Nieuwsbrief van de Nederlandse Vereniging voor Theoretische Informatica

Jan Willem Klop, Jan Rutten, Susanne van Dam (redactie) *

Inhoudsopgave

*CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands. Email: susanne@cwi.nl.

1 Van de Redactie

Beste NVTI-leden,

Graag bieden wij u hierbij het zevende nummer aan van de jaarlijkse NVTI-Nieuwsbrief. Bij het samenstellen hebben we weer de formule van de vorige zes nummers gevolgd. Zo vindt u naast het programma van de jaarlijkse Theoriedag en de bijgewerkte ledenlijst ook weer enkele bijdragen van collega's met een korte inleiding in hun speciale gebied van expertise. Evenals voorgaande jaren zouden deze Nieuwsbrief en de Theoriedag niet tot stand hebben kunnen komen zonder de financiele steun van onze sponsors: NWO-EW, Elsevier Publishing Company, en de onderzoekscholen IPA en SIKS. Namens de NVTI gemeenschap onze hartelijke dank voor deze middelen die ons voortbestaan mogelijk maken! Een speciaal woord van dank willen we namens de gehele NVTI gemeenschap richten aan Mieke Bruné. Mieke heeft de afgelopen elf jaar de Nieuwsbrief (van respectievelijk WTI, VTI, NVTI) op voortreffelijke wijze verzorgd. Mieke heeft sinds het begin van dit jaar haar werk overgedragen aan Susanne van Dam, die voortaan het aanspreekpunt zal zijn voor redactionele Nieuwsbrief zaken. Mieke, nogmaals dank, en Susanne, welkom!

De redactie,
Jan Willem Klop (jwk@cwi.nl)
Jan Rutten (janr@cwi.nl)
Susanne van Dam (susanne @cwi.nl)

2 Samenstelling Bestuur

Prof.dr. J.C.M. Baeten (TUE)
Dr. H.L. Bodlaender (UU)
Prof.dr. J.W. Klop (VUA/CWI/KUN) voorzitter
Prof.dr. J.N. Kok (RUL)
Prof.dr. J.-J.Ch. Meyer (UU)
Prof.dr. G.R. Renardel de Lavalette (RUG)
Prof.dr. G. Rozenberg (RUL)
Prof.dr. J.J.M.M. Rutten (CWI/VUA) secretaris
Dr. L. Torenvliet (UvA)

3 Van de voorzitter

Geacht NVTI-lid,

Ook dit jaar heeft het Bestuur zich beijverd om een interessante Theoriedag te organiseren, met vier prominente sprekers uit binnen- en buitenland. De Theoriedag zal gehouden worden op vrijdag 7 maart, in Hoog-Brabant, Utrecht. We hopen en vertrouwen erop dat het programma voor velen van u weer interessant is. Graag tot ziens op 7 maart in Utrecht!

Jan Willem Klop, voorzitter NVTI

4 Theoriedag 2003

Vrijdag 7 maart 2003, Hoog Brabant, Utrecht
Theoriedag 2003 van de NVTI (Nederlandse Vereniging voor Theoretische Informatica)
Vrijdag 7 maart 2003 Vergadercentrum Hoog Brabant Radboudkwartier 23 Hoog Catharijne Utrecht
Het is ons een genoegen u uit te nodigen tot het bijwonen van de Theoriedag 2003 van de NVTI, de Nederlandse Vereniging voor Theoretische Informatica, die zich ten doel stelt de theoretische informatica te bevorderen en haar beoefening en toepassingen aan te moedigen. De Theoriedag 2003 zal gehouden worden op vrijdag 7 maart aanstaande, in Vergadercentrum Hoog Brabant te Utrecht, gelegen in winkelcenrum Hoog Catharijne, op enkele minuten loopafstand van CS Utrecht, en is een voortzetting van de reeks jaarlijkse bijeenkomsten van de NVTI die acht jaar geleden met de oprichtingsbijeenkomst begon. Evenals vorige jaren hebben wij een aantal prominente sprekers uit binnen- en buitenland bereid gevonden deze dag gestalte te geven met voordrachten over recente en belangrijke stromingen in de theoretische informatica. Naast een wetenschappelijke inhoud heeft de dag ook een informatief gedeelte, in de vorm van een algemene vergadering waarin de meest relevante informatie over de NVTI gegeven zal worden, alsmede presentaties van de onderzoekscholen.
Programma (samenvattingen volgen beneden)

09.30-10.00: Ontvangst met koffie

10.00-10.10: Opening

10.10-11.00: Lezing Prof.dr. L. Fortnow (Nec Laboratories America)
Titel: Church, Kolmogorov and von Neumann: Their Legacy Lives in Complexity

11.00-11.30: Koffie

11.30-12.20: Lezing Prof.dr. P. Stevenhagen (UL)
Titel: Primes is in P

12.20-12.50: Presentatie Onderzoeksscholen (OZL, IPA, SIKS)

12.50-14.10: Lunch (Zie beneden voor registratie)

14.10-15.00: Lezing Prof.dr. M. Vardi (Rice University, USA)
Titel: The Design of A Formal Property-Specification Language

15.00-15.20: Thee

15.20-16.10: Lezing Dr. M. de Rijke (UvA)
Titel: Intelligent Information Access

16.10-16.40: Algemene ledenvergadering NVTI

Lunchdeelname

Het is mogelijk aan een georganiseerde lunch deel te nemen; hiervoor is aanmelding verplicht. Dit kan per email of telefonisch bij Susanne van Dam (susanne@cwi.nl, 020-592 4189), tot een week voor de bijeenkomst (28 februari). De kosten kunnen ter plaatse voldaan worden; deze bedragen Euro 14. Wij wijzen erop dat in de onmiddellijke nabijheid van de vergaderzaal ook uitstekende lunchfaciliteiten gevonden kunnen worden, voor wie niet aan de georganiseerde lunch wenst deel te nemen.

Lidmaatschap NVTI

Aan het lidmaatschap zijn geen kosten verbonden; u krijgt de aankondigingen van de NVTI per email of anderszins toegestuurd. Wilt u lid van de NVTI worden, dan kunt u zich aanmelden bij Susanne van Dam (susanne@cwi.nl) met vermelding van de relevante gegevens (naam, voorletters, affiliatie indien van toepassing, correspondentieadres, email, URL, telefoonnummer).

Steun

De activiteiten van de NVTI worden mede mogelijk gemaakt door de ondersteuning (financieel en anderszins) van de volgende instellingen: NWO/EW, CWI, Onderzoeksscholen IPA, SIKS, OZSL, Elsevier Science B.V.

Samenvattingen van de lezingen

Prof.dr. L. Fortnow (Nec Laboratories, USA)

Church, Kolmogorov and von Neumann: Their Legacy Lives in Complexity

In the year 1903, several of the greatest early computer scientists entered our world. In this talk we look at the work of three of these giants: Alonzo Church, Andrey Kolmogorov and John von Neumann honoring the 100th anniversary of their births. We will focus on how their research has and continues to play a major role in the development of computational complexity and our understanding of what we can compute.

Alonzo Church developed the lambda-calculus, a computation model equivalent to the Turing machine. He co-developed independently with Alan Turing what we now call the Church-Turing thesis that states that every computable is computable by a Turing machine (or the lambda-calculus).

John von Neumann's work in quantum mechanics, game theory, automata theory and his development of early computers have played major roles in the development of algorithms and complexity. We will spend most of the seminar discussing the influence of Andrey Kolmogorov, whose work on algorithmic randomness has had a more direct impact on computational complexity and certainly my own research. We will give an overview of Kolmogorov complexity and several examples of computational restricted versions of this measure have helped us better understand the nature of efficient computation.

Prof.dr. P. Stevenhagen (UL)

Primes is in P

In August 2002, the Indian computer scientists Agrawal, Kayal and Saxena proved that primality of an integer can be tested by means of a deterministic algorithm that runs in polynomial time. For several decades, this had been an outstanding problem. We discuss the importance of the result in theory and practice, and give an impression of the mathematics that goes into it.

Prof.dr. M. Vardi (Rice University, USA)

The Design of A Formal Property-Specification Language

In recent years, the need for formal specification languages is growing rapidly as the functional validation environment in semiconductor design is changing to include more and more validation engines based on formal verification technologies. In particular, the usage of Formal Equivalence Verification and Formal Property Verification is growing, new symbolic simulation engines are

introduced and hybrid environments of scalar and symbolic simulators are developed. To facilitate the use of these new-generation validation engines - properties, checkers and reference models need to be developed in a formal language.
In this talk we describe the design of the ForSpec Temporal Logic (FTL), the new temporal logic of ForSpec. Intel's new formal property-specification language, which is today part of Synopsis OpenVera hardware verification language (www.open-vera.com). The key features of FTL are: it is a linear temporal logic, based on Pnueli's LTL, it enables the user to define temporal connectives over time windows, it enables the user to define regular events, which are regular sequences of Boolean events, and then relate such events via special connectives, and it contains constructs that enable the user to model multiple clock and reset signals, which is useful in the verification of globally asynchronous and locally synchronous hardware designs.
The focus of the talk is on design rationale, rather than a detailed language description.

Dr. M. de Rijke (UvA)

Intelligent Information Access

Search is one of the core topics in theoretical computer science, and online search has become a day-to-day activity for many of us. Finding keywords in a text file is easy. Using keyword search, current retrieval systems allow users to find documents that are relevant to their information needs, but most leave it to the user to extract the useful information from those documents. This leaves the (often unwilling) user with a relatively large amount of text to consume. People have questions and they need answers, not documents. There is a need for tools that reduce the amount of text one might have to read to obtain the desired information.
In this talk I review ongoing research initiatives (such as novelty detection, question answering, and XML retrieval) aimed at moving beyond traditional document retrieval, and I will try to identify theoretical issues that arise from these initiatives.

5 Mededelingen van de onderzoekscholen

Hieronder volgen korte beschrijvingen van de onderzoekscholen:

- Instituut voor Programmatuurkunde en Algoritmiek
- The School for Information and Knowledge systems (SIKS)

5.1 Institute for Programming research and Algorithmics

The research school IPA (Institute for Programming Research and Algorithmics) educates researchers in the field of programming research and algorithmics. This field encompasses the study and development of formalisms, methods and techniques to design, analyse, and construct software systems and components. IPA has three main research areas: Algorithmics & Complexity, Formal Methods and Software Technology. Researchers from eight universities (University of Nijmegen, Leiden University, Eindhoven University of Technology, University of Twente, Utrecht University, University of Groningen, Vrije Universiteit Amsterdam, and the University of Amsterdam), the CWI and Philips Research (Eindhoven) participate in IPA.
In 1997, IPA was formally accredited by the Royal Dutch Academy of Sciences (KNAW) for a period of five years. During 2001, an application for the extension of the accreditation was prepared. This extension was granted in the summer of 2002 (for a period of five years) and in October IPA threw a big party at the Efteling to celebrate its renewed accreditation. 2002 was also a good year for IPA in other respects: it brought 16 IPA Ph.D. defenses, a record number in the history of the school.

Activities in 2002

IPA has two multi-day events per year, the Lentedagen and the Herfstdagen, which focus on a particular subject. In the 2002 - 2006 period, each of the Herfstdagen will be dedicated to one of IPA's four so-called application areas: Networked Embedded Systems, Security, Intelligent Algorithms, and Compositional Programming Methods. In 2002, the Lentedagen were on Middleware, the Herfstdagen on Networked Embedded Systems.

Lentedagen Middleware is the term used to refer to a software layer between the operating system and distributed applications that interact via a network. Its task is to facilitate the interaction among the applications. Due to the spectacular advances in hardware and networking, Middleware now has to shield applications from the heterogeneity of computer architectures, operating systems, programming languages and networking technologies, and its tasks have been broadened to include dealing with things like data distribution, parallelism, quality of service and information security. The increasing scope and complexity of Middleware leads to issues which have traditionally been addressed in IPA research for other reasons.

By its very nature, the topic Middleware is not just of interest to IPA, but also to other research schools in computer science. In composing the program for the Lentedagen Michel Chaudron (TU/e, IPA) cooperated successfully with Maarten van Steen (VU, ASCI) and Marten van Sinderen (UT, CTIT). Their cooperation resulted in an overview of Middleware research in and around IPA, including contributions on industrial experience with the development of Middleware. It contained sessions on: Mobile and Wireless Middleware, Peer-to-Peer Systems, Quality-aware Middleware, Aspect-oriented Design, In-home Networks, and Differentiated Data Distribution. Anthony Rowstron of Microsoft Research Cambridge was special guest speaker. Abstracts, papers and handouts can be found at the web site.
see: http://www.win.tue.nl/ipa/activities/springdays2002/index.html

Herfstdagen Embedded systems communicate more and more with each other and with the environment. Hence new issues are becoming important in Embedded Systems research, traditionally a mainstay of IPA's research program. IPA has chosen to address this development by making Networked Embedded Systems one of the four application areas for IPA-research in the period 2002-2006.

In networking, algorithms and protocols for self-organising networks and ad-hoc networking come to the fore, in mobile and wireless computing, localisation of devices becomes an issue, and the increasing use of sensors and actuators makes the study of hybrid systems more urgent. The program of the IPA Herfstdagen contained sessions on these topics, as well as on more familiar themes like power consumption, testing and verification. As with Middleware, the topic of Networked Embedded Systems is not exclusively of interest to IPA, and in preparing the program Wan Fokkink (CWI), Jozef Hooman (KUN), and Joost-Pieter Katoen (UT) of IPA cooperated with Koen Langendoen (TUD, ASCI) and Martin Rem (ESI). For abstracts, hand-outs and papers see: http://www.win.tue.nl/ipa/archive/falldays2002/

In addition to the Lentedagen and Herfstdagen, IPA contributed to the 5th international conference on Formal Methods for Open Object-based Distributed Systems (FMOODS2002), which had a special student workshop where Ph.D. students could present their work in the area of the conference, and to the first international symposium on Formal Methods for Components, Objects and their Implementation (FMCO2002).

On the European front, IPA continued its cooperation in the European Educational Forum (EEF) with the research schools BRICS (Denmark), TUCS (Finland), UKII (United Kingdom), IP (Italy), GEFI (Germany) and FI (France). EEF activities included the EEF Trends School on Massive Data Sets in Aarhus (27 June - 1 July, organised by BRICS), and the EEF Foundations School on Specification, Refinement and Verification in Turku (19 - 31 August, organised by TUCS).

