Jürgen Dix
Stephen J. Hegner (Eds.)

LNCS 3861

Foundations of Information and Knowledge Systems

4th International Symposium, FoIKS 2006
Budapest, Hungary, February 2006
Proceedings

Springer

Jürgen Dix Stephen J. Hegner (Eds.)

Foundations of Information and Knowledge Systems

4th International Symposium, FoIKS 2006
Budapest, Hungary, February 14-17, 2006
Proceedings

Springer

Volume Editors

Jürgen Dix
Clausthal University of Technology
Dept. of Computer Science
Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany
E-mail: dix@tu-clausthal.de

Stephen J. Hegner
Umeå University
Department of Computing Science
90187 Umeå, Sweden
E-mail: hegner@cs.umu.se

Library of Congress Control Number: 2005939041

CR Subject Classification (1998): H.2, H.3, H.5, I.2.4, F.3.2, G.2

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-31782-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-31782-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11663881 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3861

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–3770

please contact your bookseller or Springer

Vol. 3870: S. Spaccapietra, P. Atzeni, W.W. Chu, T. Catarci, K.P. Sycara (Eds.), Journal on Data Semantics V. XIII, 237 pages. 2006.

Vol. 3863: M. Kohlhase (Ed.), Mathematical Knowledge Management. XI, 405 pages. 2006. (Sublibrary LNAI).

Vol. 3861: J. Dix, S.J. Hegner (Eds.), Foundations of Information and Knowledge Systems. X, 331 pages. 2006.

Vol. 3860: D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006. XI, 365 pages. 2006.

Vol. 3857: M. Fossorier, H. Imai, S. Lin, A. Poli (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. XI, 350 pages. 2006.

Vol. 3855: E. A. Emerson, K.S. Namjoshi (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 443 pages. 2005.

Vol. 3853: A.J. Ijspeert, T. Masuzawa, S. Kusumoto (Eds.), Biologically Inspired Approaches to Advanced Information Technology. XIV, 388 pages. 2006.

Vol. 3852: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision - ACCV 2006, Part II. XXXI, 977 pages. 2005.

Vol. 3851: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision - ACCV 2006, Part I. XXXI, 973 pages. 2006.

Vol. 3850: R. Freund, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 371 pages. 2006.

Vol. 3848: J.-F. Boulicaut, L. De Raedt, H. Mannila (Eds.), Constraint-Based Mining and Inductive Databases. X, 401 pages. 2006. (Sublibrary LNAI).

Vol. 3847: K.P. Jantke, A. Lunzer, N. Spyratos, Y. Tanaka (Eds.), Federation over the Web. X, 215 pages. 2006. (Sublibrary LNAI).

Vol. 3844: J.-M. Bruel (Ed.), Satellite Events at the MoDELS 2005 Conference. XIII, 360 pages. 2006.

Vol. 3843: P. Healy, N.S. Nikolov (Eds.), Graph Drawing. XVII, 536 pages. 2006.

Vol. 3842: H.T. Shen, J. Li, M. Li, J. Ni, W. Wang (Eds.), Advanced Web and Network Technologies, and Applications. XXVII, 1057 pages. 2006.

Vol. 3841: X. Zhou, J. Li, H.T. Shen, M. Kitsuregawa, Y. Zhang (Eds.), Frontiers of WWW Research and Development - APWeb 2006. XXIV, 1223 pages. 2006.

Vol. 3840: M. Li, B. Boehm, L.J. Osterweil (Eds.), Unifying the Software Process Spectrum. XVI, 522 pages. 2006.

Vol. 3839: J.-C. Filliâtre, C. Paulin-Mohring, B. Werner (Eds.), Types for Proofs and Programs. VIII, 275 pages. 2006.

Vol. 3838: A. Middeldorp, V. van Oostrom, F. van Raamsdonk, R. de Vrijer (Eds.), Processes, Terms and Cycles: Steps on the Road to Infinity. XVIII, 639 pages. 2005.

Vol. 3837: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks. IX, 307 pages. 2005.

Vol. 3836: J.-M. Pierson (Ed.), Data Management in Grids. X, 143 pages. 2006.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005. (Sublibrary LNAI).

