

Preface

The workshops on (constraint) logic programming (WLP) are the annual meeting of the Society of Logic Programming (GLP e.V.) and bring together researchers interested in logic programming, constraint programming, and related areas like databases, artificial intelligence and operations research. In this decade, previous workshops took place in Dresden (2008), Würzburg (2007), Vienna (2006), Ulm (2005), Potsdam (2004), Dresden (2002), Kiel (2001), and Würzburg (2000). Contributions to workshops deal with all theoretical, experimental, and application aspects of constraint programming (CP) and logic programming (LP), including foundations of constraint/logic programming. Some of the special topics are constraint solving and optimization, extensions of functional logic programming, deductive databases, data mining, nonmonotonic reasoning, , interaction of CP/LP with other formalisms like agents, XML, JAVA, program analysis, program transformation, program verification, meta programming, parallelism and concurrency, answer set programming, implementation and software techniques (e.g., types, modularity, design patterns), applications (e.g., in production, environment, education, internet), constraint/logic programming for semantic web systems and applications, reasoning on the semantic web, data modelling for the web, semistructured data, and web query languages.

The topics of the presentations of the 23rd WLP (WLP2009) are grouped into the major areas: Databases, Answer Set Programming, Theory and Practice of Logic Programming as well as Constraints and Constraint Handling Rules.

The topics of the invited talks deal especially with constraint solving. Tom Schrijvers discusses the advantages of monadic constraint solving which supplies a framework for unifying constraint solving over conjunctive constraints and search in a disjunctive search tree into a common and flexible strategy. Neng-Fa Zhou discusses the importance of choosing the right modeling and programming technique for an efficient solving of constraint problems.

In the **Database** session the declarative aspect of Logic Programming is used to make saver manipulation of relational databases and to extend relational databases to deductive databases. Michael Hanus and Sven Koschnike describe in “An ER-based framework for declarative web programming” a framework which ensures automatically the consistency of a database when the user performs update operations. The basic idea of this framework is an implementation of the conceptual model of a relation database in a declarative programming language. Dietmar Seipel proposes in the contribution “Practical applications of extended deductive databases in DATALOG*” a language DATALOG* for deductive databases which allows - different to classical Datalog - program execution in backward chaining manner. The advantages are given by a greater flexibility formulating programs and requests by using features of the PROLOG language and having available default negation, aggregation and complex data structures.

In the session on **Answer Set Programming** two papers deal with extensions of the ASP formalism and a further paper is devoted to the problem of program analysis.

In the paper on “xpanda: A (Simple) Preprocessor for Adding Multi-valued Propositions to ASP” Martin Gebser, Henrik Hinrichs, Torsten Schaub and Sven Thiele describe a transformation-based approach which makes it possible to model and solve constraint satisfaction problems in the ASP formalism. The authors discuss the problems of grounding in ASP which may lead to large search spaces in their first approach of combining ASP and constraint-solving. The paper on “Existential quantifiers in the rule body” of Pedro Cabalar deals with the extension of ASP for using existential quantifiers and double negations in the bodies of rules. The given introduction to the problem is nice readable and discusses the change over the last years in considering variables in ASP. The author describes a translation algorithm which uses the general technique of introduction of auxiliary predicates but which keeps these predicate hidden from the programmer for avoiding confusion. The third paper on “Kato: A Plagiarism-Detection Tool for Answer-Set Programs” in this session is presented by Johannes Oetsch, Martin Schwengerer and Hans Tompits. For different reasons there is a demand for finding similarities in different programs, the plagiarism detection. The authors present the presumable first such algorithm for ASP programs. Although the algorithm contains language-dependent parts, the authors stress the adaptability of the method for other languages (e.g. PROLOG).

The session on **Theory of Logic Programming** starts with a contribution of Heinrich Herre and Axel Hummel on “A Para-consistent Semantics for Generalized Logic Programs”. For using logic as a basis for knowledge representation, generalizations of standard logic programs with declarative semantics are needed which take into account the occurrence of contradictions. The authors discuss a generalization where arbitrary quantifier-free formulas in rules, which may contain default negations, may occur. A next step may be admitting quantifiers in the rules. The paper on “Stationary generated models of generalized Logic Programs“ by the same authors is devoted to the general problem of finding an adequate declarative semantics for generalized logic programs. The authors introduce for this purpose the notion of a so-called “intended partial model” and prove their properties. Gourinath Banda and John Gallagher present in their paper on a “Constraint-Based Abstraction of a Model Checker for Infinite State Systems” a new approach to verify properties of infinite-state systems by going back to the roots of abstract interpretation and exclusively applying standard techniques for constructing abstractions. The advantage consists in verifying any property of a system and a better efficiency, which originates partly from the integration of constraint solving techniques. The authors present implementation details and demonstrate their approach on two examples. Stefan Brass deals in „Range Restriction for General Formulas“ with the gap between the features of SQL and the current standard of Datalog. Some problems which arise by the use of general logic formulas are discussed. The basic question is to answer, which formulas should be allowed in logic rules.

