

Foreword: special issue on term and graph rewriting

RACHID ECHAHED

CNRS and Université Grenoble-Alpes, Grenoble, France
Email: rachid.echahed@imag.fr

Received 1 May 2018

Rewriting techniques constitute a foundational theory of computing science. They are being investigated for several structures, such as lambda-terms, strings, first-order terms or graphs, and have been successfully used in many areas such as programming languages, automated reasoning, program verification, security, etc. The growing interest in this research area is witnessed by the leading international events, such as ICGT (International Conference on Graph Transformation) and the recent FSCD conference (International Conference on Formal Structures for Computation and Deduction), which gathers all topics of the former international conferences RTA (Rewriting Techniques and Applications) and TLCA (Typed Lambda Calculi and Applications). During the last decade, a particular interest has been devoted to the study of the impact of shared structures in term and graph rewriting through the international workshops editions of TERMGRAPH and GCM (Graph Computation Models).

This special issue of *Mathematical Structures in Computer Science* features some recent developments in term and graph rewriting and contains six contributions ranging from theoretical aspects of term and graph transformation, proof methods and semantics to application issues and tools.

String rewrite systems have been formally investigated since the beginning of the 20th century (see, e.g., semi-Thue systems). Their extension to graphs and formal languages is key to master complex modern languages where graph shapes are handled. In the paper, *Recognizable Languages of Arrows and Cospans* by H.J. Sander Bruggink and Barbara König, the authors generalize Courcelle's recognizable graph languages and results on monadic second-order logic to more general structures. The paper provides a category-theoretical characterization of recognizability. A recognizable subset of arrows in a category is defined via a functor into the category of relations on finite sets. This can be seen as a straightforward generalization of finite automata. In addition, a simple logic that allows one to quantify over the sub-objects of a categorical object is introduced and shown, for the category of graphs, to be equally expressive as monadic second-order graph logic. Furthermore, the authors show that in the more general setting of hereditary pushout categories, a class of categories closely related to adhesive categories, one can recover Courcelle's result that every property expressible in monadic second-order graph logic is recognizable.

Reasoning on graph transformations did not reach yet the same level of maturity as that of first-order term rewriting; much remains to be done in this area. The paper *Interactive and automated proofs for graph transformations* by Martin Strecker provides a contribution in this direction. The author defines a general framework for representing graphs and

graph rewriting steps and identifies fragments of the proposed graph description language in which one can reduce reasoning about global graph properties to reasoning about local properties, involving only a bounded number of nodes, which can be decided by Boolean satisfiability solving or even by deterministic computations of low complexity.

Term graph rewriting provides a formalism for implementing term rewriting in an efficient manner by emulating duplication via sharing. Infinitary term rewriting has been introduced to study infinite term reduction sequences. Such infinite reductions can be used to model *non-strict evaluation*. In the paper, *Convergence in Infinitary Term Graph Rewriting Systems is Simple* by Patrick Bahr, the author unifies term graph rewriting and infinitary term rewriting thereby addressing both components of *lazy evaluation*: non-strictness and sharing. The proposed approach is based on a generalization of the modes of convergence of infinitary term rewriting. It is shown that this new approach is better suited for infinitary term graph rewriting since it is simpler and more general than former ones. In addition, the notions of convergence on term graphs are shown to be sound w.r.t. the ones employed in infinitary term rewriting in the sense that convergence is preserved by unravelling term graphs to terms. Moreover, the resulting infinitary term graph calculi provide a unified framework for both infinitary term rewriting and term graph rewriting, which makes it possible to study the correspondences between these two worlds more closely.

Term graph rewriting is also considered in the paper *Linear Pattern Matching of Compressed Terms and Polynomial Rewriting* by Manfred Schmidt-Schauß where the author considers a class of term rewriting under sharing in the form of compression by singleton tree grammars (STG), which is more general than term dags. Algorithms for the subtasks of rewriting are analyzed: finding a redex for rewriting by locating a position for a match, performing a rewrite step by constructing the compressed result, and executing a sequence of rewrite steps. The author shows that locating a match of a linear STG-compressed term in another STG-compressed term can be performed in polynomial time, while the general case of compressed sub-matching can be computed in non-deterministic polynomial time. He also shows that in case there is an oracle for the redex position, a sequence of parallel or single-step rewriting steps under STG-compression can be performed in polynomial time.

As said earlier, term and graph rewriting have several application domains. In the paper, *Non-size increasing Graph Rewriting for Natural Language Processing* by Guillaume Bonfante and Bruno Guillaume, the authors provide a very nice application of graph transformations in Natural Language Processing. They define a dedicated graph transformation framework tailored for the targeted representation of natural languages and focus on the termination aspect of the considered graph transformations. In particular, they propose termination criteria for weighted rewrite systems and give complexity bounds for the lengths of graph rewrite derivations.

Rewriting notions like termination, normal forms and confluence can be described in an abstract way referring to rewriting only as a binary relation. Several results on rewriting, like Newman's lemma, can be proved in this abstract setting. To investigate possible generalizations of such theorems, it is fruitful to have counter-examples showing that particular generalizations do not hold. In the paper, *Finding small counterexamples*

for abstract rewriting properties by Hans Zantema, the author develops a technique to find such counter-examples fully automatically, and describes a tool, called Carpa, that follows this technique. The basic idea is to fix the number of objects of the abstract rewrite system, and to express the conditions and the negation of the conclusion in a satisfiability (SAT) formula, and then call a SAT solver. In case the formula turns out to be satisfiable, the resulting satisfying assignment yields a counter-example to the encoded property. The paper provides several examples of finite abstract rewrite systems having remarkable properties that have been found automatically.

I would like to thank all those who have contributed to this special issue, especially the reviewers for their valuable contributions to the selection process as well as the contributing authors, whether their article was accepted or not. I also would like to express my gratitude to Giuseppe Longo and Pierre-Louis Curien from the MSCS editorial board for their help in making this issue possible.