Vol. 3834: D.G. Feitelson, E. Frachtenberg, L. Rudolph, U. Schwiegelshohn (Eds.), Job Scheduling Strategies for Parallel Processing. VIII, 283 pages. 2005.

Vol. 3833: K.-J. Li, C. Vangenot (Eds.), Web and Wireless Geographical Information Systems. XI, 309 pages. 2005.

Vol. 3832: D. Zhang, A.K. Jain (Eds.), Advances in Biometrics. XX, 796 pages. 2005.

Vol. 3831: J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, J. Štuller (Eds.), SOFSEM 2006: Theory and Practice of Computer Science. XV, 576 pages. 2006.

Vol. 3829: P. Pettersson, W. Yi (Eds.), Formal Modeling and Analysis of Timed Systems. IX, 305 pages. 2005.

Vol. 3828: X. Deng, Y. Ye (Eds.), Internet and Network Economics. XVII, 1106 pages. 2005.

Vol. 3827: X. Deng, D.-Z. Du (Eds.), Algorithms and Computation. XX, 1190 pages. 2005.

Vol. 3826: B. Benatallah, F. Casati, P. Traverso (Eds.), Service-Oriented Computing - ICSOC 2005. XVIII, 597 pages. 2005.

Vol. 3824: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embedded and Ubiquitous Computing – EUC 2005. XXIII, 1204 pages. 2005.

Vol. 3823: T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, L.T. Yang (Eds.), Embedded and Ubiquitous Computing – EUC 2005 Workshops. XXXII, 1317 pages. 2005.

Vol. 3822: D. Feng, D. Lin, M. Yung (Eds.), Information Security and Cryptology. XII, 420 pages. 2005.

Vol. 3821: R. Ramanujam, S. Sen (Eds.), FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science. XIV, 566 pages. 2005.

Vol. 3820: L.T. Yang, X. Zhou, W. Zhao, Z. Wu, Y. Zhu, M. Lin (Eds.), Embedded Software and Systems. XXVIII, 779 pages. 2005.

Vol. 3819: P. Van Hentenryck (Ed.), Practical Aspects of Declarative Languages. X, 231 pages. 2005.

Vol. 3818: S. Grumbach, L. Sui, V. Vianu (Eds.), Advances in Computer Science – ASIAN 2005. XIII, 294 pages. 2005.

Vol. 3817: M. Faundez-Zanuy, L. Janer, A. Esposito, A. Satue-Villar, J. Roure, V. Espinosa-Duro (Eds.), Nonlinear Analyses and Algorithms for Speech Processing. XII, 380 pages. 2006. (Sublibrary LNAI).

Vol. 3816: G. Chakraborty (Ed.), Distributed Computing and Internet Technology. XXI, 606 pages. 2005.

Vol. 3815: E.A. Fox, E.J. Neuhold, P. Premsmit, V. Wuwongse (Eds.), Digital Libraries: Implementing Strategies and Sharing Experiences. XVII, 529 pages. 2005.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005. (Sublibrary LNAI).

Vol. 3813: R. Molva, G. Tsudik, D. Westhoff (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. VIII, 219 pages. 2005.

Vol. 3810: Y.G. Desmedt, H. Wang, Y. Mu, Y. Li (Eds.), Cryptology and Network Security. XI, 349 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005. (Sublibrary LNAI).

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005. (Sublibrary LNAI).

Vol. 3807: M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005 Workshops. XV, 275 pages. 2005.

Vol. 3806: A.H. H. Ngu, M. Kitsuregawa, E.J. Neuhold, J.-Y. Chung, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005. XXI, 771 pages. 2005.

Vol. 3805: G. Subsol (Ed.), Virtual Storytelling. XII, 289 pages. 2005.

Vol. 3804: G. Bebis, R. Boyle, D. Koracin, B. Parvin (Eds.), Advances in Visual Computing. XX, 755 pages. 2005.

Vol. 3803: S. Jajodia, C. Mazumdar (Eds.), Information Systems Security. XI, 342 pages. 2005.

Vol. 3802: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part II. XLII, 1166 pages. 2005. (Sublibrary LNAI).