IPA Ph.D. Defenses in 2002

T. Kuipers. *Techniques for Understanding Legacy Software Systems.* Faculty of Natural Sciences, Mathematics and Computer Science, UvA February 26, IPA-Dissertation Series 2002-03

M.C. van Wezel. *Neural Networks for Intelligent Data Analysis: theoretical and experimental aspects.* Faculty of Mathematics and Natural Sciences, UL March 7, IPA-Dissertation Series 2002-01

V. Bos & J.J.T. Kleijn. *Formal Specification and Analysis of Industrial Systems.* Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e March 7, IPA-Dissertation Series 2002-02

S.P. Luttik. *Choice Quantification in Process Algebra.* Faculty of Natural Sciences, Mathematics, and Computer Science, UvA April 3, IPA-Dissertation Series 2002-04

A. Fehnker. *Citius, Vilius, Melius: Guiding and Cost-Optimality in Model Checking of Timed and Hybrid Systems.* Faculty of Science, Mathematics and Computer Science, KUN April 15, IPA-Dissertation Series 2002-08

R.J. Willemen. *School Timetable Construction: Algorithms and Complexity.* Faculty of Mathematics and Computer Science, TU/e April 16, IPA-Dissertation Series 2002-05

M.I.A. Stoelinga. *Alea Jacta Est: Verification of Probabilistic, Real-time and Parametric Systems.* Faculty of Science, Mathematics and Computer Science, KUN April 22, IPA-Dissertation Series 2002-06

R. van Stee. *On-line Scheduling and Bin Packing.* Faculty of Mathematics and Natural Sciences, UL May 8, IPA-Dissertation Series 2002-09

N. van Vugt. *Models of Molecular Computing.* Faculty of Mathematics and Natural Sciences, UL May 26, IPA-Dissertation Series 2002-07

D. Tauritz. *Adaptive Information Filtering: Concepts and Algorithms.* Faculty of Mathematics and Natural Sciences, UL June 25, IPA-Dissertation Series 2002-10

M.B. van der Zwaag. *Models and Logics for Process Algebra.* Faculty of Natural Sciences, Mathematics, and Computer Science, UvA October 11, IPA-Dissertation Series 2002-11

J.I. den Hartog. *Probabilistic Extensions of Semantical Models.* Faculty of Sciences, Division of Mathematics and Computer Science, VUA October 17, IPA-Dissertation Series 2002-12

S. Andova. *Probabilistic Process Algebra.* Faculty of Mathematics and Computer Science, TU/e November 26, IPA-Dissertation Series 2002-15

J.I. van Hemert. *Applying Evolutionary Computation to Constraint Satisfaction and Data Mining.* Faculty of Mathematics and Natural Sciences, UL November 28, IPA-Dissertation Series 2002-14

Y.S. Usenko. *Linearization in μCRL.* Faculty of Mathematics and Computer Science, TU/e December 2, IPA-Dissertation Series 2002-16

L. Moonen. *Exploring Software Systems.* Faculty of Natural Sciences, Mathematics, and Computer Science, UvA December 5, IPA-Dissertation Series 2002-13

Activities in 2003

IPA is looking forward to a busy spring when it will stage its annual Lentedagen (April 23 - 25, at the NH Hotel in Best) and the EEF Summerschool on Concurrency. In addition, IPA will support the organisation of ICALP2003 which takes place in the Netherlands.

EEF Foundations School on Concurrency Working together with fellow research schools BRICS, TUCS, and UKII in the EEF, IPA organises a series of four summer schools on the Foundations of Computer Science. The last event of this series is dedicated to Concurrency and will be hosted by IPA. It will take place at conference center Kapellerput in Heeze from May 19 - 31. The aim of the school is to provide in-depth knowledge to young scientists on the foundations of Concurrency from a number of approaches. For each approach, training in theoretical foundations will be combined with hands-on experience with tools that have been developed within the approach. The school program consists of the following elements. *Process algebra*: lecturers Jos Baeten and Jan Friso Groote (TU/e), tool μCRL toolset; *Modelchecking*: lecturers Dennis Dams (Bell Labs) and Dragan Bošnački (TU/e), tool Spin; *Verification of non-functional properties*: lecturer Kim Larsen (Aalborg University), tool Uppaal; *Theorem proving*: lecturer Jozef Hooman (University of Nijmegen), tool PVS; and *Petri nets*: lecturer Wil van der Aalst (TU/e), tool Woflan. More information on the program of the school and the possibilities for participation grants for young scientists will become available through the IPA website: http://www.win.tue.nl/ipa/activities/EEFschool/.

The EEF Trends series of summer schools will also end in 2003 with the Trends School on Mobile Computing, which will be hosted by UKII and held in Edinburgh from July 7 - 12. For more information on EEF and its activities, see: http://www.win.tue.nl/EEF/.

ICALP2003 The 30th annual meeting of the European Association of Theoretical Computer Science, ICALP 2003 will be held at the Technische Universiteit Eindhoven from June 30 - July 4. As usual there will two tracks to the conference: Track A of the meeting covers Algorithms, Automata, Complexity and Games, while Track B covers Logic, Semantics and Theory of Programming. In the weekends surrounding the conference there is an extensive program of pre- and post-conference workshops. IPA groups are involved in the organisation of a number of these workshops.

ICALP2003 is collocated with two other international conferences on computer science, which will take place from June 23 - 27: the 24th International Conference on Application and Theory of Petri Nets (ATPN 2003), and the Conference on Business Process Management (BPM 2003). More information can be found at: http://www.win.tue.nl/icalp2003/.

Addresses

Visiting address
Eindhoven University of Technology
Main Building HG 7.22
Den Dolech 2
5612 AZ Eindhoven
The Netherlands

Postal address
IPA, Fac. of Math. and Comp. Sci.
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands

tel. (+31)-40-2474124 (IPA Secretariat)
fax (+31)-40-2475361
e-mail ipa@tue.nl url http://www.win.tue.nl/ipa/

5.2 The School for Information and Knowledge systems (SIKS)

Introduction

SIKS is the Dutch Research School for Information and Knowledge systems. It was founded in 1996 by researchers in the field of Artificial Intelligence, Databases and Information Systems and Software Engineering. Its main concern is research and education in the field of information and computing sciences, more particularly in the area of information and knowledge systems (IKS). SIKS is an interuniversity research school that comprises 12 research groups from 10 universities and CWI. When SIKS received its accreditation by KNAW in 1998, only 35 Ph.D. students and about 70 research fellows were involved in the school. Currently, over 240 researchers are active, including about 110 Ph.D.-students. The Vrije Universiteit in Amsterdam is SIKS' administrative university. The office of SIKS is located at Utrecht University.

Management and Organisation

Prof. dr. H. Akkermans (VU) was appointed to be the new chairman of the Board of Governors as of January 1 2003, when prof. dr. R.P. van de Riet (VU) stepped down as chair. Also in 2002, a new University joined SIKS: the group of prof. Th. van der Weide from the department of computer science of KUN became a member. Finally, in 2002 prof. dr. H. van Vliet (VU) decided to step down as chairman of the Scientific advisory committee. Prof. dr. F. van Harmelen (VU) was appointed as his successor.

Apply for re-accreditation

In december 2002 SIKS applied for re-accreditation by KNAW. Earlier, the school took several preparatory steps: it conducted a critical self-evaluation, revised its educational and research programs and was visited by an External Evaluation Committee on September 16-18 2002 in Amsterdam. This committee consisted of three experts in the field of IKS, Gio Wiederhold (Stanford), Arne Slvberg (Trondheim) and Mike Wooldridge (Liverpool) and a chairman, Cor Baayen, who as scientific director of CWI and later as chairman of the VNSU commission to evaluate Mathematics and Computer Science in the Netherlands, has a great experience in evaluations within the Dutch context.

Research Program

In 2002, the research program, as defined in 1997, has proven to be still highly relevant for current research in SIKS, witness a recent survey conducted by the school amongst all its researchers. However, the program needs an update due to new developments in the field and the changing scientific interests of our researchers. Therefore, a new set of eight foci is defined and eight focus-directors were appointed to coordinate activities related to the foci.
The new SIKS foci are:

- Agent technology
- Computational Intelligence
- Knowledge Representation and Reasoning
- Web-based information systems
- E-business systems
- Human computer interaction
- Data management, storage and retrieval
- Architecture-driven system development

Educational Program

To cater for up-to-date topics and issues in the field, SIKS has revised its educational program in 2002; among other things it defined a new program of Basic courses and Advanced courses.

Basic courses

- B0. Research methods and methodology for IKS
- B1. Formal methods/logic for IKS
- B2. System modelling
- B3. Knowledge modelling
- B4. Combinatorial methods
- B5. Learning and reasoning
- B6. Agent technology
- B7. Interactive systems
- B8. Information and organisation
- B9. Information retrieval
- B10. Software architecture

Advanced Courses

- Full-text information retrieval
- Computational intelligence
- Multi-agent systems
- The semantic web
- Intelligent data analysis
- Component based design of Agent systems
- Mobile commerce
- Human Computer Interaction
- E-business systems

Activities in 2002

Basic courses

- Interactive Systems , May 13 - 15, 2002, De Bergse Bossen, Driebergen Scientific director: Prof. dr. P. de Bra (TUE)
- Databases, May 15 - 17, 2002, De Bergse Bossen, Driebergen Scientific director: dr. H. Blanken (UT)
- Information and Organisation, December 9-11 2002, Huize Bergen, Vught Scientific Director: dr. H. Weigand (UvT)
- Information Retrieval, December 11-13 2002, Huize Bergen, Vught Scientific director: prof. dr. T. van der Weide (KUN)

Advanced Courses

- "The semantic web", May 27 - 28, 2002, Conference center Woudschoten, Zeist. Course directors: Prof. dr. F van Harmelen (VU), Dr. D. Fensel (VU)
- Component-based Design of Intelligent Multi-Agent Systems, May 29 - June 4, 2002, Amsterdam. Course directors: Prof. dr. J. Treur (VU), Dr. C. Jonker (VU)
- "Mobile Commerce"(m-Commerce) June 24 and 25, 2002, Amsterdam Course directors: prof. dr. H. Akkermans (VU), Dr. N. Sadeh (VU)

Other activities (co-)organised by SIKS

- Masterclass on Machine Learning, January 10, 2002 Maastricht. (UM).
- SIKS - Strategy-day 2002, February 28, 2002 Conference Center La Vie, Utrecht.
- BNAIS 2002, March 13, 2002, Utrecht, in cooperation with BNVKI.
- Masterclass: "Logical Foundation of the Semantic Web", April 10, 2002, Amsterdam.
- Workshop on computer games, July 6 - 10, 2002, Maastricht (UM).
- Masterclass: "Human Computer Interaction", August 27, 2002, Amsterdam (VU).
- Tutorial/Lecture on Non-monotonic logic, September 11, 2002, Utrecht (UU).
- SIKS-evaluation Days, September 16-18, 2002, Amsterdam (VU).
- Conference: BNAIC2002, October 21 - 22, 2002, Leuven.
- SIKS-day 2002, November 1, 2002, City Castle Oudaen, Utrecht.
- Masterclass: "Security and Privacy in Cyberspace"November 4, 2002, Amsterdam (VU).
- Workshop: Planning and Scheduling, November 21-22, 2002, Delft (TUD).
- Symposium "Contracts and Coordination", November 22, Tiburg (UvT).
- Conference: CLIN 2002, November 29 2002 Groningen (RUG).
- Conference: BENELEARN2002, December 4 2002, Utrecht (UU).
- Third Belgium-Dutch Workshop on Information Retrieval, December 6, Leuven.

Doctoral dissertations in 2002

In 2002 17 researchers successfully defended their Ph.D.-thesis and published their work in the SIKS-dissertation Series.

Nico Lassing (VU). *Architecture-Level Modifiability Analysis.* Promotores: prof. dr. J.C. van Vliet (VU) prof. dr. D.B.B. Rijsenbrij (VU) Promotie: 12 februari 2002

Roelof van Zwol (UT). *Modelling and searching web-based document collections.* Promotor: Prof.dr. P.M.G. Apers (UT) Promotie: 26 april 2002

Henk Ernst Blok (UT). *Database Optimization Aspects for Information Retrieval.* Promotor: Prof.dr. P.M.G. Apers (UT), Co-promotor: Dr. H.M. Blanken (UT) Promotie: 12 april 2002

Juan Roberto Castelo Valdueza (UU). *The Discrete Acyclic Digraph Markov Model in Data Mining.* Promotor: Prof.dr. A Siebes (UU) Promotie: 3 june 2002

Radu Serban (VU). *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents.* Promotor: Prof. dr. H.J. van den Herik (UL / UM) Promotie: 20 June 2002

Laurens Mommers (UL). *Applied legal epistemology; Building a knowledge-based ontology of the legal domain.* Promotor: Prof. dr. H.J. van den Herik (UL / UM) Promotie: 20 June 2002

Peter Boncz (CWI). *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications.* Promotor: prof. dr. M.L. Kersten Promotie: 31 May 2002

Jaap Gordijn (VU). *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas.* Promotor: prof. dr.J.M.Akkermans(VU) Promotie: 25 June 2002

Willem-Jan van den Heuvel(KUB). *Integrating Modern Business Applications with Objectified Legacy Systems.* Promotores: Prof. dr. ir. M.P. Papazoglou (KUB), Prof. dr. P. Ribbers (KUB) Promotie: 14 juni 2002

Brian Sheppard (UM). *Towards Perfect Play of Scrabble.* Promotores: Prof.dr. H.J. van den Herik (UM), Prof.dr. J.Schaeffer (University of Alberta Promotie: 5 July 2002

Wouter C.A. Wijngaards (VU). *Agent Based Modelling of Dynamics: Biological and Organisational Applications.* Promotor:Prof.dr. J. Treur, Co-promotor Dr. C.M. Jonker Promotie: 14 oktober 2002

Albrecht Schmidt (Uva). *Processing XML in Database Systems.* Promotor: prof. dr. M.L. Kersten (Uva/CWI) Promotie: 7 november 2002

Hongjing Wu (TUE). *A Reference Architecture for Adaptive Hypermedia Applications.* Promotores: Prof. dr. ir. P. De Bra (TUE), prof. d.r. ir. L. Hardman (CWI/TUE) Promotie: 8 november 2002

Wieke de Vries (UU) . *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems.* Promotores: Prof. dr. J.-J. Meyer (UU), Prof.dr. J. Treur ((VU); Co-promotores: dr.F.S. de Boer, dr. W. van der Hoek (UU), dr. C.M. Jonker (VU) Promotie: 11 November 2002

Rik Eshuis (UT). *Semantics and Verification of UML Activity Diagrams for Workflow Modelling.* Promotor: Prof. dr. R.J. Wieringa (UT) Promotie: 25 October 2002

Pieter van Langen (VU). *The Anatomy of Design: Foundations, Models and Applications.* Promotores: prof.dr. F.M.T. Brazier (VU), prof.dr. J. Treur (VU) Promotie: 22 November 2002

Stefan Manegold (UVA). *Understanding, Modeling, and Improving Main-Memory Database Performance.* Promotores: prof.dr. M. Kersten (UVA/CWI) Promotie: 17 December 2002

Kinetic Dictionaries: How to Shoot a Moving Target*

Mark de Berg
Department of Computing Science
TU Eindhoven†

Abstract

A kinetic dictionary is a data structure for storing a set S of continuously moving points on the real line, such that at any time we can quickly determine for a given query point q whether $q \in S$. We study trade-offs between the worst-case query time in a kinetic dictionary and the total cost of maintaining it during the motions of the points.

