



The South African Institute for Computer Scientists and  
Information Technologists

**ANNUAL RESEARCH AND DEVELOPMENT  
SYMPOSIUM**

23-24 NOVEMBER 1998

CAPE TOWN

Van Riebeeck hotel in Gordons Bay

Hosted by the University of Cape Town in association with the CSSA,  
Potchefstroom University for CHE and  
The University of Natal

**PROCEEDINGS**

**EDITED BY**  
D. PETKOV AND L. VENTER

**SPONSORED BY**







**The South African Institute for Computer Scientists and  
Information Technologists**

**ANNUAL RESEARCH AND DEVELOPMENT  
SYMPOSIUM**

**23-24 NOVEMBER 1998  
CAPE TOWN  
Van Riebeeck hotel in Gordons Bay**

**Hosted by the University of Cape Town in association with the CSSA,  
Potchefstroom University for CHE and  
The University of Natal**

**GENERAL CHAIR : PROF G. HATTINGH, PU CHE**

**PROGRAMME CO-CHAIRS:  
PROF. L VENTER, PU CHE (Vaal Triangle), PROF. D. PETKOV, UN-PMB**

**LOCAL ORGANISING CHAIR: PROF. P. LICKER, UCT - IS**

**PROCEEDINGS**

**EDITED BY  
D. PETKOV AND L. VENTER**

**SYMPOSIUM THEME:**

**Development of a quality academic CS/IS infrastructure in South Africa**

**SPONSORED BY**



Copyrights reside with the original authors who may be contacted directly.

Proceedings of the 1998 Annual Research Conference of the South African Institute for Computer Scientists and Information Technologists.

Edited by Prof. D. Petkov and Prof. L. Venter

Van Reebeck Hotel, Gordons Bay, 23-24 November 1998

**ISBN: 1-86840-303-3**

**Keywords: Computer Science, Information Systems, Software Engineering.**

The views expressed in this book are those of the individual authors and not of the South African Institute for Computer Scientists and Information Technologists.

Office of SAICSIT: Prof. J.M.Hatting, Department of Computer Science and information Systems, Potchefstroom University for CHE, Private Bag X6001, Potchefstroom, 2520, RSA.

**Produced by the Library Copy Centre, University of Natal, Pietermaritzburg.**

## FOREWORD

The South African Institute for Computer Scientists and Information Technologists (SAICSIT) promotes the cooperation of academics and industry in the area of research and development in Computer Science, Information Systems and Technology and Software Engineering. The culmination of its activities throughout the year is the annual research symposium. This book is a collection of papers presented at the 1998 such event taking place on the 23<sup>rd</sup> and 24<sup>th</sup> of November in Gordons Bay, Cape Town. The Conference is hosted by the Department of Information Systems, University of Cape Town in cooperation with the Department of Computer Science, Potchefstroom University for CHE and and Department of Computer Science and Information Systems of the University of Natal, Pietermaritzburg.

There are a total of 46 papers. The speakers represent practitioners and academics from all the major Universities and Technikons in the country. The number of industry based authors has increased compared to previous years.

We would like to express our gratitude to the referees and the paper contributors for their hard work on the papers included in this volume. The Organising and Programme Committees would like to thank the keynote speaker, Prof M.C.Jackson, Dean, University of Lincolnshire and Humberside, United Kingdom, President of the International Federation for Systems Research as well as the Computer Society of South Africa and The University of Cape Town for the cooperation as well as the management and staff of the Potchefstroom University for CHE and the University of Natal for their support and for making this event a success.

