Jean-Daniel Zucker
Lorenza Saitta (Eds.)

LNAI 3607

# Abstraction, Reformulation and Approximation

6th International Symposium, SARA 2005
Airth Castle, Scotland, UK, July 2005
Proceedings

Springer

TP18-53
S243
2005

Jean-Daniel Zucker   Lorenza Saitta (Eds.)

# Abstraction, Reformulation and Approximation

6th International Symposium, SARA 2005
Airth Castle, Scotland, UK, July 26-29, 2005
Proceedings

华南理工大学
图书馆藏书

E200501657

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Jean-Daniel Zucker
LIM&BIO, EPML-CNRS 32
Université Paris 13
74, rue Marcel Cachin, 93017 Bobigny, France
E-mail: zucker@limbio-paris13.org

Lorenza Saitta
Università del Piemonte Orientale
Dipartimento di Informatica
Via Bellini 25/G, 15100 Alessandria, Italy
E-mail: saitta@al.unipmn.it

Library of Congress Control Number: 2005929387

CR Subject Classification (1998): I.2, F.4.1, F.3

ISSN 0302-9743
ISBN-10 3-540-27872-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-27872-6 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11527862 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3607

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

# Preface

This volume contains the proceedings of the 6th Symposium on Abstraction, Reformulation and Approximation (SARA 2005). The symposium was held at Airth Castle, Scotland, UK, from July 26th to 29th, 2005, just prior to the IJCAI 2005 conference in Edinburgh. Previous SARA symposia took place at Jackson Hole in Wyoming, USA (1994), Ville d'Estrel in Qubec, Canada (1995), Asilomar in California, USA (1998), Horseshoe Bay, Texas, USA (2000), and Kananaskis, Alberta, Canada (2002). This was then the first time that the symposium was held in Europe. Continuing the tradition started with SARA 2000, the proceedings have been published in the LNAI series of Springer.

Abstractions, reformulations and approximations (AR&A) have found applications in a variety of disciplines and problems, including constraint satisfaction, design, diagnosis, machine learning, planning, qualitative reasoning, scheduling, resource allocation and theorem proving, but are also deeply rooted in philosophy and cognitive science. The papers in this volume capture a cross-section of the various facets of the field and of its applications. One of the primary uses of AR&A is oriented to overcome computational intractability. AR&A techniques, however, have also proved useful for knowledge acquisition, explanation and other applications, as papers in this volume also illustrate.

The aim of SARA is to provide a forum for intensive and friendly interaction among researchers in all areas of AI in which an interest in the different aspects of AR&A exist. The diverse backgrounds of participants at this and previous meetings led to rich and lively exchanges of ideas, allowed the comparisons of goals, techniques and paradigms, and helped identify important research issues and engineering hurdles. SARA has always invited distinguished members of the research community to present keynote talks. SARA 2005 was no exception to this rule with invited talks from Rada Chirkova of the North Carolina State University at Raleigh, USA Aristide Mingozzi of the University of Bologna, Italy, and Robert Zimmer of Goldsmiths College, University of London and Goldsmiths Digital Studios, London.

We would like to thank the authors of all the submitted papers and research summaries, the referees, the invited speakers, the Program Committee members for all their time and effort, and, of course, all the attendees. We also thank the members of the Steering Committee for their advice along the way. In addition, a great "merci" to the Local Chair Ian Miguel and to all those who contributed to the organization of SARA 2005, in particular Mélanie Courtine.

Paris, May 19, 2005

Jean Daniel Zucker
Lorenza Saitta

此为试读,需要完整PDF请访问: www.ertongbook.com

# Organization

http://sara2005.limbio-paris13.org

SARA 2005 was organized by Ian Miguel.

## Executive Committee

| | |
|---|---|
| Conference Chair | Jean-Daniel Zucker, University of Paris 13<br>Lorenza Saitta, Universitá del Piemonte Orientale |
| Organizing Chair | Ian Miguel, University of York |
| Proceeding Chair | Mélanie Courtine, University of Paris 13 |