1 Introduction

A *dictionary* is a data structure for storing a set S of elements—the elements are often called *keys*—such that one can quickly decide, for a given query element q, whether $q \in S$. Furthermore, the data structure should allow for insertions into and deletions from the set S. Often the keys come from a totally ordered universe. In this case one can view the keys as points on the real line. The dictionary is one of the most fundamental data structures in computer science, both from a theoretical point of view and from an application point of view. Hence, every algorithms book features various different possibilities to implement a dictionary: linked lists, (ordered) arrays, (balanced) binary search trees, hash tables, and so on.

In this note we study a variant of the dictionary problem, where the keys come from a totally ordered universe but have continuously changing values. In other words, the set S is a set of points moving continuously on the real line. This setting is motivated by a recent trend in the database community to study the indexing of moving objects—see for example [1, 9, 10, 11] and the references therein. Also in the computational-geometry community, the study of data structures for moving objects has attracted a lot of attention recently—see for example [2, 3, 5, 7] and the references therein. The traditional approach to deal with moving objects is to use time-sampling: at regular time intervals one checks which objects have changed their position, and these objects are deleted from the data structure and re-inserted at their new positions. The problem with this approach is two-fold. First, it is hard to choose the right time interval: choosing it too large will mean that important 'events' are missed, so that the data structure will be incorrect for some time. Choosing the interval very small, on the other hand, will be very costly—and even in this case one is likely to miss important events, as these will usually occur at irregular times. Therefore the above-mentioned papers from computational geometry use so-called *kinetic data structures* (KDSs, for short), as introduced by Basch *et al.* in their seminal paper [5]. We also employ the KDS framework. Here we do not describe the KDS framework in detail, but we describe only the concepts needed in this note. A more extensive treatment can be found in the excellent survey by Guibas [6].

The problem we study is the following. Let S be a set of n points moving continuously on the real line. Here 'continuous' does not mean that all points necessarily move all the time—this need not be the case—but rather that the motions are continuous: the value of x_i at time t is a continuous function, $x_i(t)$, of time. We define $S(t) = \{x_1(t), \ldots, x_n(t)\}$. When no confusion can

*Part of this research was done during a visit to Stanford University.
†postal address: P.O.Box 513, 5600 MB Eindhoven, the Netherlands.

arise, we often simply write S and x_i for $S(t)$ and $x_i(t)$. Our goal is to maintain a data structure for S such that, at any time t, we can quickly determine for a query point q whether $q \in S(t)$. We call such a data structure a *kinetic dictionary*. (In this note, we do not consider updates on the set S, so perhaps the name dictionary is a slight abuse of the terminology.) We assume that, at any time t, we can compute the current position $x_i(t)$ of any point x_i in $O(1)$ time. Note that t is not part of the query. This means that we do not allow queries in the past or in the future; we only allow queries about the current set S. The main idea behind KDSs is that, because the objects move continuously, the data structure only needs to be updated at certain *events*. For example, consider the following implementation of a kinetic dictionary: we simply store all the points in a sorted array $D[1..n]$. Then, as long as the order of the points does not change, we can answer a query with a point q in $O(\log n)$ time by doing a binary search in D, using the current values $x_i(t)$ to guide the search. On the other hand, maintaining a sorted array means that whenever two points change their order, we need to update the structure. This is the topic of our work: how often do we need to update a kinetic data structure to be able to guarantee a certain worst-case query time?

In this note we focus on lower bounds for this problem; a more extensive discussion of this problem, including upper bounds for kinetic dictionaries, will be presented in a forthcoming paper. To be able to prove lower bounds, we need to establish a suitable 'model of computation'. To this end we propose in the next section so-called *comparison graphs* as a model for kinetic dictionaries. We then define the cost of answering a query in this model, and the cost of updates. Our interest lies in trade-offs between the worst-case query time and the total maintenance cost of kinetic dictionaries: we want to bound the minimum total maintenance cost, under certain assumptions on the motions, when one has to guarantee worst-case query cost Q at all times. After describing a trivial solution with $O(n^2/Q)$ maintenance cost under the assumption that any pair of points changes order $O(1)$ times, we prove the following lower bound: any kinetic dictionary with worst-case query cost Q must have a total maintenance cost of $\Omega(n^2/Q^2)$ in the worst case, even if all points have fixed (but different) velocities.

2 A comparison-based model for kinetic dictionaries

Before we can prove lower bounds on the query cost and total maintenance cost in a kinetic dictionary, we must first establish a suitable 'model of computation': we must define the allowable operations and their cost. Let $S = \{x_1, \ldots, x_n\}$ be a set of n points on the real line. Our model is comparison-based: the operations that we count are comparisons between two data points in S and between a data point and a query point. Note that we are not interested in a single-shot problem, but in maintaining a data structure to answer queries. Hence, comparisons can either be done when answering a query, or they can be done when constructing or updating the data structure. In the latter case, the comparisons can only be between data points, and the result has to be encoded in the data structure.

The idea of the lower bound will then be as follows. A query asks for a query point q whether $q \in S$. Suppose the answer to this query is negative. To be able to conclude this, we have to know for each $x \in S$ that $q \neq x$. This information can be obtained directly by doing a comparison between q and x; this will incur a unit cost in the query time. It is also possible, however, to obtain this information indirectly. For example, if the information that $x' < x$ is encoded in the dictionary, and we find out that $q < x'$, then we can derive $q \neq x$. Thus by doing a single comparison with q, we may be able to derive the position of q relative to many points in S. This gain in query time has its cost, however: the additional information encoded in the dictionary has to be maintained. To summarize, comparisons needed to answer a query can either be done at the time of the query or they can be pre-computed and encoded in the dictionary; the first option will incur costs in the query time, the second option will incur maintenance costs.

In the remainder of this section we define our model more precisely.

The data structure. For simplicity we assume that all points in S are distinct. Of course there will be times when this assumption is invalid, otherwise the order would remain the same and the problem would not be interesting. But in our lower-bound arguments to be presented later, we will only argue about time instances where the points are distinct so this does not cause any serious problems.

A *comparison graph* for S is a directed graph $\mathcal{G}(S, A)$ with node set[1] S that has the following property: if $(x_i, x_j) \in A$ then $x_i < x_j$. The reverse is not true: $x_i < x_j$ does not imply that (x_i, x_j) must be present in A. Note that a comparison graph is acyclic.

Query cost. Let q be a query point, and let $\mathcal{G} := \mathcal{G}(S, A)$ be a comparison graph. An *extended comparison graph* for q is a graph $\mathcal{G}_q^*$ with node set $S \cup \{q\}$ and arc set $A^* \supset A$. The arcs in A^* are of two types, regular arcs and equality arcs (=-arcs, for short). They have the following property: if $(a, b) \in A^*$ is a regular arc then $a < b$, and if $(a, b) \in A^*$ is an =-arc then $a = b$. Note that for an =-arc (a, b), either a or b must be the query point q, because we assumed that the points in S are distinct. A regular arc may or may not involve q.

An extended comparison graph $\mathcal{G}_q^*$ for q *localizes* q if
(i) it contains an =-arc, or
(ii) for any point $x_i \in S$, there is a path in $\mathcal{G}_q^*$ from x_i to q or from q to x_i.

In the first case, we can conclude that $q \in S$, in the second case that $q \notin S$.

Given a comparison graph $\mathcal{G} = \mathcal{G}(S, A)$ for S and a query point q, we define the *query cost of q in $\mathcal{G}$* to be the minimum number of arcs we need to add to $\mathcal{G}_q = (S \cup \{q\}, A)$ to obtain an extended comparison graph $\mathcal{G}_q^*$ that localizes q. The (worst-case) query cost of $\mathcal{G}$ is the maximum query cost in $\mathcal{G}$ over all possible query points q. Our definition of query cost is justified by the following lemma.

Lemma 2.1 *Let $\mathcal{G}_q^*$ be an extended comparison graph with node set $S \cup \{q\}$ and arc set A^*. Suppose that $\mathcal{G}_q^*$ does not localize q. Then there are values for S and q that are consistent with the arcs in A^* such that $q \notin S$, and there are also values for S and q that are consistent with the arcs in A^* such that $q \in S$.*

Proof. If $\mathcal{G}_q^*$ does not localize q, then there are only regular arcs in A^*. Since $\mathcal{G}_q^*$ is acyclic, there exists a topological ordering of the nodes in the graph. By assigning each node the value corresponding to its position in the topological ordering, we obtain an assignment consistent with the arcs in A^* such that $q \notin S$.

Next we change the values of the nodes to obtain an assignment with $q \in S$. Consider the node $x_i \in S$ closest to q in the topological ordering—ties can be broken arbitrarily—such that there is no path in $\mathcal{G}_q^*$ between x_i and q. Because $\mathcal{G}_q^*$ does not localize q, such a node must exist. Now we make the value of x_i equal to the value of q. Assume that x_i was smaller than q in the original assignment; the case where x_i is larger can be handled similarly. Then the only arcs that might have become invalid by changing the value of x_i are arcs of the form (x_i, x_j) for some x_j that lies between the original value of x_i and q. Such a node x_j must have been closer to q in the topological ordering. By the choice of x_i, this implies that there is a path from x_j to q. But then the arc (x_i, x_j) cannot be in A^*, otherwise there would be a path from x_i to q, contradicting our assumptions. We can conclude that making x_i equal to q does not make any arcs invalid, so we have produced an assignment consistent with A^* such that $q \in S$. □

Note that the query cost does not change when we restrict our attention to the transitive reduction [4] of the comparison graph. This is the subgraph consisting of all *non-redundant arcs*, that is, arcs that are not implied by other arcs because of transitivity. (The transitive reduction of an acyclic graph is unique.)

Our definition of query cost is quite weak, in the sense that it gives a lot of power to the query algorithm: the query algorithm is allowed to consult, free of charge, an oracle telling it which arcs to add to the graph (that is, which comparisons to do). This will only make our lower bounds stronger.

[1] In the sequel we often do not distinguish between a point in S and the corresponding node in $\mathcal{G}_\mathcal{D}$.

Maintenance cost. We define the cost of updating a comparison graph to be equal to the number of new non-redundant arcs, that is, the number of non-redundant arcs in the new comparison graph that were not present in the transitive closure of the old comparison graph.

The rationale behind this is as follows. When using a kinetic data structure, one has to know when to update it. In our case, this is when some of the ordering information encoded in the kinetic dictionary is no longer valid. This happens exactly when the two points connected by an arc in the comparison graph change order; at that time, such an arc is necessarily non-redundant. To know the next time such an event happens, the 'failure times' of these arcs are stored in an event queue. This means that when a new non-redundant arc appears, we have to compute its failure time and insert it into the event queue.

Examples. Next we have a look at some well known dictionary structures, and see how they relate to the model.

First, consider a binary search tree. This structure contains all ordering information, that is, the comparison graph corresponding to a binary search tree in the complete graph on S, with the arcs directed appropriately. The transitive reduction in this case is a single path containing all the nodes. A sorted array on the points has the same comparison graph as a binary search tree; sorted arrays and binary search trees are simply two different ways to implement a dictionary whose comparison graph is the complete graph.

The worst-case query cost of the complete graph is $O(1)$: when $q \notin S$ we need to add at most two regular arcs to localize any query point q (one from the predecessor of q in S and one to the successor of q in S), and when $q \in S$ a single $=$-arc suffices. This is less than the query time in a binary search tree, because we do not charge for the extra time needed to find out which two comparisons can do the job. In an actual implementation we would need to do a binary search to find the predecessor and successor of q in S, taking $O(\log n)$ time.

Our model does not apply to hash tables, since they are not comparison-based: with a hash function we can determine that $q \neq x$ without doing a comparison between q and x, so without knowing whether $q < x$ or $q > x$.

3 A trivial upper bound

Suppose we want to have a kinetic dictionary whose worst-case query cost is Q at all times, for some $2 \leq Q \leq n$. A trivial way to achieve this is to partition the set S into $\lfloor Q/2 \rfloor$ subsets of size $O(n/Q)$ each, and to maintain each subset in a sorted array. Thus (the transitive reduction of) the corresponding comparison graph consists of $\lfloor Q/2 \rfloor$ paths. We need to add at most two arcs to localize a query point in a path, so the total query cost will be at most Q. (The actual query time in the real-RAM model would be $O(Q \log(n/Q))$.) The total maintenance cost is linear in the number of pairs of points from the same subset changing order. If any pair of points changes order $O(1)$ times—which is true for instance when the motions are constant-degree algebraic functions—then this implies that the maintenance cost of a single subset is $O((n/Q)^2)$. We get the following result.

Theorem 3.1 *Let S be a set of n points moving on the real line, and suppose that any pair of points changes order at most a constant number of times. Then there is a comparison graph for S that has worst-case query cost Q and whose total maintenance cost is $O(n^2/Q)$.*

The comparison graph can be implemented such that the actual query time is $O(Q \log(n/Q))$, and the actual cost to process all the updates is $O(n^2/Q)$.

Our main interest lies in the question whether one can improve upon this trivial upper bound.

4 Lower bounds for linear motions

We now turn our attention to lower bounds for kinetic dictionaries in the comparison-graph model. Our goal is to prove lower bounds regarding possible trade-offs between query cost and maintenance cost: what is the minimum amount of work we have to spend on updates if we want to guarantee cost Q always? Of course we will have to put some restrictions on the motions of the points, as otherwise we could always swap a pair of points that defines an arc in the comparison graph.

Here we consider a very limited scenario, where we only allow the points to move linearly. That is, all points have fixed (but possibly different) velocities. In this case we can show that any comparison graph that guarantees query cost at most Q must have a total update cost of $\Omega(n^2/Q^2)$. Our construction is based on the following lemma.

Lemma 4.1 *Let $\mathcal{G}$ be a comparison graph for a set S of n points, and let Q be a parameter with $1 \leq Q \leq n/2$. Suppose $\mathcal{G}$ has query cost Q. Then the subgraph induced by any subset of $2Q$ consecutive points from S contains at least Q non-redundant arcs.*

Proof. Let $x_1 < x_2 < \cdots < x_n$ be the sorted sequence of points in S, and consider a subset $\{x_i, x_{i+1}, \ldots, x_{i+2Q-1}\}$, for some i with $1 \leq i < n - 2Q$. Suppose we want to answer a query with a point q such that $x_i < q < x_{i+1}$. Note that $q \notin S$. In order to localize q, we need to add arcs to $\mathcal{G}$ such that for any point in S there is a path to or from q. In particular, there must be a path between q and each of the points $x_i, x_{i+1}, \ldots, x_{i+2Q-1}$. Such a path cannot contain points x_j with $j < i$ or $j > i + 2Q - 1$. Hence, the subgraph induced by $\{x_i, x_{i+1}, \ldots, x_{i+2Q-1}\} \cup \{q\}$ is connected (when viewed as an undirected graph) after the addition of the arcs by the query algorithm. This means that it contains at least $2Q$ non-redundant arcs. Since the number of arcs added by the query algorithm is bounded by Q by definition, $\mathcal{G}$ must have contained at least Q non-redundant arcs between points in $\{x_i, x_{i+1}, \ldots, x_{i+2Q-1}\}$. □

Lemma 4.1 implies that after the reversal of a group of $2Q$ consecutive points, the graph contains at least Q new non-redundant arcs. Hence, the reversal will induce a maintenance cost of at least Q.