Giel Hattingh, Paul Licker, Lucas Venter and Don Petkov



| <b>Table of Contents</b>  | <b>Page</b> |
|---|-------------|
| <b>Lynette Drevin:</b> Activities of IFIP wg 11.8 (computer security education) & IT related ethics education in Southern Africa  | 1           |
| <b>Reinhardt A. Botha and Jan H.P. Eloff:</b> exA Security Interpretation of the Workflow Reference Model                         | 3           |
| <b>Willem Krige and Rossouw von Solms:</b> Effective information security monitoring using data logs                              | 9           |
| <b>Eileen Munyiri and Rossouw von Solms:</b> Introducing Information Security: A Comprehensive Approach                           | 12          |
| <b>Carl Papenfus and Reinhardt A. Botha:</b> A shell-based approach to information security                                       | 15          |
| <b>Walter Smuts:</b> A 6-Dimensional Security Classification for Information  | 20          |
| <b>Philip Machanick and Pierre Salverda:</b> Implications of emerging DRAM technologies for the RAM page Memory hierarchy         | 27          |
| <b>Susan Brown:</b> Practical Experience in Running a Virtual Class to Facilitate On-Campus Under Graduate Teaching               | 41          |
| <b>H.D. Masethe, T.A Dandadzi:</b> Quality Academic Development of CS/IS Infrastructure in South Africa                           | 49          |
| <b>Philip Machanick:</b> The Skills Hierarchy and Curriculum  | 54          |
| <b>Theda Thomas:</b> Handling diversity in Information Systems and Computer Science Students: A social Constructivist Perspective | 63          |
| <b>Udo Averweg and G J Erwin:</b> Critical success factors for implementation of Decision support systems                         | 70          |
| <b>Magda Huisman:</b> A conceptual model for the adoption and use of case technology  | 78          |
| <b>Paul S. Licker:</b> A Framework for Information Systems and National Development Research                                      | 79          |
| <b>K. Niki Kunene and Don Petkov:</b> On problem structuring in an Electronic Brainstorming (EBS) environment                     | 89          |

|   |     |
|---|-----|
| <b>Derek Smith:</b> Characteristics of high-performing Information Systems Project Managers and Project Teams   | 90  |
| <b>Lucas Venter:</b> INSTAP: Experiences in building a multimedia application   | 102 |
| <b>Scott Hazelhurst, Anton Fatti, and Andrew Henwood:</b> Binary Decision Diagram Representations of Firewall and Router Access Lists   | 103 |
| <b>Andre Joubert and Annelie Jordaan:</b> Hardware System interfacing with Delphi 3 to achieve quality academic integration between the fields of Computer Systems and Software Engineering | 113 |
| <b>Borislav Roussev:</b> Experience with Java in an Advanced Operating Systems Module   | 121 |
| <b>Conrad Mueller:</b> A Static Programming Paradigm  | 122 |
| <b>Sipho Langa:</b> Management Aspects of Client/Server Computing   | 130 |
| <b>T Nepal and T Andrew:</b> An Integrated Research Programme in AI applied to Telecommunications at ML Sultan Technikon  | 135 |
| <b>Yuri Velinov:</b> Electronic lectures for the mathematical subjects in Computer Science  | 136 |
| <b>Philip Machanick:</b> Disk delay lines   | 142 |
| <b>D Petkov and O Petkova:</b> One way to make better decisions related to IT Outsourcing   | 145 |
| <b>Jay van Zyl:</b> Quality Learning, Learning Quality  | 153 |
| <b>Matthew O Adigun:</b> A Case for Reuse Technology as a CS/IS Training Infrastructure   | 162 |
| <b>Andy Bytheway and Grant Hearn:</b> Academic CS/IS Infrastructure in South Africa: An exploratory stakeholder perspective   | 171 |
| <b>Chantel van Niekerk:</b> The Academic Institution and Software Vendor Partnership  | 172 |
| <b>Christopher Chalmers:</b> Quality aspects of the development of a rule-based architecture  | 173 |
| <b>Rudi Harmse:</b> Managing large programming classes using computer mediated communication and cognitive modelling techniques   | 174 |