The session **Constraint Handling Rules** contains a paper describing an application of CHR programming for program development and a theoretical contribution dealing with the extension of the CHR paradigm.

Slim Abdennadher, Haythem Ismail and Frederick Khoury discuss in their paper “Transforming Imperative Algorithms to Constraint Handling Rules” the idea using the flexibility and expressiveness of CHR for proving properties of imperative programs. Vice-versa this method could be used for an automatic generation of global constraint solvers. The authors describe the methodology for the conversion of imperative programs into CHR programs and discuss it with examples.

Hariolf Betz, Frank Raiser and Thom Frühwirth deal in “Persistent Constraints in Constraint Handling Rules” with the drawback that CHR as a high-level declarative language has to use a non-declarative token store in its implementation for avoiding trivial non-termination. A solution of this problem is presented which supplies a new operational semantics for CHR programs. Advantages are a higher degree of declarativity with avoidance of non-trivial termination and improved behaviour for concurrency.

The papers of the last session **Practice of Logic Programming** deal with a real world application of (Constraint) Logic Programming and a closer consideration of the difference list formalism.

Hans-Joachim Goltz and Norbert Pieth describe in their paper “A Tool for Generating Partition Schedules of Multiprocessor Systems” the application of Constraint-Logic Programming for modeling and solving a complex scheduling problem. Details for the derivation of constraints from the problem description and the design of the search are discussed. The optimization criterion is a high processor load. A graphical interface supports interactive control of the scheduling process by the user who may change weak constraints. The paper on “Efficiency of Difference-List Programming” by Ulrich Geske and Hans-Joachim Goltz analyses the use of a program construct in PROLOG which has a significant effect on the speed of list processing. The intention of the authors is the promotion of application of this construct, especially for PROLOG novices. Therefore simple syntactical patterns are derived which allow to control the order of elements in a list and an append-free insertion of elements into lists.

Finally, we would like to thank all the authors who have submitted papers, all colleagues who have presented invited talks, and all members of the program committee and external referees for reviewing the submissions and for their contributions to the success of the workshop.

Potsdam, September 15, 2009

Ulrich Geske
Armin Wolf

Organization

Program Chairs

Ulrich Geske (University of Potsdam)

Armin Wolf (Fraunhofer FIRST, Berlin)

Program Committee

Slim Abdennadher (German University Cairo)

Christoph Beierle (FernUniv. Hagen)

Stefan Brass (MLU Halle-Wittenberg)

Jürgen Dix (Clausthal University of Technology)

Tim Furche (LMU München)

Ulrich Geske (University of Potsdam)

Hans-Joachim Goltz (Fraunhofer FIRST, Berlin)

Michael Hanus (CAU Kiel)

Heinrich Herre (University of Leipzig)

Steffen Hölldobler (TU Dresden)

Petra Hofstedt (TU Berlin)

Ulrich John (SIR Plan GmbH)

Michael Leuschel (Univ. Düsseldorf)

Ulrich Neumerkel (TU Wien)

Frank Raiser (University of Ulm)

Georg Ringwelski (Hochschule Zittau/Görlitz)

Sibylle Schwarz (Hochschule Zwickau)

Dietmar Seipel (University of Würzburg)

Michael Thielscher (TU Dresden)

Hans Tompits (TU Wien)

Armin Wolf (Fraunhofer FIRST, Berlin)

Contents

Invited Talks

- Overview of the Monadic Constraint Programming Framework 1
Tom Schrijvers
- What I have learned from all these solver competitions 17
Neng-Fa Zhou

Databases

- An ER-based Framework for Declarative Web Programming 35
Michael Hanus and Sven Koschnicke
- Practical Applications of Extended Deductive Databases in DATALOG* 37
Dietmar Seipel

Answer Set Programming

- xpanda: A (Simple) Preprocessor for Adding Multi-valued Propositions to ASP 51
Martin Gebser, Henrik Hinrichs, Torsten Schaub and Sven Thiele
- Existential Quantifiers in the Rule Body 59
Pedro Cabalar
- Kato: A Plagiarism-Detection Tool for Answer-Set Programs 75
Johannes Oetsch, Martin Schwengerer and Hans Tompits

Theory of Logic Programming

- A Paraconsistent Semantics for Generalized Logic Programs 81
Heinrich Herre and Axel Hummel
- Stationary Generated Models of Generalized Logic Programs 95
Heinrich Herre and Axel Hummel
- Constraint-Based Abstraction of a Model Checker for Infinite State Systems 109
Gourinath Banda and John Gallagher
- Range Restriction for General Formulas 125
Stefan Brass

Constraint Handling Rules

Transforming Imperative Algorithms to Constraint Handling Rules 139
Slim Abdennadher, Haythem Ismail and Frederick Khoury

Persistent Constraints in Constraint Handling Rules 155
Hariolf Betz, Frank Raiser and Thom Frühwirth

Practice of Logic Programming

A Tool for Generating Partition Schedules of Multiprocessor Systems 167
Hans-Joachim Goltz and Norbert Pieth

Efficiency of Difference-List Programming 177
Ulrich Geske and Hans-Joachim Goltz