We proceed by exhibiting a set of points moving linearly, such that there are many time instances where a subset of $2Q$ consecutive points completely reverses order. By the above lemma, this will then give a lower bound on the total maintenance cost.

The set of points is defined as follows. We can assume without loss of generality that $n/(2Q)$ is an integer. We have $n/(2Q)$ groups of points, each consisting of $2Q$ points. The points in S_i, the i-th group, are all coincident at $t = 0$: they all have value i at that time. (It is easy to remove this degeneracy.) The points are all moving linearly, so the trajectories of the points in the tx-plane are straight lines. In particular, in each group there is a point whose trajectory has slope j, for any integer j with $0 \leq j < 2Q$. More formally, we have groups $S_1, \ldots, S_{n/(2Q)}$ defined as follows:

$$S_i := \{x_{ij}(t) : 0 \leq j < 2Q \text{ and } j \text{ integer}\}, \quad \text{where } x_{ij}(t) := i + jt.$$

Now consider a point (s, a) in the tx-plane, where a and s are integers with $n/(4Q) < a < n/(2Q)$ and $0 < s \leq n/(8Q^2)$. Then there are exactly $2Q$ trajectories passing through this point. To see this, consider a slope j with j integer and $0 \leq j < 2Q$. Then the line $x(t) = jt + (a - sj)$ passes through (s, a) and the restrictions on a and s ensure that $a - sj$ is an integer with $0 \leq a - sj < n/(2Q)$, so this line is one of the trajectories. We conclude that there are $\Omega(n^2/Q^3)$ points (s, a) in the tx-plane such that $2Q$ trajectories meet at (s, a).

Theorem 4.1 *There is a set S of n points, each moving with constant velocity on the real line, such that, in the comparison-graph model, any kinetic dictionary for S with worst-case query cost Q has total update cost $\Omega(n^2/Q^2)$.*

Proof. In the construction described above there are $\Omega(n^2/Q^3)$ points (s, a) in the ty-plane at which $2Q$ points meet. These $2Q$ points are consecutive just before and just after time s, and their order completely reverses at time s. It follows from Lemma 4.1 that the reversal of one such

group forces the comparison graph to create at least Q new non-redundant arcs within the group. Hence, the total update cost is $\Omega(n^2/Q^2)$. □

5 Discussion

In this note we discussed the problem of maintaining a dictionary on a set of points moving continuously on the real line. We defined a model for such kinetic dictionaries—the comparison-graph model and in this model we studied trade-offs between the worst-case query cost and the total maintenance cost of kinetic dictionaries. In particular, we gave a trivial data structure with query cost Q whose total maintenance cost is $O(n^2/Q)$, assuming any pair of points changes order $O(1)$ time, and we proved that $\Omega(n^2/Q^2)$ is a lower bound on the total maintenance cost of any kinetic dictionary with query time Q, even when each point has a fixed velocity. For fixed velocities, this result is almost tight: using techniques from computational geometry—in particular, results on so-called simplicial partitions [8]—one can design a kinetic dictionary with $O(Q)$ query cost whose total maintenance cost is $O(n^{2+\varepsilon}/Q^2)$ for fixed velocities, for any $\varepsilon > 0$. In fact, this solution even works under the weaker assumption that any pair of points changes order at most once, provided that the complete motions are known in advance. (These issues will be discussed in detail in a forthcoming paper.)

There are several interesting directions to explore. For instance, we could study the case where points can be inserted and deleted, as for standard dictionaries. Here one can probably also use results from computational geometry to achieve similar bounds as mentioned above. The most interesting question is what happens when the motions are not known in advance, or when a pair of points can change order some constant (greater than one) number of times. Can one beat the trivial solution for this case?

Acknowledgement

I would like to thank Julien Basch and Otfried Cheong for stimulating discussions.

References

[1] P.K. Agarwal, L. Arge, and J. Erickson. Indexing moving points. In *Proc. Annu. ACM Sympos. Principles Database Syst.*, pages 175–186, 2000.

[2] P.K. Agarwal, J. Basch, M. de Berg, L.J. Guibas, and J. Hershberger. Lower bounds for kinetic planar subdivisions. *Discrete Comput. Geom.* 24:721–733 (2000).

[3] P.K. Agarwal, S. Har-Peled. Maintaining approximate extent measures of moving points. In *Proc. 12th ACM-SIAM Symp. Discrete Algorithms*, 2001.

[4] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications.* Springer-Verlag, 2001.

[5] J. Basch, L.J. Guibas, and J. Hershberger. Data structures for mobile data. *J. Alg.* 31:1 28 (1999).

[6] L.J. Guibas. Kinetic data structures a state-of-the-art report. In *Proc. 3rd Workshop Algorithmic Found. Robot.*, pages 191 209, 1998.

[7] D. Kirkpatrick and B. Speckmann. Kinetic maintenance of context-sensitive hierarchical representations of disjoint simple polygons. In *Proc. 18th Annu. ACM Symp. Comput. Geom.*, pages 179 188, 2002.

[8] J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.* 8:315 334 (1992).

[9] D. Pfoser, C.J. Jensen, and Y. Theodoridis. Novel approaches to the indexing of moving object trajectories. In *Proc. 26th Int. Conf. Very Large Databases*, pages 395–406, 2000.

[10] S. Šaltenis, C.S. Jensen, S.T. Leutenegger, and M.A. Lopez. Indexing the positions of continuously moving objects. In *Proc. ACM-SIGMOD Int. Conf. on Management of Data*, pages 331–342, 2000.

[11] O. Wolfson, A.P. Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases that track mobile units. *Distributed and Parallel Databses*, pages 257–287, 1999.

Formal Methods for Security Protocols: Three Examples of the Black-Box Approach

C.J.F. Cremers[1], S. Mauw[1] & E.P. de Vink[1,2]

[1] Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
{ccremers,sjouke,evink}@win.tue.nl

[2] LIACS, Universiteit Leiden

Abstract Security protocols are hard to design, even under the assumption of perfect cryptography. With the 'classical' Needham-Schröder protocol as leading example, three so-called black-box approaches to the formal verification of security protocols are sketched. *BAN-logic* is a light-weight method under the assumption of honest agents and a passive intruder. The *Casper/FDR* approach translates a high-level description of a security protocol, together with its security requirements and a particular instantiation into CSP, that can be machine-verified using the FDR model checker. In this approach the intruder is in control of the network and is allowed to participate as one of the agents. The same holds for the *strand space* approach. By focusing on the causal dependency of actions and message passing, one aims to prove security properties of general class of protocols instantiations at the same time.

Keywords Security protocols, formal verification, BAN-logic, Casper, FDR, strand spaces

1 Introduction

Security protocols are notoriously known to be difficult to get right. Many protocols proposed in the literature and many protocols exploited in practice turned out to be flawed, or their well-functioning was found to be based on implicit assumptions. Since the late eighties various approaches have been put forward for the formal verification of security protocols to overcome the problems of faulty implementations and hidden requirements.

In this contribution we will focus on three methods for the formal verification of security protocols: BAN-logic, the Casper/FDR-approach and strand spaces. These methods, among others, have in common that they abstract away from cryptographic details. Instead they assume that suitable cryptographic primitives like encryption and decryption, digital signing and verification of signatures are given and are cryptographically safe. So, considerations of weak RSA-moduli or smooth primes numbers, for example, are not part of the investigations. Of course, these aspects are relevant to the safety of a security protocol. No application should be packed and shipped before the crypto-analytic weaknesses of the various underlying algorithms have been carefully pondered about and the subsequent risks have been weighted against various odds and evens. However, even when abstracting away from the cryptographic issues by starting from their complete safety, security protocols can still go to wreck.

A classical example in the short history of security protocols is the protocol due to Needham and Schröder. One of the aims of the protocol run by two, possibly unacquainted parties, Alice and Bob say, is that they can authenticate each other on the basis of their public keys. The protocol presumes a public/private key infrastructure, which means that every agent

has a secret private key and a corresponding public key which is known to all other agents. The protocol is explained by means of a message sequence chart in Figure 1. Encryption of a message m by key K is denoted as $\{m\}_K$. We assume Alice and Bob to hold or be

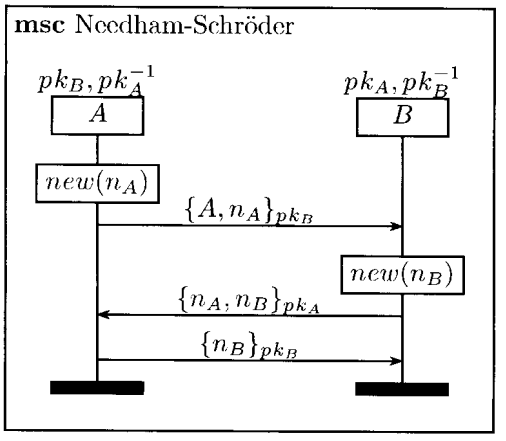

Figure 1: Message sequence chart of the Needham-Schröder protocol

able to look up the public keys of each other. Alice creates a new so-called nonce, a fresh random number n_A. Then Alice sends an initiating message to Bob consisting of her name and the nonce, both encrypted with the public key pk_B of Bob. In this way, Alice can make sure that only Bob can unpack the message and learn the nonce n_A. Bob, assumed to be in the possession of the private key pk_B^{-1} corresponding to the public key pk_B, decrypts the incoming message and retrieves two components, viz. the nonce n_A and the identity of Alice. Knowing which protocol to follow, Bob creates a fresh nonce of his own and replies to Alice by adding his nonce to the one of Alice, both encrypted with the public key pk_A of Alice. The fact that he replies to the previous message in the correct way serves to give evidence to Alice that he is really Bob. Alice decrypts the message with her private key, checks that it contains her original nonce n_A and finds a new nonce, apparently from Bob. She responds to Bob by sending out this latter nonce encrypted under the public key of Bob. Thus, she proves her identity to Bob. So, at the end of the protocol both Alice and Bob are convinced that they are engaged in a protocol run with each other and that one nonce is coming from Bob and the other is coming from Alice.

In the remainder of this paper we want, by means of illustration of the various formal approaches, to verify a single security property of the Needham-Schröder protocol only. It will turn out that the underlying security model, that is, the capabilities of the intruder, is crucial to the proper interpretation of the merits of the protocol. We start off with the method set out by Burrows, Abadi and Needham.

Note The reported here is no original research of the authors. For activities of the Eindhoven Computer System Security group see `http://www.win.tue.nl/~eccs`.

2 BAN-logic

BAN-logic, named after its inventors, is a first-order modal logic with modalities of observation '$\triangleleft$', '$|\sim$' and '$\vdash$' and a modality of belief '$\equiv$'. In the presentation here, we restrict

to predicates expressing possession '$\ni$', key binding 'pk', freshness '$fresh$' and sharing of secrets '$secret$'.

Within the logic, statements can be made about a current run of a protocol in an imperfect network eavesdropped by an intruder. The intuition of the formula $P \lhd X$, read as 'P is told X', is that agent P receives X either as an entire message or as a submessage that he can distill. A distinction is made between the present and the past. The current run takes place in the present; all other runs have taken place in the past. For example, the formula $P \mid\sim \phi$ reads as 'P once said ϕ, either now or earlier', whereas the formula $P \mid\!- \phi$ states that 'P uttered ϕ in the current run'. The operator '$\models$' is used for constructs as $P \models \phi$, meaning that P is entitled to believe that ϕ holds. Formulas that are derivable in the logic are valid statements about the current run of the protocol irrespective of the residuals of possible earlier runs. In general, we want to prove statements of the form $P \models Q \mid\!- \phi$ or $P \models Q \models \psi$, i.e. that agent P is entitled to believe that agent Q said something in the current run of the protocol or that agent P legitimately believes that agent Q is at present entitled to believe something else.

Other ingredients that we consider in the BAN-analysis of the Needham-Schröder protocol: freshness predicates $fresh(X)$ stating that a message has been generated in the present, hence not in the past; possession formula $P \ni X$ for some key, nonce or name X expressing that agent P is in possession of or knows that key, nonce or name; public-key claims $pk(P, K)$ that the public key K is associated with agent P; statements of shared secrets $secret(X, P, Q)$ saying that nonce or key X is a secret shared by agents P and Q.

As deduction rules we will focus on the following three rules:

$$\text{(Dec)} \quad \frac{P \lhd \{X\}_K, \quad P \ni K^{-1}}{P \lhd X} \qquad \text{(SS)} \quad \frac{P \lhd X, Y, \quad P \models secret(X, P, Q)}{P \models Q \mid\sim X, Y}$$

$$\text{(Say)} \quad \frac{P \models Q \mid\sim X, \quad P \models fresh(X)}{P \models Q \mid\!- X}$$

The decryption rule (Dec) states that if agent P has received a message X encrypted with the key K and P is in possession of the decryption key corresponding to K, P has received X. The shared secret rule (SS) expresses that if agent P is told the message consisting of X and Y and P believes X to be a secret it shares with agent Q, then the message X, Y has been sent (either now or in the past) by agent Q. Note that the soundness of the rule assumes that the intruder is not capable of manipulating and sending out messages himself. Otherwise, the intruder simply replaces the component Y by his favorite component Z, making agent P to believe that agent Q once sent X, Z. The presence of fresh items pinpoints the sending of messages at the present as captured by the rule (Say): if agent P believes that agent Q once said message X and P also believes that X has been generated in the current run, then P believes that Q uttered the message X in fact in the current run.