Vol. 3801: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part I. XLI, 1122 pages. 2005. (Sublibrary LNAI).

Vol. 3799: M. A. Rodríguez, I.F. Cruz, S. Levashkin, M.J. Egenhofer (Eds.), GeoSpatial Semantics. X, 259 pages. 2005.

Vol. 3798: A. Dearle, S. Eisenbach (Eds.), Component Deployment. X, 197 pages. 2005.

Vol. 3797: S. Maitra, C. E. V. Madhavan, R. Venkatesan (Eds.), Progress in Cryptology - INDOCRYPT 2005. XIV, 417 pages. 2005.

Vol. 3796: N.P. Smart (Ed.), Cryptography and Coding. XI, 461 pages. 2005.

Vol. 3795: H. Zhuge, G.C. Fox (Eds.), Grid and Cooperative Computing - GCC 2005. XXI, 1203 pages. 2005.

Vol. 3794: X. Jia, J. Wu, Y. He (Eds.), Mobile Ad-hoc and Sensor Networks. XX, 1136 pages. 2005.

Vol. 3793: T. Conte, N. Navarro, W.-m.W. Hwu, M. Valero, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XIII, 317 pages. 2005.

Vol. 3792: I. Richardson, P. Abrahamsson, R. Messnarz (Eds.), Software Process Improvement. VIII, 215 pages. 2005.

Vol. 3791: A. Adi, S. Stoutenburg, S. Tabet (Eds.), Rules and Rule Markup Languages for the Semantic Web. X, 225 pages. 2005.

Vol. 3790: G. Alonso (Ed.), Middleware 2005. XIII, 443 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005. (Sublibrary LNAI).

Vol. 3788: B. Roy (Ed.), Advances in Cryptology - ASIACRYPT 2005. XIV, 703 pages. 2005.

Vol. 3787: D. Kratsch (Ed.), Graph-Theoretic Concepts in Computer Science. XIV, 470 pages. 2005.

Vol. 3785: K.-K. Lau, R. Banach (Eds.), Formal Methods and Software Engineering. XIV, 496 pages. 2005.

Vol. 3784: J. Tao, T. Tan, R.W. Picard (Eds.), Affective Computing and Intelligent Interaction. XIX, 1008 pages. 2005.

Vol. 3783: S. Qing, W. Mao, J. Lopez, G. Wang (Eds.), Information and Communications Security. XIV, 492 pages. 2005.

Vol. 3782: K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, T.R. Roth-Berghofer (Eds.), Professional Knowledge Management. XXIII, 739 pages. 2005. (Sublibrary LNAI).

Vol. 3781: S.Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, D. Zhang (Eds.), Advances in Biometric Person Authentication. XI, 250 pages. 2005.

Vol. 3780: K. Yi (Ed.), Programming Languages and Systems. XI, 435 pages. 2005.

Vol. 3779: H. Jin, D. Reed, W. Jiang (Eds.), Network and Parallel Computing. XV, 513 pages. 2005.

Vol. 3778: C. Atkinson, C. Bunse, H.-G. Gross, C. Peper (Eds.), Component-Based Software Development for Embedded Systems. VIII, 345 pages. 2005.

Vol. 3777: O.B. Lupanov, O.M. Kasim-Zade, A.V. Chaskin, K. Steinhöfel (Eds.), Stochastic Algorithms: Foundations and Applications. VIII, 239 pages. 2005.

Vol. 3776: S.K. Pal, S. Bandyopadhyay, S. Biswas (Eds.), Pattern Recognition and Machine Intelligence. XXIV, 808 pages. 2005.

Vol. 3775: J. Schönwälder, J. Serrat (Eds.), Ambient Networks. XIII, 281 pages. 2005.

Vol. 3774: G. Bierman, C. Koch (Eds.), Database Programming Languages. X, 295 pages. 2005.

Vol. 3773: A. Sanfeliu, M.L. Cortés (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XX, 1094 pages. 2005.

Vol. 3772: M.P. Consens, G. Navarro (Eds.), String Processing and Information Retrieval. XIV, 406 pages. 2005.