|  |     |
|--|-----|
| <b>Michael Muller:</b> How to gain Quality when developing a Repository Driven User Interface  | 184 |
| <b>Elsabe Cloete and Lucas Venter:</b> Reducing Fractal Encoding Complexities  | 193 |
| <b>Jean Bilbrough and Ian Sanders:</b> Partial Edge Visibility in Linear Time  | 200 |
| <b>Philip Machanick:</b> Design of a scalable Video on Demand architecture   | 211 |
| <b>Freddie Janssen:</b> Quality considerations of Real Time access to Multidimensional Matrices  | 218 |
| <b>Machiel Kruger and Giel Hattingh:</b> A Partitioning Scheme for Solving the Exact $k$ -item 0-1 Knapsack Problem                            | 229 |
| <b>Ian Sanders:</b> Non-orthogonal Ray Guarding  | 230 |
| <b>Fanie Terblanche and Giel Hattingh:</b> Response surface analysis as a technique for the visualization of linear models and data            | 236 |
| <b>Olga Petkova and Dewald Roode:</b> A pluralist systemic framework for the evaluation of factors affecting software development productivity | 243 |
| <b>Peter Warren and Marcel Viljoen:</b> Design patterns for user interfaces  | 252 |
| <b>Andre de Waal and Giel Hattingh:</b> Refuting conjectures in first order theories   | 261 |
| <b>Edna Randiki:</b> Error analysis in Selected Medical Devices and Information Systems  | 262 |



# REFUTING CONJECTURES IN FIRST-ORDER THEORIES

DA de Waal and GN Hattingh  
Potchefstroom University for Christian Higher Education,  
Potchefstroom, South Africa  
BWIDADW@puknet.puk.ac.za, RKWGNH@sb.puk.ac.za}

## Abstract

Given some first-order theory, a formula (also called a conjecture) may or may not be a theorem of some theory. A theorem prover attempting to prove that the given formula is a theorem may terminate, proving that the formula is (or isn't) a theorem or run forever (in practice the theorem prover will usually halt when some resource is exhausted). In the latter case we are left with an unanswered question about the status of the formula. There are basically two ways to proceed. First, accept the limitations of the theorem prover and the inconclusiveness of the result or second, attempt to use some other technique to make a judgement about the status of the given formula.

So far there seems to be very little research exploring the detection of non-theorems in first-order logic. Most theorem provers concentrate on the detection of theorems and the automatic detection of non-theorems is usually neglected. For resolution based theorem provers a reason for this might be that refutation completeness gives a strong incentive to concentrate on proving theorems as we are certain that the theorem prover will answer affirmatively, in finite time, if the formula is indeed a theorem. In general, such a result does not hold if the formula is not a theorem.

One general system that can be used to refute conjectures is based on meta-programming, partial evaluation and regular approximations. The idea is to approximate the behaviour of a proof procedure on a specific theorem proving problem. If the generated approximation is empty we have a proof that the formula we are interested in is not a theorem of the particular theory. As failure is decidable in the approximation, this can be done in finite time.

Another general system that is capable of refuting conjectures is FINDER (Finite Domain Enumerator). It takes as input a first-order theory, expressed as a set of clauses, and gives as output the models of that theory with domains of given finite cardinality.

We developed a system dedicated to detecting failure in definite logic programs (refuting conjectures in Horn theories). Our system uses a top-down evaluation strategy with tabulation and constraint logic programming technology as its basic mechanisms. As FINDER has no equivalent of a top-down strategy, it behaves poorly on many problems with huge search spaces.

FINDER is at least an order of magnitude faster in raw speed than our prototype system, but our results suggests that our basic strategy (using a goal directed strategy with tabulation) is sound and scales up better than the bottom-up strategy used in FINDER. However it lacks in power compared to FINDER as it is only designed to handle Horn clauses, where FINDER takes as input first-order clauses.

The obvious question now is if we can extend our top-down strategy to cover first-order theories and formulas. Unfortunately such an extension is not straightforward as we rely heavily on results from logic programming for success. However, there is a way in which the given results for our system can be extended to the first-order case. The idea is to use meta-programming to "reduce" the first-order problem to a Horn clause problem and furthermore to use program specialisation to reduce the overhead introduced by the meta-program.

In the paper we explain the proposed approach in more detail and give some preliminary results and conclusions.