The security goal of the Needham-Schröder protocol that we are focusing on can be rephrased in terms of the BAN-logic as $B \models A \mid\!- n_B$, i.e. Bob believes that Alice said n_A, in the last step of the protocol. From this it follows that Alice was engaged in the protocol with Bob. A proof in BAN-logic that the protocol supports this runs as follows: First we list the initial knowledge. We assume $A \models pk(B, pk_B)$, $A \ni pk_A^{-1}$, i.e. A believes that the public-key of B is the key pk_B and that A possesses his private key. Likewise for Bob, $B \models pk(A, pk_A)$, $B \ni pk_B^{-1}$. After the first action of the Needham-Schröder protocol we have

that $A \models fresh(n_A)$ as Alice has generated it herself in the current run. Next, after the first message sent out by Alice, we obtain $A \models secret(n_A, A, B)$ and $B \triangleleft \{A, n_A\}_{pk_B}$, meaning, respectively, that Alice believes that she shares the secret n_A with Bob as she has sent it under Bob's public key, and, that Bob receives the name A together with the nonce n_A encrypted with his public key. The formulas $A \models fresh(n_A)$, $A \models secret(n_A, A, B)$ and $B \triangleleft \{A, n_A\}_{pk_B}$ are typical of what 'axioms' can be obtained directly from the protocol: nonce or key generation, exchange of secrets, receipt of message. Using these kinds of facts we have, for example, the following derivation in Figure 2.

(1)	$B \triangleleft \{n_B\}_{pk_B}$	(message 3)
(2)	$B \ni pk_B^{-1}$	(assumption)
(3)	$B \triangleleft n_B$	(1, 2, PK)
(4)	$B \models secret(n_B, A, B)$	(message 2)
(5)	$B \models A \mid\sim n_B$	(3, 4, SS)
(6)	$B \models fresh(n_B)$	(action 2)
(7)	$B \models A \mid\sim n_B$	(5, 6, Say)

Figure 2: Proof in BAN-logic of $B \models A \mid\sim n_B$

In the proof we have exploited all the rules mentioned above, initial knowledge of the agent A, and results of the preceding action and message. The conclusion is that Alice rightly believes that Bob is involved in the protocol. Stated otherwise, Bob is authenticated by Alice. As touched upon in the discussion of the shared secret rule, the underlying model assumes that A and B are honest, i.e. behave according to the protocol, and that the intruder can overhear but not disturb the sending and receiving of messages. In the next section we will reconsider this and assume the intruder to be substantially more powerful.

3 Casper/FDR

The Needham-Schröder protocol was published in 1978. It came as a surprise that Lowe found a weakness in 1996 only, as the protocol was considered to be safe. However, in the late seventies computer networks were closed networks. This had changed completely in the mid nineties. The setting of computer networks had changed from closed to open networks. As a consequence the capabilities that should be attributed to an intruder had grown, and, unfortunately, the BAN-analysis of the Needham-Schröder protocol is not valid in this context. it is relatively simple to intercept, alter and spoof ethernet or Internet traffic.

Nowadays the standard setting is that agents are not necessarily honest and that the network is under control of the intruder who can block, inject and invent messages. However, cryptography is assumed to be perfect. The intruder, and all others involved, can only see the contents of an encrypted message if they are in possession of the corresponding decryption key. This is the black box assumption.

The attack of Lowe is a so-called man-in-the-middle attack, in which the intruder fools Bob to think that he is talking to Alice where he is in fact talking to the intruder. And what is worse, Bob might subsequently reveal secrets to the intruder that are only to the discretion of Alice and Bob. The attack assumes that the intruder has a public-key private-key pair and

an identity I of its own in the network, and that Alice initiates a session with I. See Figure 3 for the message sequence chart. Note that no cryptographic tricks have been used. Simply by

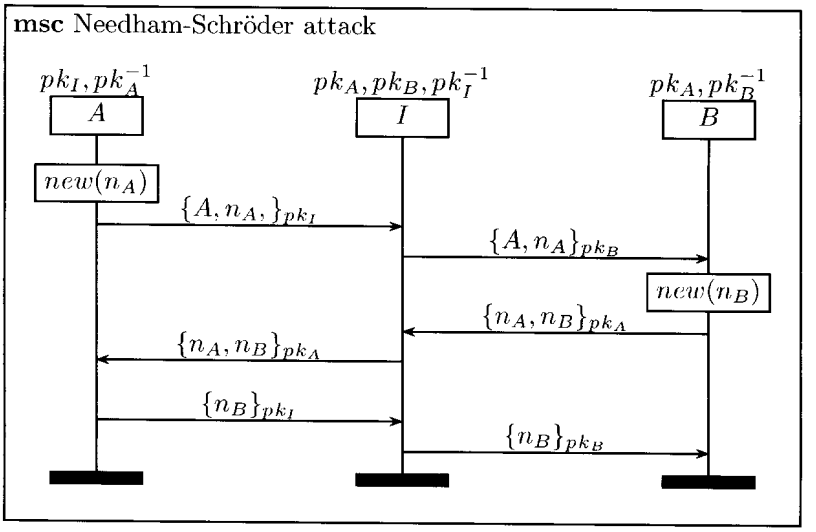

Figure 3: Man-in-the-middle attack on the Needham-Schröder protocol

redirecting messages while running two runs of the protocol in parallel, the intruder manages to impersonate himself to Bob.

Finding attacks mechanically with a tool is one of the appealing characteristics of the Casper/FDR approach. The Casper tool, developed by Lowe, translates a high-level description of a protocol into CSP, the process algebra of Communicating Sequential Processes developed by Hoare and co-workers. The CSP-program comprises various processes representing the agents involved in the protocol that run in parallel with an intruder that controls the network. Additionally the CSP-program contains the specification of the security requirements the protocol should meet.

The CSP-program is fed in to the FDR-tool, an industrial-size modelchecker for CSP. The modelchecker checks for all the traces obtained from interleaving of the possible actions of the agents and intruder (blocking, forwarding, and spoofing of messages) that they fit the format described by the security requirements. So, FDR verifies that the implementation is a trace refinement of its specification. If not, a counter example is found. Casper can translate the FDR-output in human-readable form, yielding a blueprint for an attack on the protocol. This Casper-FDR-Casper chain can be automated through the use of the CasperFDR script.

A Casper input file, see Figure 4, consists of a general part (variable and process declaration, protocol description and specification of requirement) and a specific part that instantiates the general part and specifies the intruder knowledge. The free variables section contains the functions `pk` and `sk` that associate a public key and a secrete or private key with each agent. The inverse keys declaration specifies that for any particular agent these keys are the inverses of each other. The processes section introduces an initiator process with the agent `A` and the nonce `nA` as parameters. Moreover, the process has all public keys and the private key of the agent `A` at its disposal. Likewise, a responder process is declared. The protocol section gives the protocol itself. In the step '`0. -> A : B`' the environment, say the user of the machine `A`, is providing the name `B`. The security requirements can be found in the specification section. Here we have only have one agreement requirement, stating that the

```
-- Needham Schroeder

#Free variables
A, B : Agent
nA, nB : Nonce
pk : Agent -> PublicKey
sk : Agent -> SecretKey
InverseKeys = (pk, sk)

#Processes
INITIATOR(A,nA) knows pk, sk(A)
RESPONDER(B,nB) knows pk, sk(B)

#Protocol description
0.    -> A : B
1.  A -> B : {A, nA}{pk(B)}
2.  B -> A : {nA, nB}{pk(A)}
3.  A -> B : {nB}{pk(B)}

#Specification
Agreement(B,A,[nA,nB])
```

```
#Actual variables
Alice, Bob, Ivey : Agent
Na, Nb, Ni : Nonce

#Functions
symbolic pk, sk

#System
INITIATOR(Alice, Na)
RESPONDER(Bob, Nb)

#Intruder Information
Intruder = Ivey
IntruderKnowledge =
  {Alice, Bob, Ivey, Ni, pk, sk(Ivey)}
```

Figure 4: Protocol description of Needham-Schröder in Casper

agent `A` is authenticated to agent `B` and that they agree on the values of the nonces `nA` and `nB`.

In the instantiation of the general part we provide, in the actual variables section, besides two honest agents `Alice` and `Bob` also a third agent `Ivey` that will play the role of the intruder. We will not use specific properties of the `pk` and `sk` functions apart from being each others inverse. The systems section specifies the processes that comprise the system, in this case `Alice` as initiator and `Bob` as responder. The process for the intruder `Ivey` is implicit. The process is inserted by Casper into the CSP file. The intruder can do whatever he likes based on the knowledge he has gathered so far.

The Casper compiler checks that the protocol is feasible, i.e. that parties are able to send out the messages as specified. This is related to the active role of the intruder. The intruder can inject any message that it can construct at that time. Therefore, a mechanism should be in place that controls the nondeterminism of the intruder. It is also important to eliminate improper traces as soon as possible because of the state space explosion problem.

When the protocol specification for the Needham-Schröder protocol as given in Figure 4 is processed through the CasperFDR script, a violation to the security requirement is found. The particular trace found by FDR is interpreted by Casper, see Figure 5. The intruder plays both the role of Ivey and of Alice, thereby successfully masquerading its identity to Bob, who believes he is dealing with Alice.

With the Casper/FDR machinery available one can easily play with the protocol and try to find an improvement of the Needham-Schröder protocol that is not amenable to the man-in-the-middle attack. This way, the solution proposed by Lowe himself can be recovered. The message sequence chart of the improved Needham-Schröder-Lowe protocol is given in Figure 6. The repair consists of including the responder's identity in the second message. For this new protocol and the particular instance of the system the modelchecker does not discover violating traces any more.

```
Attack found:
Top level trace:
  Alice believes she is running the protocol,
  taking role INITIATOR, with Ivey, using data items Na, Nb
  Bob believes he has completed a run of the protocol,
  taking role RESPONDER, with Alice, using data items Na, Nb

System level:
0.         ->  Alice   : Ivey
1.  Alice  -> I_Ivey   : {Alice, Na}{pk(Ivey)}
1. I_Alice ->   Bob    : {Alice, Na}{pk(Bob)}
2.   Bob   -> I_Alice  : {Na, Nb}{pk(Alice)}
2. I_Ivey  ->  Alice   : {Na, Nb}{pk(Alice)}
3.  Alice  -> I_Ivey   : {Nb}{pk(Ivey)}
3. I_Alice ->   Bob    : {Nb}{pk(Bob)}
```

Figure 5: CasperFDR output indicating the man-in-the-middle attack

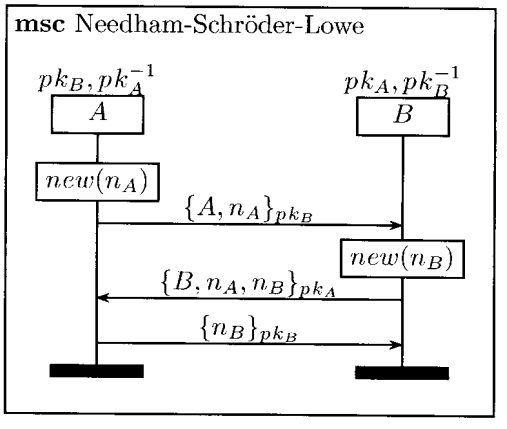

Figure 6: Message sequence chart of the Needham-Schröder-Lowe protocol

Note that in the system instantiation in Figure 4 only one initiator and one responder are specified. This is an important restriction of this approach. If we want to analyze the situation with two protocols running in parallel, say between Alice and Bob and between Alice and Claire, we would write, e.g.,

```
INITIATOR(Alice, Na1)
RESPONDER(Bob, Nb)
INITIATOR(Alice, Na2)
RESPONDER(Claire, Nc)
```

The guarantee of the Casper/FDR analysis restricts to the particular case as specified (with a strong intruder), whereas the BAN-analysis quantifies over all possible past behaviors (but with a weaker intruder).

In general, machine tools can only analyze a finite search space. Casper enforces this by analyzing a protocol for a specific number of protocol runs. Another restriction of the approach is that Casper comes with a number of predefined security requirements. To correctly model a protocol, a thorough understanding of the subtleties of the security requirements in Casper is needed.

4 Strand spaces

The last method of analyzing security protocols that we discuss here is the so-called strand spaces approach as advocated by Thayer Fábrega, Herzog and Guttman. Strand spaces are also based on the so-called Dolev-Yao model where the intruder is a first-class citizen and, moreover, in control of the network. In the Casper/FDR approach a specific system instantiation is considered in isolation. In the strand spaces approach one seeks, instead, to prove properties of arbitrary combinations of protocols running at the same time.

In the strand space approach agent activity is modelled as a number of sequential threads put in parallel. These threads are the strands. The nodes on the strands are the actions the agent performs. The sequence of actions along the strand is referred to as its trace. The nodes on a single strand are causally related. Between the strands there is in general the sending and receiving of messages. So, also between a sending node and a receiving node there is a causal relationship. Now, a strand space is a collection of strands reflecting the activity of honest agents involved in one or more protocols together with a number of strands of the intruder. The nodes on all the strands together form a partial order when endowed with the causality relation induced by the sequentiality on a single strand and the sending and receiving of messages. The strands of the intruder are of a restricted form to be given in a minute.

In Figure 7 a strand space is depicted showing the attack on the Needham-Schröder protocol as described in Section 3. The double lines are strands, the nodes of which are executed in top to bottom order; the isolated bullets are single node strands. The signs attached to nodes indicate whether the node is an input or an output node. The minus sign of an input node indicates that this node represents the receipt of a message. A plus sign of an output node indicates that this node represents the sending of a message or that the particular message belongs to the knowledge of the agent. This latter situation is for example the case in the single node strands $+pk_I^{-1}$ and $+pk_B$. The arrows connect an input node with a corresponding output node. In a strand space, by definition, each input node must be connected to a unique output node. There are no messages of 'outer space'.