Vol. 3771: J.M.T. Romijn, G.P. Smith, J. van de Pol (Eds.), Integrated Formal Methods. XI, 407 pages. 2005.

¥410.00元

Preface

This volume contains the papers presented at the 4th International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2006), which was held at the Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, Hungary, from February 14 to 17, 2006.

FoIKS is a biennial event with a focus on the theoretical foundations of information and knowledge systems. The goal is to bring together researchers working on the theoretical foundations of information and knowledge systems, as well as to attract researchers working in mathematical fields such as discrete mathematics, combinatorics, logics and finite model theory who are interested in applying their theories to research on database and knowledge base theory.

FoIKS took up the tradition of the conference series Mathematical Fundamentals of Database Systems (MFDBS), which enabled East-West collaboration in the field of database theory. The first FoIKS symposium was held in Burg, Spreewald (Germany) in 2000, the second FoIKS symposium was held in Salzau Castle (Germany) in 2002, and the third FoIKS symposium was held in Vienna (Austria) in 2004. Former MFDBS conferences were held in Dresden (Germany) in 1987, Visegrád (Hungary) in 1989 and in Rostock (Germany) in 1991. Proceedings of these previous MFDBS and FoIKS events were published by Springer as volumes 305, 364, 495, 1762, 2284, and 2942 of the LNCS series, respectively.

The FoIKS symposium is intended to be a forum for intensive discussions. For this reason the time slots for long and short contributions are 50 and 30 minutes, respectively, followed by 20 and 10 minutes for discussions, respectively. Each such discussion is led by the author of another paper, who is asked in advance to prepare a focused set of questions and points for further elaboration.

The FoIKS 2006 call for papers solicited contributions dealing with the foundational aspects of information and knowledge systems, including the following topics:

- Mathematical Foundations: discrete methods, Boolean functions, finite model theory
- Database Design: formal models, dependency theory, schema translations, desirable properties
- Query Languages: expressiveness, computational and descriptive complexity, query languages for advanced data models, classifications of computable queries
- Semi-structured Databases and WWW: models of Web databases, querying semi-structured databases, Web transactions and negotiations
- Security in Data and Knowledge Bases: cryptography, steganography, information hiding
- Integrity and Constraint Management: verification, validation, and enforcement of consistency, triggers

- Information Integration: heterogeneous data, views, schema dominance and equivalence
- Database and Knowledge Base Dynamics: models of transactions, models of interaction, updates, consistency preservation, concurrency control
- Intelligent Agents: multi-agent systems, autonomous agents, foundations of software agents, cooperative agents
- Logics in Databases and AI: non-classical logics, spatial and temporal logics, probabilistic logics, deontic logic, logic programming
- Knowledge Representation: planning, description logics, knowledge and belief, belief revision and update, non-monotonic formalisms, uncertainty
- Reasoning Techniques: theorem proving, abduction, induction, constraint satisfaction, common-sense reasoning, probabilistic reasoning, reasoning about actions

The Program Committee received 54 submissions. Each paper was carefully reviewed by at least two experienced referees, and most of the papers were reviewed by three referees. Fourteen papers were chosen for long presentations and three papers for short presentations. This volume contains polished versions of these papers with respect to the comments made in the reviews. The best papers will be selected for further extension and publishing in a special issue of the journal *Annals of Mathematics and Artificial Intelligence.*

We would like to thank everyone involved with FoIKS 2006 for their contribution to the success of the symposium.

Jürgen Dix
Stephen J. Hegner

Organization

Program Committee Co-chairs

Jürgen Dix	Clausthal University of Technology (Germany)
Stephen J. Hegner	Umeå University (Sweden)