## Program Committee

J. Christopher Beck, University of Toronto
Berthe Y. Choueiry, University of Nebraska-Lincoln
Stefan Edelkamp, Albert Ludwigs University Freiburg
Tom Ellman, Vassar College
Jérôme Euzenat, INRIA Rhône-Alpes
Mike Genesereth, Stanford University
Robert C. Holte, University of Alberta
Daniel Kayser, University of Paris Nord
Sven Koenig, University of Southern California
Michael Lowry, NASA Ames Research Center
Hiroshi Motoda, Osaka University
Pandurang Nayak, PurpleYogi.com
Doina Precup, McGill University
Peter Revesz, University of Nebraska-Lincoln
Marie-Christine.Rousset, University of Paris XI
Bart Selman, Cornell University
Barbara Smith, University College Cork
Miroslav Velev, CMU
Toby Walsh, Cork Constraint Computation Centre, University College Cork
Robert Zimmer, Goldsmiths College, University of London
Weixiong Zhang, Washington University in St Louis

## Steering Committee

Berthe Y. Choueiry, University of Nebraska-Lincoln
Tom Ellman, Vassar College

Mike Genesereth, Stanford University
Fausto Giunchiglia, University of Trento and ITC-IRST
Alon Halevy, University of Washington
Robert Holte, University of Alberta
Sven Koenig, Georgia Institute of Technology
Michael Lowry, NASA Ames Research Center
Pandurang Nayak, PurpleYogi.com
Jeffrey Van Baalen, University of Wyoming
Toby Walsh, Cork Constraint Computation Centre, University College Cork

## Supplementary Referees

Yann Chevaleyre
Olivier Cogis
James Ezick
Attilio Giordana
Joel Gompert
Brahim Hnich
Shahid Jabbar
Anagh Lal
Dejan Nickovic
Francesca Rossi
Peter Szilagyi
Shang-Wen Cheng

# Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005.

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005.

Vol. 3554: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005.

Vol. 3538: L. Ardissono, P. Brna, A. Mitrovic (Eds.), User Modeling 2005. XVI, 533 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005.

Vol. 3518: T.B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V. A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

Vol. 3492: P. Blache, E. Stabler, J. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3476: J. Leite, A. Omicini, P. Torroni, P. Yolum (Eds.), Declarative Agent Languages and Technologies II. XII, 289 pages. 2005.

Vol. 3464: S.A. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (Eds.), Engineering Self-Organising Systems. XIII, 299 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3451: M.-P. Gleizes, A. Omicini, F. Zambonelli (Eds.), Engineering Societies in the Agents World. XIII, 349 pages. 2005.

Vol. 3446: T. Ishida, L. Gasser, H. Nakashima (Eds.), Massively Multi-Agent Systems I. XI, 349 pages. 2005.

Vol. 3445: G. Chollet, A. Esposito, M. Faundez-Zanuy, M. Marinaro (Eds.), Nonlinear Speech Modeling and Applications. XIII, 433 pages. 2005.

Vol. 3438: H. Christiansen, P.R. Skadhauge, J. Villadsen (Eds.), Constraint Solving and Language Processing. VIII, 205 pages. 2005.

Vol. 3430: S. Tsumoto, T. Yamaguchi, M. Numao, H. Motoda (Eds.), Active Mining. XII, 349 pages. 2005.

Vol. 3419: B. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Constraint Satisfaction and Constraint Logic Programming. X, 217 pages. 2005.

Vol. 3416: M. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005.

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005.

Vol. 3398: D.-K. Baik (Ed.), Systems Modeling and Simulation: Theory and Applications. XIV, 733 pages. 2005.

Vol. 3397: T.G. Kim (Ed.), Artificial Intelligence and Simulation. XV, 711 pages. 2005.

Vol. 3396: R.M. van Eijk, M.-P. Huget, F. Dignum (Eds.), Agent Communication. X, 261 pages. 2005.

Vol. 3394: D. Kudenko, D. Kazakov, E. Alonso (Eds.), Adaptive Agents and Multi-Agent Systems II. VIII, 313 pages. 2005.

Vol. 3392: D. Seipel, M. Hanus, U. Geske, O. Bartenstein (Eds.), Applications of Declarative Programming and Knowledge Management. X, 309 pages. 2005.

Vol. 3374: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems. X, 279 pages. 2005.

Vol. 3371: M.W. Barley, N. Kasabov (Eds.), Intelligent Agents and Multi-Agent Systems. X, 329 pages. 2005.

Vol. 3369: V.R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.), Law and the Semantic Web. XII, 249 pages. 2005.

Vol. 3366: I. Rahwan, P. Moraitis, C. Reed (Eds.), Argumentation in Multi-Agent Systems. XII, 263 pages. 2005.

Vol. 3359: G. Grieser, Y. Tanaka (Eds.), Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets. XIV, 257 pages. 2005.