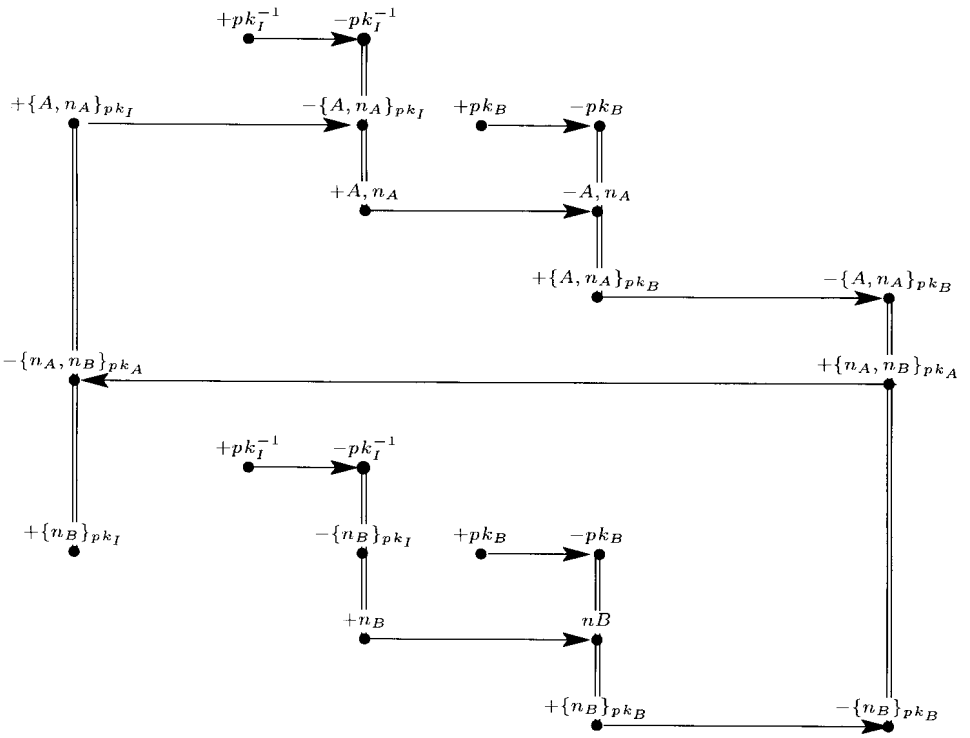

Figure 7: Needham-Schröder strand space

In Figure 7, the left-most and right-most strand are 'regular' strands expressing the activity of honest users, where as the remaining six strands in the middle are 'penetrator' strands. They represent intruder activity. The regular strands are given by the prescribed behaviour of the agents. The penetrator strands are strands with the following traces: text message or knowledge $\langle +t \rangle$, flushing $\langle -x \rangle$, tee $\langle -x, +x, +x \rangle$, concatenation $\langle -x, -y, +xy \rangle$, separation $\langle -xy, +x, +y \rangle$, encryption $\langle -k, -x, +\{x\}_k \rangle$, and decryption $\langle -k^{-1}, -\{x\}_k, +x \rangle$. These strands express the various capabilities of the intruder. The intruder can output any text or message it knows. The intruder can absorb or block messages via the flushing strand and it can make copies using the tee strand. Combination of messages or separation of messages is done with the corresponding strand. The encryption strand can be read as, given a key k and given a message x the intruder is capable of sending the message $\{x\}_k$. Similarly for a decryption strand, given the inverse key k^{-1} and an encrypted message $\{x\}_k$, the intruder can recover the contents x.

The attractive characteristic of the strand space approach is that properties of a whole class of protocol instantiations can be proven, instead of one instance at the time. For example, for the improved Needham-Schröder-Lowe protocol, one can prove on the basis of the causal structure of the strand spaces involved that if a responder can finish his part of the protocol then, under reasonable conditions, there must have been an agent running the protocol with the same parameters as an initiator.

More precisely, let $Init[A, B, n_A, n_B]$ be a strand of the format

$$\langle +\{A, n_A\}_{pk_B}, -\{B, n_A, n_B\}_{pk_A}, +\{n_B\}_{pk_B} \rangle$$

and $Resp[A, B, n_A, n_B]$ the complementary strand of the form

$$\langle -\{A, n_A\}_{pk_B}, +\{B, n_A, n_B\}_{pk_A}, -\{n_B\}_{pk_B} \rangle$$

Let Σ be any strand space based on initiator strands and responder strands as described, with A, B ranging over agent names and n_A, n_B ranging over nonces, and any number of penetrator strands of the seven formats given. If Σ contains a strand of the form $Resp[A, B, n_A, n_B]$, Σ does not have a knowledge strand $\langle +pk_A^{-1} \rangle$ and there is no other strand at which the nonce n_B is created as well, then Σ contains necessarily an initiator strand $Init[A, B, n_A, n_B]$. I.e., if the private key of the initiator A is not revealed and there is no confusion about the nonce n_B used by the responder B, then, upon completion of the protocol, the responder can be sure that the initiator has finished her part of the protocol using the nonce n_A.

The proof of this authentication property is based on a case analysis of the possible penetrator strands. The cutting down of the intruder behavior to specific atomic forms is instrumental to this. No specific assumptions, except for the one stated, are needed on the intruders actions. Starting from an input node of the responder the information flows, as it were, backwards into the intruder strands, thereby instantiating the parameters of the strands. Then, based on the secrecy of the private keys and the form of the actions of the initiator and responder strands all 'bad' situations in the strand space are ruled out. It should be noted that the proof indeed makes use of the form of the adapted message $\{B, n_A, n_B\}_{pk_A}$, confirming that the proof does not apply to the original Needham-Schröder protocol.

5 Conclusions

Above we have given a brief outline of three complementary approaches to formal verification of security protocols. Each approach has different limitations. In all the three methods described cryptographic details were abstracted away. It is simply assumed that an agent, including the intruder, is not capable of decrypting a message without holding the proper key. This is what is referred to as black-box cryptography.

First, the logical approach of Burrows, Abadi and Needham, that is valid in a setting of honest agents, in which the intruder is only capable of overhearing the messages. There are two drawbacks to this method. Defining a semantics for the logic is still a topic of ongoing research. In particular, a semantics combining Krypke-structures and action updates would be attractive to have. However, the underlying intruder model is significantly weaker than the common Dolev-Yao model.

Second, the modelchecker based approach of Casper/FDR in which the front-end Casper compiles a high-level description in a CSP program, that can be proven or disproven to be a trace-refinement of its specification, with the FDR-tool. Casper provides a powerful intruder model and a user-friendly interface, but choosing the right Casper requirements requires thorough knowledge of the subject. The number of protocols runs and specific instantiations must be explicitly modeled in the specifications to avoid an infinite search space.

Third, the strand spaces approach where agent activities are seen as a partial order and the intruder behavior is fragmented into basic steps, allowing for a causality based reasoning. Although mathematically appealing, the state-of-the-art is rather ad-hoc. No general method of evaluating security prperties has been emerged so far. Directly related to this is the lack of machine support for the analyis of security protocol in the strand space approach.

We are currently researching security protocol models that do not have these restrictions. Whether an approach is possible that includes the strengths of the methods mentioned here, whilst avoiding their weaknesses, is still an open question.

6 Bibliographic remarks

In this overview we have been short and, for the sake of presentation, been imprecise at certain points. The reader can consult the following literature for more detail: In [2] an authentication logic is discussed that has later been baptized BAN-logic by others. In the accessible [5] an early variation of BAN-logic is presented. The formal semantics of BAN-logic, based on so-called runs, is addressed in [1, 23]. Other extensions of the logic can be found in [11, 19, 17]. The application of CSP to security protocols was pioneered by Roscoe and Lowe [14, 9, 15] using the CSP model checking tool FDR and the Casper compiler [18, 10]. Using these tools the catalog of Clark and Jacob [3, 8] has been exhaustively analyzed in a student project [4]. The strand space approach is advocated by Thayer Fábrega, Herzog and Guttman in a series of papers [21, 22, 6]. The approach has various ideas in common with the inductive approach of Paulson [12, 13]. The latter uses the theorem prover Isabelle for the computer aided verification of security protocols. General text books that we would like to mention here include [20, 7, 16].

References

[1] M. Abadi and M. Tuttle. A semantics for a logic of authentication. In *Proc. Principles of Distributed Computing*, pages 201 216, Montreal, 1991.

[2] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8:18 36, 1990.

[3] John A. Clark and Jeremy L. Jacob. A survey of authentication protocol literature. Technical Report 1.0, Department of Computer Science, University of York, 1997.

[4] B. Donovan, P. Norris, and G. Lowe. Analyzing a library of security protocols using Casper and FDR, 1999.

[5] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocl analysis. In *Proc. IEEE Symposium on Research in Security and Privacy*, pages 234–248, 1990.

[6] J.D. Guttman and F.J. Thayer. Authentication tests and the structure of bundles. *Theoretical Computer Science*, 238:333 380, 2002.

[7] C. Kaufman, R. Perlman, and M. Speciner. *Network security: private communication in a public world (2nd ed.)*. Computer Networking and Distributed Systems. Prentice Hall PTR, 2002.

[8] Laboratoire Spécification et Vérification, ENS de Cachan. Security protocols open repository. http://www.lsv.ens-cachan.fr/spore/.

[9] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proc. TACAS*, pages 147 166. LNCS 1055, 1996.

[10] Gavin Lowe. Casper: A compiler for the analysis of security protocols. In *Proc. 10th IEEE Computer Security Foundations Workshop*, 1997. http://web.comlab.ox.ac.uk/oucl/work/gavin.lowe/Security/Casper/index.html.

[11] P.C. van Oorschot. Extending cryptographic logics of belief to key agreement protocols. In *1st ACM Conference on Computer and Communications Security*, pages 232 243, Fairfax, Virginia, 1993.

[12] L. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85–128, 1998.

[13] L.C. Paulson. Proving properties of security protocols by induction. In *Proc. 10th Computer Security Foundations Workshop*, pages 70–83, Rockport, Massachusetts, 1997.

[14] A.W. Roscoe. Modelling and verifying key-exchange protocols using CSP and FDR. In *Proc. 8th IEEE Computer Security Foundations Workshop*, pages 98–107, Kenmare, 1995.

[15] P.Y.A. Ryan and S.A. Schneider. *Modelling and Analysis of Security Protocols: the CSP Approach.* Addison-Wesley, 2001. With M.H. Goldsmith, G. Lowe and A.W. Roscoe.

[16] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.).* Wiley, 1995.

[17] S.G. Stubblebine and R.N. Wright. An authentication logic with formal semantics supporting synchronization, revocation, and recency. *IEEE Transaction on Software Engineering*, 28:256–285, 2002.

[18] Formal Systems. http://www.formal.demon.co.uk/FDR2.html.

[19] P.F. Syverson and P.C. van Oorschot. A unified cryptographic protocol logic. CHACS Report 5540-227, NRL, 1996.

[20] G. Tel. *Cryptografie: Beveiliging van de digitale maatschappij.* Pearson Education, 2002. In Dutch.

[21] F.J. Thayer Fábrega, J.C. Herzog, and J.D. Guttman. Strand spaces: Why is a security protocol correct? In *Proc. 1998 IEEE Symposium on Security and Privacy*, pages 66 77, Oakland, California, 1998.

[22] F.J. Thayer Fábrega, J.C. Herzog, and J.D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7:191–230, 1999.

[23] G. Wedel and V. Kessler. Formal semantics for authentication logics. In E. Bertino, H. Kurth, and G. Martella, editors, *Proc. ESORICS'96*, pages 219 241. LNCS 1146, 1996.

The Systems Validation Centre in Retrospect

Henk Eertink (Telematica Instituut) Wan Fokkink (CWI)
Izak van Langevelde (CWI) Holger Hermanns (Univ. Twente)

Abstract

Solid theoretical foundations, state-of-the-art tools and industrial case studies were among the buzzwords of the blueprints of the Dutch Systems Validation Centre (SVC) when it was launched four years ago [11]. Now that the SVC has closed according to plan per December 31, 2002, it is time to take stock.

1 Introduction

In 1999, the Systems Validation Centre was initiated as a joint effort of Telematica Instituut, the Embedded Systems Group at CWI and the Formal Methods and Tools Group at the University of Twente, with support through industrial partners, including CMG, IBM, KPN and Lucent [11]. The rationale of this structure was to have the academic partners lay the theoretical foundations for the development of tools and techniques, and to have Telematica Instituut bring in, through its contacts, the industrial case studies.

This paper gives an overview of the SVC research on foundations, tools and cases. It contains a list of theses and publications in which SVC was involved.

2 Foundations

The theoretical spectrum of the SVC is spanned by two foundational cornerstones: process algebra with abstract data types and stochastic process algebras. The former is centered around the specification language μCRL, developed at CWI, while the latter is concentrated in the work on compositional Markov model generation and analysis, developed in Twente.

The fundamental work on μCRL concentrated on the transformation of a specification into an equivalent specification without parallel merge, communication, encapsulation and abstraction. It is this so-called linear format which is the pivot in the μCRL toolset, in the sense that all tools benefit from this format. The previously existing transformation [18] was restricted to a subset of μCRL, which was useful for a rich variety of system behaviour, but was not able to express, for instance, dynamic process creation. With the extension of this transformation to arbitrary μCRL specifications, the full spectrum of system behaviour is made accessible to the μCRL tools. This research was the main theme of Yaroslav Usenko's PhD thesis [3]; it was published in [35].

Work on compositional specification of stochastic models focused on the theory of interactive Markov chains [19,21] and of stochastic automata [13]. Both models constitute generalisations of different stochastic processes which occur in standard performance and discrete event simulation approaches, but is fully compositional. The stochastic automata model and the process algebra SPADES have been the core contribution of the PhD thesis of Pedro D'Argenio [1]. Related activities include algorithms to perform model checking of

stochastic processes [6,7], and to minimise such models [23], as well as results answering the question what constitutes an adequate form of composition of discrete stochastic models [12].

Further foundational research was performed on finite partial-order unfoldings for process algebra [26] and on flow analysis for model checking asynchronous systems [25].

3 Tools

The two main pillars of the tools developed in SVC directly rest on the two foundation stones of theoretical research: the μCRL toolset and the tools developed for compositional Markov model construction and model checking.

The μCRL toolset consists of efficient tools for linearisation, state space generation, optimisation, reduction, theorem proving and model checking [8]. One example of a promising development is the design and implementation of distributed tools for generation and reduction [9], which shifted the dimensions of systems that can be successfully analysed with several orders of magnitude. A second example worth mentioning is the fruitful combination of theorem proving and state space generation in a state space generator which benefits from the marking of confluent τ steps to generate a reduced state space [10]. Third, a number of important optimisation techniques for linear equations were implemented [16]. Fourth, a rewriter based on strategy annotations was developed [30,31]. A comparison of the generators in the μCRL and SPIN toolsets was made on the basis of a leader election protocol within the PIEEE 1394 Standard for Home Audio/Video Interoperability [34].

Major parts of the PhD thesis of Theo Ruys [2] are devoted to the *art of modelling* in the context of exhaustive verification techniques, using the SPIN tool, while [33] introduced a project management system for SPIN. The TIPPTOOL for modelling and evaluation of performance and dependability models was improved substantially within the SVC project [22]. The ETMCC model checker [20], the first verification tool for Markov models and continuous stochastic logic, was developed and successfully applied to case studies.

4 Case Studies

Firm foundations and state-of-the-art tools facilitated bridging the gap between theory and practice. In a number of industrial case studies, the expressive and analysing strength of theory and tools were assessed and improved.

Transaction Capabilities Procedures (TCAP) are part of the Signalling System No. 7, of which an optimisation was implemented in Erlang by Ericsson. Both the original TCAP and its optimisation were specified in μCRL, after which the μCRL (and the French Caesar/Aldebaran) tools were used to check equivalence of the two specifications. As a result, a number of small bugs were localised and fixed; one of these involved a memory leak which would have been hard to track down by conventional means [4,5].

Coordination architectures served as a fruitful domain for a number of cases. A detailed specification of Splice, developed by Thales was written in μCRL and model checked using Caesar/Aldebaran in order to establish soundness and completeness of read/write actions [14]. Other topics covered are transparent replication [24], JavaSpaces [32] and distribution of shared data spaces [28].

The verification of the IEEE P1394.1 Draft Standard for High Performance Serial Bus Bridges is a large case, in cooperation with the Technical University Eindhoven. This study targeted the correctness of the net update procedure, which is to guarantee that

the service of a network of buses connected through bridges is not interrupted by addition or removal of bridges. The relevant parts of the standard were specified in Promela and analysed using Spin, which revealed several shortcomings and one bug in the loop detection algorithm [27].