Program Committee

Peter Baumgartner	Australian National University (Australia)
Leopoldo Bertossi	Carleton University (Canada)
Joachim Biskup	University of Dortmund (Germany)
Piero Bonatti	University of Naples (Italy)
Stefan Brass	University of Halle–Wittenberg (Germany)
Guozhu Dong	Wright State University (USA)
Thomas Eiter	Vienna University of Technology (Austria)
Burkhard Freitag	University of Passau (Germany)
Floris Geerts	University of Edinburgh (UK) and University of Limburg (Belgium)
Sven Hartmann	Massey University (New Zealand)
Lauri Hella	University of Tampere (Finland)
Gyula Katona	Alfréd Rényi Institute of Mathematics (Hungary)
Gabriele Kern-Isberner	University of Dortmund (Germany)
Hans-Joachim Klein	University of Kiel (Germany)
Christoph Koch	Saarland University (Germany)
Nicola Leone	University of Calabria (Italy)
Mark Levene	Birbeck University of London (UK)
Ilkka Niemelä	Helsinki University of Technology (Finland)
Domenico Saccà	University of Calabria (Italy)
Attila Sali	Alfréd Rényi Institute of Mathematics (Hungary)
Vladimir Sazonov	University of Liverpool (UK)
Nicole Schweikardt	Humboldt University (Germany)
Dietmar Seipel	University of Würzburg (Germany)
Nicolas Spyratos	University of Paris-South (France)
Bernhard Thalheim	University of Kiel (Germany)
Rodney Topor	Griffith University (Australia)
José María Turull-Torres	Massey University (New Zealand)
Andrei Voronkov	University of Manchester (UK)
Marina de Vos	University of Bath (UK)

External Referees

Marcelo Arenas
Matthias Beck
Bernhard Beckert
Alfredo Cuzzocrea
Alex Dekhtyar
Esra Erdem
Wolfgang Faber
Flavio A. Ferrarotti
Michael Fink
Alfredo Garro
Giuseppe De Giacomo
Martin Grohe
Michael Guppenberger
Giovambattista Ianni
Mirjana Jaksic
Eiji Kawaguchi
Kevin Keenoy
Henning Koehler
Markus Lehmann
Hans-Joachim Lenz
Sebastian Link
Irina Lomazova
Carlo Meghini
Dan Olteanu
Martin Otto
Viet Phan-Luong
Andrea Pugliese
Francesco Ricca
Flavio Rizzolo
Francesco Scarcello
Michele Sebag
Stefanie Scherzinger
Roman Schindlauer
Peggy Schmidt
Cristina Sirangelo
Viorica Sofronie-Stokkermans
Umberto Straccia
Andrea G.B. Tettamanzi
Cesare Tinelli
Hans Tompits
Yannis Tzitzikas
Domenico Ursino
Jing Wang
Stefan Woltran
Dell Zhang

Local Arrangements

Dezső Miklós   Alfréd Rényi Institute of Mathematics (Hungary)

Publicity and Website Management

Markus Kirchberg   Massey University (New Zealand)

Table of Contents

The Semijoin Algebra

Jan Van den Bussche

Universiteit Hasselt, Agoralaan D,
B-3590 Diepenbeek, Belgium
jan.vandenbussche@uhasselt.be

Abstract. When we replace, in the classical relational algebra, the join operator by the semijoin operator, we obtain what we call the semijoin algebra. We will show that, when only equi-joins are used, the semijoin algebra is equivalent with the guarded fragment of first-order logic, and thus it inherits many of the nice properties of the latter logic. When more general theta-joins are used, however, we move outside the realm of guarded logics, and we will show how the notion of guarded bisimilarity can be extended accordingly. Last but not least, we show how the semijoin algebra can be used as a tool to investigate the complexity of queries expressed in the relational algebra, where we are mainly interested in whether or not a relational algebra expression for the query needs to produce intermediate results of nonlinear size. For example, we will show that the division operation cannot be expressed by a linear relational algebra expression.

This talk is a survey of work done in collaboration with Dirk Leinders, Jerzy Tyszkiewicz, and Maarten Marx.

References

1. D. Leinders, M. Marx, J. Tyszkiewicz and J. Van den Bussche. The semijoin algebra and the guarded fragment. *Journal of Logic, Language and Information*, 14:331–343, 2005.
2. D. Leinders and J. Van den Bussche. On the complexity of division and set joins in the relational algebra. *Proceedings 24th ACM Symposium on Principles of Database Systems*, pages 76–83. ACM Press, 2005.
3. D. Leinders, J. Tyszkiewicz and J. Van den Bussche. On the expressive power of semijoin queries. *Information Processing Letters*, 91:93–98, 2004.