Vol. 3346: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 249 pages. 2005.

Vol. 3345: Y. Cai (Ed.), Ambient Intelligence for Scientific Discovery. XII, 311 pages. 2005.

Vol. 3343: C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, T. Barkowsky (Eds.), Spatial Cognition IV. XIII, 519 pages. 2005.

Vol. 3339: G.I. Webb, X. Yu (Eds.), AI 2004: Advances in Artificial Intelligence. XXII, 1272 pages. 2004.

Vol. 3336: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. X, 523 pages. 2004.

Vol. 3327: Y. Shi, W. Xu, Z. Chen (Eds.), Data Mining and Knowledge Management. XIII, 263 pages. 2005.

Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004. XX, 987 pages. 2004.

Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004.

Vol. 3301: G. Kern-Isberner, W. Rödder, F. Kulmann (Eds.), Conditionals, Information, and Inference. XII, 219 pages. 2005.

Vol. 3276: D. Nardi, M. Riedmiller, C. Sammut, J. Santos-Victor (Eds.), RoboCup 2004: Robot Soccer World Cup VIII. XVIII, 678 pages. 2005.

Vol. 3275: P. Perner (Ed.), Advances in Data Mining. VIII, 173 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004.

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004.

Vol. 3259: J. Dix, J. Leite (Eds.), Computational Logic in Multi-Agent Systems. XII, 251 pages. 2004.

Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004.

Vol. 3248: K.-Y. Su, J. Tsujii, J.-H. Lee, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2004. XVIII, 817 pages. 2005.

Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004.

Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004.

Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), Advances in Natural Language Processing. XII, 488 pages. 2004.

Vol. 3229: J.J. Alferes, J. Leite (Eds.), Logics in Artificial Intelligence. XIV, 744 pages. 2004.

Vol. 3228: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff (Eds.), Formal Approaches to Agent-Based Systems. VIII, 290 pages. 2004.

Vol. 3215: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LVII, 906 pages. 2004.

Vol. 3214: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LVIII, 1302 pages. 2004.

Vol. 3213: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LVIII, 1280 pages. 2004.

Vol. 3209: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web. IX, 201 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004.

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004.

Vol. 3192: C. Bussler, D. Fensel (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 522 pages. 2004.

Vol. 3191: M. Klusch, S. Ossowski, V. Kashyap, R. Unland (Eds.), Cooperative Information Agents VIII. XI, 303 pages. 2004.

Vol. 3187: G. Lindemann, J. Denzinger, I.J. Timm, R. Unland (Eds.), Multiagent System Technologies. XIII, 341 pages. 2004.

Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning. IX, 241 pages. 2004.

Vol. 3171: A.L.C. Bazzan, S. Labidi (Eds.), Advances in Artificial Intelligence – SBIA 2004. XVII, 548 pages. 2004.

Vol. 3159: U. Visser, Intelligent Information Integration for the Semantic Web. XIV, 150 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advances in Case-Based Reasoning. XIII, 822 pages. 2004.

¥490.88元

# Table of Contents

## Full Papers

## Extended Abstracts

此为试读，需要完整PDF请访问：www.ertongbook.com

## Invited Talks

## Research Summaries

# Verifying the Incorrectness of Programs and Automata*

Scot Anderson and Peter Revesz

Department of Computer Science and Engineering,
University of Nebraska-Lincoln, Lincoln, NE 68588, USA
{scot, revesz}@cse.unl.edu

**Abstract.** Verification of the incorrectness of programs and automata needs to be taken as seriously as the verification of correctness. However, there are no good general methods that always terminate and prove incorrectness. We propose one general method based on a *lower bound* approximation of the semantics of programs and automata. Based on the lower-bound approximation, it becomes easy to check whether certain error states are reached. This is in contrast to various abstract interpretation techniques that make an *upper bound* approximation of the semantics and test that the error states are not reached. The precision of our lower bound approximation is controlled by a single parameter that can be adjusted by the user of the MLPQ system in which the approximation method is implemented. As the value of the parameter decreases the implementation results in a finer program semantics approximation but requires a longer evaluation time. However, for all input parameter values the program is guaranteed to terminate. We use the lower bound approximation to verify the incorrectness of a subway train control automaton. We also use the lower bound approximation for a problem regarding computer security via trust management programs. We propose a trust management policy language extending earlier work by Li and Mitchell. Although, our trust management programming language is Turing-complete, programs in this language have semantics that lend themselves naturally to a lower-bound approximation. Namely, the lower bound approximation is such that no unwarranted authorization is given at any time, although some legitimate access may be denied.

