

Advances in Computational Logic (CILC08)

Preface

This special issue of *Fundamenta Informaticae* contains the revised and expanded versions of selected papers presented at CILC08, the 23rd Italian Congress on Computational Logic, held in Perugia, Italy, 10–12 July, 2008. This event was organized under the auspices of the Italian Association for Logic Programming (GULP), the Department of Mathematics and Informatics of the University of Perugia, and the Italian National Group for Scientific Computing (GNCS-INdAM).

The annual congresses of the Italian Association of Logic Programming, since the first one in 1986, have always been fruitful and stimulating occasions for researchers, both from Italy and abroad, to present scientific results, exchange ideas, and propose innovative projects in the area of Computational Logic. The unceasing success of these congresses reflects the vibrant state of a prolific research that covers the theoretical aspects and also the practical applications of Computational Logic.

The papers of this special issue were selected on the basis of the reviews of the program committee, the presentations at the congress, and the discussions which followed the presentations. The authors were invited to submit revised, extended versions of their contributions. Those versions were subject to a second reviewing process and a final selection was made. Seven papers were chosen and, after a further revision by the authors, they are now collected in this special issue.

In the paper *Integrating Finite Domain and Set Constraints into a Set-based Constraint Language*, the authors F. Bergenti, A. Dal Palù, and G. Rossi propose a constraint language that generalizes existing languages over finite domains and sets. This paves the way for the design of an integrated constraint solver that embodies and improves over previously proposed solvers.

The paper *Actions over a Constructive Semantics for Description Logics*, by L. Bozzato, M. Ferrari, and P. Villa, proposes an action formalism based on a constructive information-terms semantics for the description logic \mathcal{ALC} . The formalism naturally supports the modeling of states and actions, so that executability of actions, their effects, and the existence/consistency of target states can be algorithmically determined.

The paper *Normal Form Nested Programs*, by A. Bria, W. Faber, and N. Leone, considers Normal Form Nested Programs, an extension of ordinary logic programs with answer set semantics. The language allows disjunctions of conjunctions of atoms in the heads of the rules, whose bodies are formulas in conjunctive normal form. Interesting properties and theoretical results regarding this class of programs are reported.

A novel approach for computing answer sets of non-ground logic programs is proposed in the paper *GASP: Answer Set Programming with Lazy Grounding*, by A. Dal Palù, A. Dovier, E. Pontelli, and G. Rossi. Usually ASP-solvers perform a preliminary grounding phase. Instead, in this approach, the grounding of a program is lazily performed during the computation of the answer sets. The implementation relies on a translation to Constraint Logic Programming over Finite Domains.

The paper *Model Checking for Graded CTL*, by A. Ferrante, M. Napoli, and M. Parente, deals with the model checking problem on Graded Computational Tree Logic (Graded CTL), that is, an extension of CTL that incorporates graded modalities. Algorithmic issues as well as a characterization of computational complexity are discussed.

The paper *ALC + T: a Preferential Extension of Description Logics*, by L. Giordano, V. Gliozzi, N. Olivetti, and G.L. Pozzato, focuses on the problem of extending Description Logic with a ‘typicality’ operator that supports the reasoning about prototypical properties and inheritance with exceptions. Issues such as complexity, decidability, and mechanization of reasoning in *ALC + T* are investigated.

Finally, in the paper *A Folding Rule for Eliminating Existential Variables from Constraint Logic Programs*, the authors V. Senni, A. Pettorossi, and M. Proietti, present a folding rule for constraint logic programs defined over rational and real numbers. The authors propose an algorithm based on constraint solving and rewriting techniques, to check that the applicability conditions of that folding rule are met.

We would like to thank the reviewers for their hard work and their helpful comments and suggestions. We also express our gratitude to all the authors of the papers.

The Guest Editors

Andrea Formisano

Department of Mathematics and Informatics
University of Perugia, Perugia, Italy
formis@dmi.unipg.it

Alberto Pettorossi

Department of Informatics, Systems, and Production
University of Rome “Tor Vergata”, Rome, Italy
pettorossi@info.uniroma2.it