J. Dix and S.J. Hegner (Eds.): FoIKS 2006, LNCS 3861, p. 1, 2006.

Equational Constraint Solving Via a Restricted Form of Universal Quantification

Javier Álvez* and Paqui Lucio

Dpto. de Lenguajes y Sistemas Informáticos, UPV-EHU, Paseo Manuel de Lardizabal, 1, 20080-San Sebastián, Spain
{jibalgij, jiplucap}@si.ehu.es

Abstract. In this paper, we present a syntactic method for solving first-order equational constraints over term algebras. The presented method exploits a novel notion of *quasi-solved* form that we call *answer*. By allowing a restricted form of universal quantification, *answers* provide a more compact way to represent solutions than the purely existential solved forms found in the literature. *Answers* have been carefully designed to make satisfiability test feasible and also to allow for boolean operations, while maintaining expressiveness and user-friendliness. We present detailed algorithms for (1) satisfiability checking and for performing the boolean operations of (2) negation of one *answer* and (3) conjunction of n *answers*. Based on these three basic operations, our solver turns any equational constraint into a disjunction of *answers*. We have implemented a prototype that is available on the web.

Keywords: equality, constraint satisfaction, solver, term algebra, *answer*.

1 Introduction

An *equational constraint* is an arbitrary first-order formula built over a signature Σ of function symbols and equality as unique predicate symbol. Equational constraints are interpreted over term algebras. An *equational solving method* takes as input an equational constraint and produces the set of all its solutions or, more precisely, some particular representation of it. Syntactic methods are *rewriting processes* that transform the input constraint into an equivalent disjunction of constraints, in the so-called *solved form*, which represents its solutions. In particular, those solutions serve to decide whether the input constraint is satisfiable or not.

On one hand, equational constraint solving is an very important tool in many areas of automated deduction. The integration of efficient equational solvers in theorem provers has been a challenging problem, important for many practical applications. Equational constraints can be used for restricting the set of ground

* Corresponding author.

J. Dix and S.J. Hegner (Eds.): FoIKS 2006, LNCS 3861, pp. 2–21, 2006.

instances in order to define more eficient resolution mechanism (cf. [6]). In automated model building, equational constraints play a crucial role for model representation (e.g. [5, 11]). On the other hand, equational constraint solving may be applied for several purposes in the areas of functional/logic programming and databases. Therein, many problems related to semantics and implementation issues can be reduce to equational constraint solving problems. Some well known examples include the problems of answering negative goals, decidindg whether a case-definition is complete, evaluating a boolean conjunctive query on a relational database, etc. Besides, equational constraint solving have been found useful in other areas such as formal verification tools, computational linguistic, machine learning, program transformation, etc.

It is well known that the *free equality theory*[1], originally introduced by Malcev in [15], is non-elementary (see [10, 21]). Besides, the inherent complexity of the satisfiability problem of equational problems (i.e. where the quantifier prefix is of the form $\forall^*\exists^*$) for finite signature is studied in [19]. The most well known algorithms for equational solving [9, 14, 15] and later extensions to richer theories (see [20]) are based on quantifier elimination with solved forms that combine equations and disequations. Negation should be allowed since, for example, the constraint $\forall v(\ x \neq f(v,v)\ )$ cannot be finitely represented without negation (disequations). As opposed to negation, universal quantification can be dropped from any equational formula by the well-known *quantifier elimination technique*. As a consequence, most solved form notions (see [8] for a survey) are boolean combinations of certain kind of existential formulas whose satisfiability test is trivial even in the case of finite signature. However, in exchange for the simplicity of the test, the solver must remove all universal quantifiers. This often requires application of the so-called[2] *Explosion Rule* ([9]) that we recall in Fig. 1, that implies substitution of a formula by a disjunction of as many formulas as there are function symbols in the finite signature Σ. A more compact

$$\text{(Exp)}\quad \forall\overline{y}(\ \varphi\ ) \longmapsto \bigvee_{f\in\Sigma} \exists\overline{z}\forall\overline{y}(\ \varphi \wedge w = f(\overline{z})\ )$$

if there exists an equation $x = t$ or a disequation $x \neq t$ such that some $y_i \in \overline{y}$ occurs in t, $\overline{z}$ are fresh and Σ is finite