## 1 Introduction

Testing the correctness of a program or an automaton can be done by finding an *upper approximation* of its semantics. If the upper approximation *does not* contain the error states needed to be checked, then the automaton can be said to be correct. However, if the upper approximation *contains* the error states, then the actual program or automaton may still be correct.

* This research was supported in part by NSF grant EIA-0091530 and a NASA Space and EPSCoR grant.

J.-D. Zucker and L. Saitta (Eds.): SARA 2005, LNAI 3607, pp. 1–13, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Similarly, if the *lower bound* approximation of the semantics contains an error state, then we know that it is incorrect. If it does not, then the program may still be incorrect.

Hence an *upper bound* may be good to verify that a program is correct, while a *lower bound* may be good to verify that it is incorrect. The *verification of incorrectness* is just as important in practice as the verification of correctness, because many users are reluctant to change incorrect and expensive programs unless those are proven incorrect. For example, if a banking system allows invalid access to some bank accounts, then a lower bound approximation would be needed to verify the incorrectness.

Until recently, in the verification area the focus was in verifying correctness using *abstract interpretation* [8, 16, 22] or *model checking* [1, 5, 9, 30, 36]. In contrast, in this paper, we focus on verifying incorrectness.

Verifying incorrectness is needed when we suspect a program to be incorrect, and we want to prove that it is indeed incorrect. For example, if there is an accident with a space shuttle, then we need to find what caused it. Was it caused by an incorrect program?

There are many reasons that a program may be suspected to be incorrect. For example, a program that fails a verification for correctness using abstract interpretation or model checking would be suspicious.

There are some problems that naturally lend themselves to a lower-bound approximation. For example, the semantics of a computer security system would contain the facts that describe who gets access to which resource at what time. In this case a lower-bound approximation is meaningful, conservative, and safe to use. That is, it never gives unwarranted authorizations, although some legitimate access may be denied at certain time instances. For example, not being able to access one's own bank account at a particular time is frustrating, but it is certainly less frustrating than if someone else, who should not, can access it.

We use the above idea in proposing a Turing-complete extension of the *trust management* language RT [25, 26, 27], which is a recent approach to computer security in a distributed environment. The latest version of the RT language uses Datalog but with simpler constraints than we allow in this paper. We choose the RT trust management family of languages *as an example* of how to use constraint database approximation techniques in other areas beyond database systems where lower-bound approximations are meaningful. (See the survey [15] and the recent article [24] about trust management in general.)

The rest of this paper is organized as follows. Section 2 gives a brief review of constraint database approximation theory and its implementation in the MLPQ constraint database system [38]. Section 3 applies the approximation method to verify the incorrectness of an automaton. Section 4 applies the approximation method to find a safe evaluation of a trust management program. Section 5 discusses some related work. Finally, Section 6 gives some conclusions and future work.

## 2 Review of Constraint Database Approximation Theory

The *constraint logic programming* languages proposed by Jaffar and Lassez [17], whose work led to CLP(R) [19], by Colmerauer [7] within Prolog III, and by Dincbas et al. [10] within CHIP, were Turing-complete. Kanellakis, Kuper, and Revesz [20, 21] considered those to be impractical for use in database systems and proposed less expressive *constraint query languages* that have nice properties in terms of guaranteed and efficient evaluations. Many researchers advocated extensions of those languages while trying to keep termination guaranteed. For example, the least fixed point semantics of Datalog (Prolog without function symbols and negation) with integer gap-order constraint programs can be always evaluated in a finite constraint database representation [33].[1]

With gap-order constraints many NP-complete problems can be expressed that cannot be expressed in Datalog without constraints. However, even Datalog with addition constraints, which seems only a slight extension, is already Turing-complete. Hence Revesz [35] introduced an approximate evaluation for Datalog with addition constraints.

This approximation is different from *abstract interpretation* methods (for a recent review see [8]). The main difference is that, at least in theory, in [35] both a lower and an upper bound approximation of the least fixed point can be arbitrarily close to the actual least fixed point with the decrease of a single parameter towards $-\infty$. The decrease indirectly increases the running time.

Below we focus on the definitions that are relevant to approximations. The reader can find more details in the surveys [18, 34] and the books [23, 28, 37] about constraint logic programming and constraint databases.