Other cases are the verification using μCRL of a distributed system for lifting trucks [17], an in-flight data acquisition unit for Lynx helicopters [15], and a cache coherence protocol for a Java Distributed Memory system [29].

Together with Lucent Technologies, the stability and control of an SDH data communication network, by measuring the performance of an experimental network at Lucent using both simulation and numerical methods. Although the main focus was on performance issues, as a bonus the measurements revealed system traces that do not adhere to the specification of the intended network behaviour.

5 Spin-off Projects

The SVC experience stimulated the start-up of a number of projects, guaranteeing the continuation of efforts initiated during SVC. At CWI, the PROGRESS projects "Improving the Quality of Embedded Systems by Formal Design and Systematic Testing" and "Formal Design, Tooling, and Prototype Implementation of a Real-Time Distributed Shared Data Space", and the NWO projects "Integrating Techniques for the Verification of Distributed Systems" and "Tools and Techniques for Integrating Performance Analysis and System Verification" build on the heritage of SVC. Among the spin-offs launched by the UT are the PROGRESS projects "Verification of Hard and Softly Timed Systems" and "Atom Splitting in Embedded Systems Testing", the NWO projects "Specification-Based Performability Checking" and "Systematic Testing of Real-Time Software Systems", and the NWO Vernieuwingsimpuls "Verification of Performance and Dependability".

Theses

1. D'Argenio, P.R., *Algebras and Automata for Timed and Stochastic Systems.* IPA Dissertation Series 1999-10, October 1999.

2. Ruys, Th.C., *Towards Effective Model Checking.* IPA Dissertation Series 2001-10, March 2001.

3. Usenko, Y.S., *Linearization in μCRL.* IPA Dissertation Series 2002-16, December 2002.

Selected Publications

4. Arts, Th. and Langevelde, I.A. van, How μCRL Supported a Smart Redesign of a Real-Life Protocol. In: *Proc. FMICS'99*, pp. 31 53, 1999.

5. Arts, Th. and Langevelde, I.A. van, Correct Performance of Transaction Capabilities. In: *Proc. ICACSD'01*, pp. 35–42, IEEE, 2001.

6. Baier, C., Katoen, J.P. and Hermanns, H., Approximate Symbolic Model Checking of Continuous-Time Markov Chains. In: *Proc. CONCUR'99*, LNCS 1664, pp. 146–162, Springer, 1999.

7. Baier, C., Katoen, J.P. and Hermanns, H., Model Checking Continuous-Time Markov Chains by Transient Analysis. In: *Proc. CAV'00*, LNCS 1855, pp. 358–372, Springer, 2000.

8. Blom, S.C.C., Fokkink, W.J., Groote, J.F., Langevelde, I.A. van, Lisser, B. and Pol, J.C. van de, μCRL: A Toolset for Analysing Algebraic Specifications. In: *Proc. CAV'01*, LNCS 2102, pp. 250–254, Springer, 2001.

9. Blom, S.C.C. and Orzan, S.M., A Distributed Algorithm for Strong Bisimulation Reduction of State Spaces. In: *Proc. PDMC'02*, ENTCS 68(4), Elsevier, 2002.

10. Blom, S.C.C. and Pol, J.C. van de, State Space Reduction by Proving Confluence. In: *Proc. CAV'02*, LNCS 2404, pp. 596–609, Springer, 2002.

11. Brinksma, E. and Groote, J.F., Validatietechnieken Houden Complexe Systemen Hanteerbaar. *Automatiseringgids* 19, 1999.

12. D'Argenio, P.R., Hermanns, H. and Katoen, J.P., On Generative Parallel Composition. ENTCS 22, Elsevier, 1999.

13. D'Argenio, P.R., Katoen, J.P. and Brinksma, E., Specification and Analysis of Soft Real-Time Systems: Quantity and Quality. In: *Proc. RTSS'99*, pp. 104 114, IEEE, 1999.

14. Dechering, P.F.G. and Langevelde, I.A. van, The Verification of Coordination. In: *Proc. COORDINATION'00*, LNCS 1906, pp. 335 340, Springer, 2000.

15. Fokkink, W.J., Ioustinova, N., Kesseler, E., Pol, J.C. van de, Usenko, Y.S. and Yushtein, Y., Refinement and Verification Applied to an In-Flight Data Acquisition Unit. In: *Proc. CONCUR'02*, LNCS 2421, pp. 1 23, Springer, 2002.

16. Groote, J.F. and Lisser, B. Computer Assisted Manipulation of Algebraic Process Specifications. In: *Proc. VCL'02*, Report DSSE-TR-2002-5, University of Southampton, 2002.

17. Groote, J.F., Pang, J. and Wouters, A.G., Analysis of a Distributed System for Lifting Trucks. *Journal of Logic and Algebraic Programming* 55(1/2), pp. 21–56, 2003.

18. Groote, J.F., Ponse, A. and Usenko, Y.S., Linearization of Parallel pCRL. *Journal of Logic and Algebraic Programming* 48(1/2), pp. 39–72, 2001.

19. Hermanns, H., *Interactive Markov Chains and the Quest for Quantified Quality.* LNCS 2428, Springer, 2002.

20. Hermanns, H., Katoen, J.P., Meyer-Kayser, J. and Siegle, M., A Markov Chain Model Checker. In: *Proc. TACAS'00*, LNCS 1785, pp. 347–362. Springer, 2000.

21. Hermanns, H., Herzog, U. and Katoen, J.P., Process algebra for performance evaluation. *Theoretical Computer Science* 274(1/2), pp. 43–87, 2002.

22. Hermanns, H., Herzog, U., Klehmet, U., Mertsiotkis, V., and Siegle, M., Compositional Performance Modelling with the TIPPtool. *Performance Evaluation* 39(1/4), pp. 5–35, 2000.

23. Hermanns, H. and Siegle, M., Bisimulation Algorithms for Stochastic Process Algebras and their BDD-based Implementation. In: *Proc. ARTS'99*, LNCS 1601, pp. 144–264, Springer 1999.

24. Hooman, J. and Pol, J.C. van de, Formal Verification of Replication on a Distributed Data Space Architecture. In: *Proc. SAC'02*, pp. 351–358, ACM, 2002.

25. Ioustinova, N., Sidorova, N. and Steffen, M., Abstraction and Flow Analysis for Model Checking Open Asynchronous Systems. In: *Proc. APSEC'02*, pp. 227–235, IEEE, 2002.

26. Langerak, R. and Brinksma, E., A Complete Finite Prefix for Process Algebra. In: *Proc. CAV'99*, LNCS 1633, pp. 184 195, Springer, 1999.

27. Langevelde, I.A. van, Romijn, J.M.T. and Goga, N., Founding FireWire Bridges through Promela Prototyping. In: *Proc. FMPPTA'03*, IEEE, 2003.

28. Orzan, S.M. and Pol, J.C. van de, Distribution of a Simple Shared Dataspace Architecture. In: *Proc. FOCLASA'02*, ENTCS 68(3), Elsevier, 2002.

29. Pang, J., Fokkink, W.J., Hofman, R. and Veldema, R., Model Checking a Cache Coherence Protocol for a Java DSM Implementation. In: *Proc. FMPPTA'03*, IEEE, 2003.

30. Pol, J.C. van de, Just-In-Time: On Strategy Annotations. In: *Proc. WRS'01*, ENTCS 57, Elsevier, 2001.

31. Pol, J.C. van de, JITty: A Rewriter with Strategy Annotations. In: *Proc. RTA'02*, LNCS 2378, pp. 367 -370, Springer, 2002.

32. Pol, J.C. van de and Valero Espada, M., Formal Specification of JavaSpaces Architecture using μCRL. In: *Proc. COORDINATION'02*, LNCS 2315, pp. 274–290, Springer, 2002.

33. Ruys, Th.C., Xspin/Project – Integrated Validation Management for Xspin, In: *Proc. SPIN'99*, LNCS 1680, pp. 108–119, Springer, 1999.

34. Usenko, Y.S., State Space Generation for the HAVi Leader Election Protocol. *Science of Computer Programming* 43(1), pp. 1–33, 2002.

35. Usenko, Y.S., Linearization of μCRL Specifications. In: *Proc. VCL'02*, Report DSSE-TR-2002-5, University of Southampton, 2002.

Approximate anytime inference: Half an answer on time is better than a perfect answer too late

Frank van Harmelen & Annette ten Teije

Department of Artificial Intelligence
Vrije Universiteit Amsterdam

Frank.van.Harmelen@cs.vu.nl
Annette@cs.vu.nl

1 Introduction

Many branches of computer science rely on logical formalisms as their foundations. Logic provides a very solid foundation on which to build new results: after 2500 years of research, logic provides us with a strong set of theoretical notions: model theory, proof theory, completeness, correctness; there exists a well established body of knowledge concerning properties such expressiveness, decidability, complexity results, etc; and, in particular since the very fruitful encounter with Computer Science in the last half century, we have at our disposal a rich battery of algorithms for implementing logical formalisms on a computer: resolution, tableau-reasoning, local-search methods, etc.

However, these strong foundations also come with a disadvantage: Logic aims to mode *perfect reasoners under ideal circumstances*. Logic does not tell us how to deal with mistakes during the deduction process; logical formalisms assume unlimited resources and fully available and fully correct knowledge. Logic does not study what it means to have "half a proof", or what it would mean for a theorem to be "almost true".

As a result of all this, the deduction models that logic provides us with are *crisp, abrupt, inefficient*:

- deduction algorithms are *crisp*, the answers computed by deduction are perfect, or they are not given at all; approximate answers are never computed.
- deduction algorithms are *abrupt*: they give answers at the very end of a computation, there is no intermediate "currently best available answer" during the computation.
- algorithms are *inefficient*: most interesting logics have high computational complexity.

However, in sharp contrast to this, many application areas demand that algorithms be instead *approximate, incremental and anytime*:

- *approximate* answers should be possible: answers that are "almost correct", with a precise notion of "almost".
- Answers should be computed *incrementally*. This would allow algorithms to return the "best available answers up until now".
- Algorithms should trade time against quality. Such *"anytime algorithms"* can be interrupted at any time to give the best answer available at that moment, with a guarantee that the quality of the answer will increase over the runtime of the algorithm.

The sketch in figure 1 captures the contrast between the behaviour of traditional deduction algorithms (figure 1a), and the behaviour demanded by many applications (figure 1b).

Figure 1: Traditional versus anytime behaviour

Approximate reasoning is a form of reasoning that can be used for solving complex problems. For approximate reasoning one defines a quality measure on the output, and the computation then tries to optimize this quality measure. This is in contrast with the conventional notion of a solution, where a solution is either correct or incorrect, with no middle ground. Optimising the quality measure does provide such a middle ground.

Approximate reasoning is interesting for several reasons. First of all, most AI-problems (tasks) are hard problems in term of their complexity measure. Planning, diagnosis and configuration are all examples of AI tasks for which even simple varieties are already intractable (e.g. [Bylander et al., 1991]). Therefore it is necessary to look for cheaper but approximate solutions instead of intractable precise solutions. Secondly, it depends on the particular application of the problem type (e.g. design, diagnosis) whether a precise solution is actually needed or whether an approximate solution suffices. For instance, in diagnostic reasoning there is not always a need for computing precise diagnoses, for example in cases where all possible diagnoses will result in the same repair action (e.g. when all fault-candidates are located on the same computer-board that must be replaced or in the medical domain when all possible diagnoses indicate the same drug to be prescribed) [van Harmelen and ten Teije, 1995]. As a third reason, it is often not possible in practice to have complete and correct data and knowledge. Examples are missing attribute values in classification, incomplete medical knowledge for performing diagnosis and incomplete requirements for design. So again, an approximate answer should be computed if possible, because an approximate answer is often better than no answer at all.

A particularly interesting form of approximate reasoning is *anytime reasoning* as introduced in [Dean and Boddy, 1988]. The most important characteristic of anytime reasoning is that with increasing runtime, the quality of the solution increases. Furthermore, the reasoning can be interrupted at any time and will return the best result computed until then. In this paper, we will concentrate on this type of reasoning.

Many types of reasoning in AI such as diagnosis, classification, design and planning are characterised in terms of logical entailment. A general approach for constructing approximate problem solving behaviour is to use an approximation of the logical entailment for characterising such a problem type and see what conclusions can be drawn using approximate reasoning for a particular problem.

This is exactly the approach that we have taken in some of our work (e.g. [ten Teije and van Harmelen, 1996, ten Teije and van Harmelen, 1997, Verberne et al., 2000], and which we will outline in this paper.

Below, we consider the approximate entailment relation proposed in [Schaerf and Cadoli, 1995] (section 2), and apply it to a simple deductive formalisation of diagnostic reasoning to investigate its effects (section 3).

2 Approximate deduction

In this section we will summarise the work in [Schaerf and Cadoli, 1995], which defines the approximate entailment relations that we will exploit for approximate diagnostic reasoning in the next section. Schaerf and Cadoli define two approximations of classical entailment, named $\vdash_1$ and $\vdash_3$ which are either unsound but complete ($\vdash_1$) or sound but incomplete ($\vdash_3$). By analogy, they sometimes write $\vdash_2$ for classical entailment. Both of these approximations are parameterised over a set of predicate letters S (written $\vdash_1^S$ and $\vdash_3^S$) which determines their accuracy. We repeat some of the basic definitions from [Schaerf and Cadoli, 1995]:

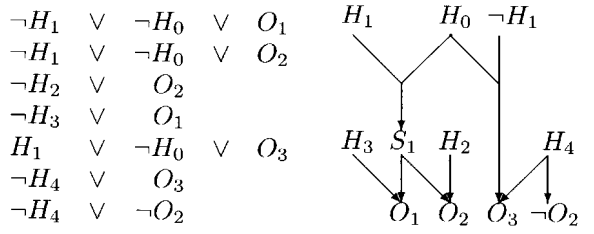

Figure 2: An example theory in clausal notation. In the formalisation of the network intermediate nodes (in this case only S_1) have been removed.

Definition 1 (1-S-assignment, 3-S-assignment)
A 1-S- and 3-S-assignment are defined as follows:
- *If $x \in S$ then x and $\neg x$ get opposite truth values*
- *If $x \notin S$ then*
  - *for a 1-S-assignment, x and $\neg x$ both become 0.*
  - *for a 3-S-assignment, x and $\neg x$ do* not *both become 0.*

In other words: for letters in S, these assignments behave as classical truth assignments, while for letters $x \notin S$ they make either all literals false (1-S-assignments) or make one or both of x and $\neg x$ true (3-S-assignments).

Satisfaction of a clause by a 1-S- or 3-S-assignment, and the notions of 1-S-entailment and 3-S-entailment are defined in the same way as classical satisfaction and entailment.