Fig. 1. The Explosion Rule (Exp)

representation of solutions reduces the blow up of the number of disjuncts along the quantifier elimination process, which in turn improves the method. At the same time, the basic operations for managing this more expressive notion must not be expensive. We propose a notion of *quasi-solved form*, called *answer*, that allows a restricted form of universal quantification, which is enough to avoid the

[1] Also called the theory of term algebra and Clark's equational theory.
[2] That is, the *Weak Domain Closure Axiom* in the nomenclature of [14].

above rule (Exp) and offers a more compact representation of solutions. *Answers* have been carefully designed to make satisfiability test feasible and also to allow for boolean operations (of negation and conjunction), while retaining expressiveness and user-friendliness. The idea of gaining efficiency via restricted forms of universal quantification has been already proposed in [18] and in [16].

A very preliminary version of this work was presented as [1]. We have implemented (in Prolog) a prototype of the general constraint solver. It is available at ~http://www.sc.ehu.es/jiwlucap/equality_constraints.html.

Outline of the paper. In the next section, we recall some useful definitions and denotational conventions. Section 3 is devoted to the details of the notion of *answer* and some examples. In Section 4, we introduce the *answer* satisfiability test with some illustrative examples. In Section 5, we show how the two other basic operations on *answers* —conjunction and negation— can be efficiently performed. Besides, we make use of these basic operations (together with the quantifier elimination technique) to provide a solving method for general equational constraints. We give a summarizing example in Section 6. Finally, we present some concluding remarks and briefly discuss some related work.

2 Definitions and Notation

Let us fix a denumerable set of variables X. Given a (finite or infinite) signature Σ, a Σ-term is a variable from X, or a constant, or a function symbol of arity n applied to n terms. A term is *ground* if it contains no variable symbols. $\mathcal{T}(\Sigma)$ stands for the algebra of all ground Σ-terms or Herbrand universe, whereas $\mathcal{T}(\Sigma, X)$ is used to denote the set of all Σ-terms. We denote by $Var(t)$ the set of all variables occurring in t and $t(\overline{v})$ denotes that $Var(t) \subseteq \overline{v}$. A term is *linear* if it contains no variable repetitions. A bar is used to denote tuples of objects. Subscripts are used to denote the components of a tuple and superscripts are used to enumerate tuples. For example, x_j denotes a component of $\overline{x}$, whereas $\overline{x}^1, \ldots, \overline{x}^j, \ldots, \overline{x}^m$ is a tuple enumeration and x_i^j is a component of the tuple $\overline{x}^j$. Concatenation of tuples is denoted by the infix operator $\cdot$, i.e. $\overline{x} \cdot \overline{y}$ represents the concatenation of $\overline{x}$ and $\overline{y}$. When convenient, we treat a tuple as the set of its components. A Σ-equation is $t_1 = t_2$, where t_1 and t_2 are Σ-terms, whereas $t_1 \neq t_2$ is a Σ-disequation (that is also written as $\neg(t_1 = t_2)$). By a *collapsing* equation (or disequation) we mean that at least one of its terms is a variable. We abbreviate collapsing Σ-equation by Σ-CoEq, and Σ-UCD stands for universally quantified collapsing Σ-disequation. We abbreviate $\bigwedge_i t_i = s_i$ by $\overline{t} = \overline{s}$ and $\bigvee_i t_i \neq s_i$ by $\overline{t} \neq \overline{s}$. To avoid confusion, we use the symbol $\equiv$ for the metalanguage equality.