**Definition 1.** *Addition constraints [37] have the form*

$$\pm x \pm y \ \theta \ b \quad \text{or} \quad \pm x \ \theta \ b$$

*where $x$ and $y$ are integer variables and $b$ is an integer constant, called a* bound, *and $\theta$ is either $\geq$ or $>$.*

In the following we will also use $x = b$ as an abbreviation for the conjunction of $x \geq b$ and $-x \geq -b$. Similarly, we use $x + y = b$ as an abbreviation for the conjunction of $x + y \geq b$ and $-x - y \geq -b$.

Each *constraint database* is a finite set of *constraint tuples* of the form:

$$R(x_1, \ldots, x_k) \ :- \ C_1, \ldots, C_m.$$

where $R$ is a $k$-ary relation symbol, each $x_i$ for $1 \leq i \leq k$ is an integer variable or constant, and each $C_j$ for $1 \leq j \leq m$ is an addition constraint over the variables. The meaning of a constraint tuple is that each substitution of the variables by integer constants that makes each $C_j$ on the right hand side of $:-$ true is a $k$-tuple that is in relation $R$.

[1] A gap-order is a constraint of the form $x - y \geq c$ or $\pm x \geq c$ where $x$ and $y$ are variables and $c$ is a non-negative integer constant.

A *Datalog program* consists of a finite set of constraint tuples and rules of the form:

$$R_0(x_1,\ldots,x_k) \;:-\; R_1(x_{1,1},\ldots,x_{1,k_1}),\ldots,R_n(x_{n,1},\ldots,x_{n,k_n}),\; C_1,\ldots,C_m.$$

where each $R_i$ is a relation name, and the $x$s are either integer variables or constants, and each $C_j$ is an addition constraint over the $x$s. The meaning of the rule is that if for some substitution of the variables by integer constants each $R_i$ and $C_j$ on the right hand side of $: -$ is true, then the left hand side is also true.

A *model* of a Datalog program is an assignment to each $k$-arity relation symbol $R$ within the program a subset of $\mathbb{Z}^k$ where $\mathbb{Z}$ is the set of integers such that each rule holds for each possible substitution. The *least fixed point* semantics of a Datalog program contains the intersection of all the models of the program.

It is easy to express in Datalog [37] with addition constraints a program that will not terminate using a standard bottom-up evaluation [37]. Consider the following Datalog with addition constraint program:

$$\begin{array}{lll} D(x,y,z) & :\!\!- & x-y=0,\;\; z=0. \\ D(x',y,z') & :\!\!- & D(x,y,z),\;\; x'-x=1,\;\; z'-z=1. \end{array} \tag{1}$$

This expresses that the *Difference* of $x$ and $y$ is $z$. Further, based on (1) we can also express a *Multiplication* relation as follows:

$$\begin{array}{lll} M(x,y,z) & :\!\!- & x=0,\;\; y=0,\;\; z=0. \\ M(x',y,z') & :\!\!- & M(x,y,z),\;\; D(z',z,y),\;\; x'-x=1. \\ M(x,y',z') & :\!\!- & M(x,y,z),\;\; D(z',z,x),\;\; y'-y=1. \end{array} \tag{2}$$

Intuitively, a standard bottom-up evaluation derives additional constraint tuples until a certain saturation is reached, and the saturation state represents in a constraint database form the least fixed point. We omit the precise definition of bottom-up evaluation of Datalog with constraint programs, because it is not needed for the rest of this paper. It is enough to note that the simple Datalog program that consists of the above two sets of rules never terminates in a standard bottom-up evaluation.

In fact, with these two relations we can express any integer polynomial equation (see Example 3). Since integer polynomial equations are unsolvable in general [29], no algorithm would be able to evaluate precisely the least fixed point semantics of the Datalog program. Hence the situation we face is not just a particular problem with the standard bottom-up evaluation, but a problem that is inherent to the least fixed point semantics of Datalog with addition constraints.

Revesz [35] introduced two methods for approximating the least fixed point evaluation by modifying the standard bottom-up evaluation.

**Definition 2.** *Let $l < 0$ be any fixed integer constant. We change in the constraint tuples the value of any bound $b$ to be $max(b,l)$. Given a Datalog program $P$ the result of a bottom-up evaluation of $P$ using this modification is denoted $P_l$.*

此为试读,需要完整PDF请访问: www.ertongbook.com