Intuitively, for 3-S-entailment the predicates outside S are deemed irrelevant for deduction, while for 1-S-entailment these predicates are taken as false. The following syntactic notions can be used to clarify these definitions. For a theory in clausal form, 1-S-entailment corresponds to classical entailment, but after removing from every clause any literals with a letter outside S. When this results in an empty clause, the theory becomes the inconsistent theory $\bot$. Similarly, 3-S-entailment corresponds to classical entailment, but after removing every clause from the theory that contains a literal with a letter outside S. This may result in the empty theory $\top$.

The main result of [Schaerf and Cadoli, 1995] is:

Theorem 1 (Approximate entailment)

$$\vdash_3^{\emptyset} \Rightarrow \vdash_3^{S} \Rightarrow \vdash_3^{S'} \Rightarrow \vdash_2 \Rightarrow \vdash_1^{S'} \Rightarrow \vdash_1^{S} \Rightarrow \vdash_1^{\emptyset}$$

where $S \subseteq S'$. (Everywhere primed letters are a superset of the unprimed letters).

This states that $\vdash_3^S$ is a sound but incomplete approximation of the classical $\vdash_2$. The counterpositive of the second half of the theorem (reading $\nvdash_1^S \Rightarrow \nvdash_1^{S'} \Rightarrow \nvdash_2$) states that $\nvdash_1^S$ is a sound but incomplete approximation of $\nvdash_2$.

Example 1 (Illustrating $\vdash_3^S$ and $\nvdash_1^S$)
We illustrate these notions with figure 2. We can see that $\vdash_3^S$ is incomplete with respect to $\vdash_2$, since in the theory BM of figure 2 we have that classically $BM \cup \{H_3\} \vdash_2 O_1$, but if we restrict S to LET$(BM) \setminus \{H_3\}$, where LET(BM) stands for all the predicate letters in BM, we do not have that $BM \cup \{H_3\} \vdash_3^S O_1$. This is so because taking $S =$ LET$(BM) \setminus \{H_3\}$ amounts to removing $H_3 \rightarrow O_1$ from BM.

Similarly, $\nvdash_1^S$ is incomplete with respect to $\nvdash_2$ (or equivalently, $\vdash_1^S$ is unsound w.r.t. $\vdash_2$) since, for example, if $S =$ LET$(BM) \setminus \{H_0\}$, then $BM \cup \{H_1\} \nvdash_2 O_1$, but $BM \cup \{H_1\} \vdash_1^S O_1$. This is so because taking $S =$ LET$(BM) \setminus \{H_0\}$ amounts to removing H_0 as a conjunct from $H_1 \wedge H_0 \rightarrow O_1$.

Furthermore, with increasing S, the accuracy of these approximations improves, until the approximate versions coincide with classical entailment when all letters are included in S.

Schaerf and Cadoli also give incremental algorithms for computing $\vdash_1^S$ and $\vdash_3^S$ when S increases. They have obtained attractive complexity results which state that even when computing $\vdash_2$ through iterative

computation of $\vdash_3^S$, the total cost of the iterated computation is not larger than the direct computation of $\vdash_2$ (and similarly for $\nvdash_1^S$ to compute $\nvdash_2$). However, the iterative computation of the approximate entailment has as important advantage that the iteration may be stopped when a confirming answer has already obtained for a smaller value of S. This yields a potentially drastic reduction of the computational costs. The size of these savings depend on the appropriate choice for S.

Although the summary above is based on a propositional calculus, [Schaerf and Cadoli, 1995] shows how the propositional results can be extended to the first-order case in a straightforward way.

3 Approximate diagnostic reasoning

In this section we summarize the results [ten Teije and van Harmelen, 1996] on applying $\vdash_1^S$ and $\vdash_3^S$ in diagnosis. We first give a common definition of diagnosis that is widespread in the literature, and then summarize the results [ten Teije and van Harmelen, 1996] on applying $\vdash_1^S$ and $\vdash_3^S$ in diagnosis.

3.1 Formalisation of diagnostic reasoning

We follow [Console and Torasso, 1991] and combine in our definition both abductive and consistency based diagnosis, which accounts for a large variety of diagnostic systems from the literature.

Definition 2 (Diagnosis problem and solution)
Given a behaviour model BM (a logical theory in clausal form), and two sets of observations O^+ and O^- (both sets of literals read as conjunctions), a solution to a diagnostic problem is a set of literals E ("E" for explanation, again read as a conjunction), which satisfies the following:

$$\text{ABD}: \quad BM \cup E \vdash O^+ \tag{1}$$

$$\text{ABD}: \quad BM \cup E \nvdash \bot \tag{2}$$

$$\text{CBD}: \quad BM \cup E \cup O^- \nvdash \bot \tag{3}$$

We will write OBS for the set of all possible observables from which the letters of O^+ and O^- must be taken, and require that E is disjoint from OBS. O^+ is the set of observations that must be explained abductively (i.e. they must be implied by the explanation E), while O^- only needs to be consistent with the explanation E.

Formulae (1) and (2) constitute the abductive part of our notion of diagnosis (ABD), and (3) the consistency based part (CBD). Although (2) directly follows from (3) for classical entailment we include both conditions explicitly, because the central idea of our method of approximations in diagnosis is to parameterise the notion of diagnosis over different approximations of the entailment relation (in particular of Schaerf & Cadoli's approximate entailment relations).

We emphasise that our particular definition of a diagnostic problem and its solution is not of crucial importance to our *central* message that approximate entailment can be usefully exploited for diagnostic reasoning to obtain interesting and efficient results.

3.2 Using approximate entailment in diagnosis

In this section we summarize the results [ten Teije and van Harmelen, 1996] on applying $\vdash_1^S$ and $\vdash_3^S$ in diagnosis. We use $\vdash_1^S$ and $\vdash_3^S$ in both the ABD-part of our definition of diagnosis (written as ABD_1^S, ABD_3^S) and the CBD-part (written as CBD_1^S, CBD_3^S). Since we write $\vdash_2$ for the classical entailment relation, we will also write ABD_2 and CBD_2.

The main intuitions behind using $\vdash_1^S$ and $\vdash_3^S$ in diagnosis are as follows. By using $\vdash_1^S$, candidate solutions more easily satisfy part (1) of our definition of diagnosis, because $\vdash_2 \Rightarrow \vdash_1^S$. Similarly, by using $\vdash_3^S$, candidate solutions more easily satisfy parts (2) and (3) of our definition of diagnosis, since $\nvdash_2 \Rightarrow \nvdash_3^S$.

We will write ABD_i^S when we intend both ABD_1^S and ABD_3^S, and similarly for CBD_i^S. Furthermore, we write $\mathcal{ABD}_i^S$ for the set of all diagnoses E which satisfy $\text{ABD}_i^S(E)$, and similarly for $\mathcal{ABD}_2$, $\mathcal{CBD}_i^S$ and $\mathcal{CBD}_2$

There are two important relations $\Subset$ and $\Supset$ for relating the ABD_i^S and CBD_i^S diagnoses.

diagnosis definition	change of S	new superset diagnosis	new subset diagnosis	nr.
ABD_1^S	$S \rightarrow S'$	yes	no	more
$\text{ABD}_1^{S'}$	$S' \rightarrow S$	no	only	less
ABD_3^S	$S \rightarrow S'$	no	yes	more
$\text{ABD}_3^{S'}$	$S' \rightarrow S$	only	no	less
CBD_1^S	$S \rightarrow S'$	only	no	more
$\text{CBD}_1^{S'}$	$S' \rightarrow S$	no	no	less
CBD_3^S	$S \rightarrow S'$	no	no	less
$\text{CBD}_3^{S'}$	$S' \rightarrow S$	no	only	more

Figure 3: Summarising some results of using approximate entailment in the diagnosis definition [ten Teije and van Harmelen, 1996]. *"yes"* means that using the new S results in superset/subset diagnoses, and similarly for *"no"*. *"only"* means that all the new computed diagnoses are superset/subset diagnoses. *more* and *less* means that the number of diagnoses increases and decreases respectively.

Definition 3 *For any set of sets P and P', $P \subseteq\!\!\!\!\rightarrow\ P'$ and $P \subseteq\!\!\!\!\leftarrow\ P'$ are defined by:*

$$P \subseteq\!\!\!\!\rightarrow\ P' \equiv \forall p \in P\ \exists p' \in P' : p \subseteq p'$$
$$P \subseteq\!\!\!\!\leftarrow\ P' \equiv \forall p' \in P'\ \exists p \in P : p \subseteq p'$$

Notice that these relations are relations between sets of sets. The required superset and subset relation is among the elements (sets) of these sets of sets.

For the set of abductive diagnoses we have the following relation:

Theorem 2 (Relations between $\mathcal{ABD}_i^S$)

$$\begin{aligned}\emptyset = \ & \mathcal{ABD}_1^{\emptyset} \subseteq \mathcal{ABD}_1^S \subseteq\!\!\!\!\rightarrow\ \mathcal{ABD}_1^{S'} \subseteq\!\!\!\!\rightarrow\ \mathcal{ABD}_2 \\ & \mathcal{ABD}_2 \subseteq\!\!\!\!\leftarrow\ \mathcal{ABD}_3^{S'} \subseteq\!\!\!\!\leftarrow\ \mathcal{ABD}_3^S \subseteq\!\!\!\!\leftarrow\ \mathcal{ABD}_3^{\emptyset} = \emptyset\end{aligned}$$

This states that ABD_1^S diagnoses consist of parts of ABD_2 diagnoses, and that ABD_3^S diagnoses contain ABD_2 diagnoses. Another result is on the number of diagnoses:

Theorem 3 (Sizes of $\mathcal{ABD}_i^S$)

$$\begin{aligned}0 = \ & |\mathcal{ABD}_1^{\emptyset}| \leq |\mathcal{ABD}_1^S| \leq |\mathcal{ABD}_1^{S'}| \leq |\mathcal{ABD}_2| \\ & |\mathcal{ABD}_2| \geq |\mathcal{ABD}_3^{S'}| \geq |\mathcal{ABD}_3^S| \geq |\mathcal{ABD}_3^{\emptyset}| = 0\end{aligned}$$

We have analogous results for the CBD-part:

Theorem 4 (Relations between CBD_i^S)

$$\begin{aligned}&\emptyset = \mathcal{CBD}_1^{\emptyset} \subset \mathcal{CBD}_1^S \Subset \mathcal{CBD}_2 \Subset \mathcal{CBD}_3^S \Subset \mathcal{CBD}_3^{\emptyset} = \mathcal{E} \\ &\emptyset = \mathcal{CBD}_1^{\emptyset} \subset \mathcal{CBD}_1^S \subset \mathcal{CBD}_2 \subset \mathcal{CBD}_3^S \subset \mathcal{CBD}_3^{\emptyset} = \mathcal{E} \\ &0 = |\mathcal{CBD}_1^{\emptyset}| < |\mathcal{CBD}_1^S| < |\mathcal{CBD}_2| < |\mathcal{CBD}_3^S| < |\mathcal{CBD}_3^{\emptyset}|\end{aligned}$$

where $\mathcal{E}$ stands for any consistent set of literals whose letters are taken from $\text{LET}(BM) \setminus \text{OBS}$. The first sequence of inclusions states that CBD_1^S diagnoses consist of parts of classical CBD-diagnoses, and that every classical diagnosis is contained in at least one CBD_3^S diagnosis. In [ten Teije and van Harmelen, 1996] we also have theorems about the type (superset or subset) of new diagnoses that can be found by changing S. *"New"* means that using the new value of S, we compute at least one superset/subset diagnosis which was not present for the old value of S. These theorems are summarized in Fig. 3.

4 Outlook

In the above, we have argued that there is a genuine need to investigate models of logical deduction that are approximate, incremental and anytime, as opposed to the traditional models which are crisp, abrupt, inefficient. We have shown an example of such an approximate, incremental and anytime model of deduction, and that they can usefully be exploited to formalise important computational patterns, such as those found in diagnostic reasoning.

Besides the model by Cadoli and Schaerf, other approximate, incremental and anytime deduction models exist. In [Verberne et al., 2000], we have investigated the use of a model developed by Dalal in [Dalal, 1996] that generalised Boolean Constraint Propagation. Essentially, this allows resolution steps on clauses of ever increasing length.

Besides diagnostic reasoning, many other important areas exist in which these ideas can be further developed. In particular, the widespread application of deduction techniques to the World Wide Web, aiming for something called the "Semantic Web" provides a fertile incubator and challenging testbed for many ideas in the same spirit as those outlined above. Our own early results in this area can be found in [Stuckenschmidt and van Harmelen, 2002] and [Stuckenschmidt, 2003], but many avenues remain unexplored as yet.

References

[Bylander et al., 1991] Bylander, T., Allemang, D., Tanner, M. C., and Josephson, J. R. (1991). The computational complexity of abduction. *Artificial Intelligence*, 49:25–60.

[Console and Torasso, 1991] Console, L. and Torasso, P. (1991). A spectrum of logical denitions of model-based diagnosis. *Computational Intelligence*, 7(21):133–141.

[Dalal, 1996] Dalal, M. (1996). Semantics of anytime family of reasoners. In *ECAI*, pages 360–364.

[Dean and Boddy, 1988] Dean, T. and Boddy, M. (1988). An analysis of time-dependent planning problems. In *Proceedings of the 7th National Conference on Artificial Intelligence*, volume 1, pages 49–54, San Mateo. Morgan Kaufmann.

[Schaerf and Cadoli, 1995] Schaerf, M. and Cadoli, M. (1995). Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249–310.

[Stuckenschmidt, 2003] Stuckenschmidt, H. (2003). Approximate information filtering with multiple classification hierarchies. *International Journal of Computational Intelligence Applications*.

[Stuckenschmidt and van Harmelen, 2002] Stuckenschmidt, H. and van Harmelen, F. (2002). Approximating terminological queries. In et al., H. L., editor, *Proceedings of the Proceedings of the 4th International Conference on Flexible Query Answering Systems (FQAS)'02)*, Advances in Soft Computing. Springer-Verlag.

[ten Teije and van Harmelen, 1996] ten Teije, A. and van Harmelen, F. (1996). Computing approximate diagnoses by using approximate entailment. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 265–256, Boston, Massachusetts.

[ten Teije and van Harmelen, 1997] ten Teije, A. and van Harmelen, F. (1997). Exploiting domain knowledge for approximate diagnosis. In Pollack, M., editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*, pages 454–459, Nagoya, Japan.

[van Harmelen and ten Teije, 1995] van Harmelen, F. and ten Teije, A. (1995). Approximations in diagnosis: motivations and techniques. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation, (SARA'95)*, Ville d'Esterl, Canada.

[Verberne et al., 2000] Verberne, A., van Harmelen, F., and ten Teije, A. (2000). Anytime diagnostic reasoning using approximate boolean constraint propagation. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, Boulder, Colorado.