A *Σ-substitution* $\sigma \equiv \{x_1 \leftarrow t_1, \ldots x_n \leftarrow t_n\}$ is a mapping from a finite set of variables $\overline{x}$, called *domain*(σ), into $\mathcal{T}(\Sigma, X)$. It is assumed that σ behaves as the identity for the variables outside *domain*(σ). A substitution σ is called a *Σ-assignment* if $\sigma(x_i) \in \mathcal{T}(\Sigma)$ for all $x_i \in$ *domain*(σ). We intentionally confuse the above substitution σ with the conjunction of equations $\bigwedge_i x_i = t_i$. The (possibly ground) term $\sigma(t)$ (also denoted $t\sigma$) is called an *(ground) instance* of the term t. The *most general unifier* of a set of terms $\{s_1,\ldots,s_m\}$, denoted $mgu(\overline{s})$, is an idempotent substitution σ such that $\sigma(s_i)\equiv\sigma(s_j)$ for all $1\le i,j\le m$ and for any other substitution θ with the same property, $\theta\equiv\sigma'\cdot\sigma$ holds for some substitution σ'. For tuples, $mgu(\overline{s}^1,\ldots,\overline{s}^m)$ is an abbreviation of $\sigma_1\cdot\ldots\cdot\sigma_n$ where $\sigma_i\equiv mgu(s_i^1,\ldots,s_i^m)$ for all $1\le i\le n$. The *most general common instance* of two terms t and s, denoted by $mgi(t,s)$, is the term whose set of ground instances is the intersection of both sets of ground instances (for t and s), and it can be computed using a unification algorithm.

An *equational constraint* is a first-order Σ-formula built over Σ-equations (as atoms) using the classical connectives and quantifiers. Atoms include the logical constants True and False. Equational constraints are interpreted in the term algebra $\mathcal{T}(\Sigma)$. A Σ-assignment σ satisfies a Σ-equation $t_1=t_2$ iff $t_1\sigma\equiv t_2\sigma$. The logical constants, connectives and quantifiers are interpreted as usual. A *solution* of an equational constraint is a Σ-assignment that satisfies the constraint. Constraint equivalence means the coincidence of the set of solutions. We make no distinction between a set of constraints $\{\varphi_1,\ldots,\varphi_k\}$ and the conjunction $\varphi_1\wedge\ldots\wedge\varphi_k$. We abbreviate $\forall x_1\ldots\forall x_n$ (resp. $\exists x_1\ldots\exists x_n$) by $\forall\overline{x}$ (resp. $\exists\overline{x}$).

3 The Notion of *Answer*

In this section, we present the notion of *answer* and give some illustrative examples. The following definition also introduces some notational conventions.

Definition 1. *Let $\overline{x}$ be a k-tuple of pairwise distinct variables. A Σ-*answer for $\overline{x}$ *is either a logical constant (*True *or* False*) or a formula of the form $\exists\overline{w}(\ a(\overline{x},\overline{w})\ )$, where $a(\overline{x},\overline{w})$ is a conjunction of Σ-*CoEq*s and Σ-*UCD*s of the form:*

$$x_1=t_1\wedge\ldots\wedge x_k=t_k\ \wedge\ \bigwedge_{i=1}^{n}\bigwedge_{j=1}^{m_i}\forall\overline{v}(\ w_i\neq s_{ij}(\overline{w},\overline{v})\ )\tag{1}$$

such that the n-tuple $\overline{w}\equiv Var(t_1,\ldots,t_k)$ is disjoint from $\overline{x}$, every term $s_{ij}(\overline{w},\overline{v})$ neither contains the variable w_i nor is a single universal variable in $\overline{v}$, and $n,m_1,\ldots,m_n\ge 0$. □

Remark 1. We abbreviate the equational part of (1) by $\overline{x}=\overline{t}(\overline{w})$. Any equation $x_j=w_i$ such that w_i does not occur in the rest of the *answer* can be left out. The scope of each universal quantifier is restricted to one single disequation, although a universal variable can occur repeatedly in a disequation. □

The following examples show that *answers* provide a compact and explanatory description of the sets of solutions that they represent.

Example 1. Let $\Sigma\equiv\{a/0,g/1,f/2\}$. Consider the following *Σ-answer*:

$$\exists w_1\exists w_2(\ x=f(w_1,w_2)\wedge\forall v_1(\ w_1\neq(v_1)\ )\wedge\forall v_2(\ w_2\neq f(w_1,v_2)\ )\ ).\tag{2}$$

By application of the rule (Exp) for eliminating v_1 and v_2, the *answer* (2) is equivalent to the disjunction of the following six existential constraints: