanke, k-NLC Graphs and Polynomial Algorithms, *Discrete Applied Mathematics* **54** (1994) 251-266