Nieuwsbrief van de Nederlandse Vereniging voor Theoretische Informatica

Joke Lammerink Joost-Pieter Katoen Joost Kok Jaco van de Pol
Femke van Raamsdonk

Inhoudsopgave

Van de redactie

Geachte NVTI-leden!

De NVTI nieuwsbrief van 2009 ligt voor u, of heeft u in de hand. Zoals gebruikelijk vindt u hierin de aankondiging voor de NVTI theoriedag, die dit jaar op vrijdag 20 maart in Utrecht plaatsvindt. Sprekers dit jaar zijn: mede-Turing award winnaar in 2008 E. Allen Emerson (University of Texas at Austin), Barbara Terhal (IBM Watson, NY), Frank de Boer (CWI, UL) en Paul Vitanyi (CWI, UvA). Kortom: meer dan voldoende reden om de stoffige buro's te verlaten en naar Utrecht te komen.

De wetenschappelijke bijdragen dit jaar zijn van Djoerd Hiemstra (UT) over probabilistische modellen voor information retrieval — mocht u zich afvragen hoe u de Google's page rank score van uw webpagina kan verhogen, dan is dit een aanrader— en van Jörg Endrullis (VU), Clemens Grabmayer (UU), Dimitri Hendriks (VU), en Jan Willem Klop (VU) over oneindige stromen en, onder andere, hun grafische voorstelling. De bijdrage van Monique Laurent (CWI) gaat over approximatieve algoritmen voor lastige combinatorische optimalisatie problemen en Freek Wiedijk (RUN) bericht over het ProofWeb systeem. Daarnaast is er een samenvatting van de belangrijkste activiteiten van de onderzoekscholen IPA en SIKS in het afgelopen jaar. Alle auteurs hartelijk dank voor hun bijdrage!

Tenslotte danken wij het CWI, NWO, Elsevier, IPA, SIKS en OzsL voor de sponsoring van de NVTI activiteiten, en Susanne van Dam voor de secretariële ondersteuning.

Namens de redactie van de NVTI,
Joost-Pieter Katoen

Samenstelling bestuur

Van de voorzitter

De nieuwsbrief, de theoriedag en de mailing list zijn de drie pijlers van de NVTI.
De theoriedag staat bekend om zijn goede sprekers (zie
http://www.nvti.nl/speakerslist.html voor een overzicht sinds 1995) en ook dit jaar ziet het programma er weer geweldig uit met E. Allen Emerson, Barbara Terhal, Frank de Boer en Paul Vitanyi. De nieuwsbrief heeft altijd een interessante wetenschappelijke inhoud en bevat ook de aankondiging van de theoriedag.

Achter de schermen van de NVTI wordt veel werk verzet. Een flink aantal mensen die de theorie een warm hart toe dragen spannen zich voor de vereniging in en ik wil graag deze mensen bedanken voor hun inzet voor de vereniging in de afgelopen jaren.

Dit jaar zijn er een aantal veranderingen: de nieuwsbrief krijgt een nieuwe redacteur in de persoon van Marielle Stoelinga die het stokje overneemt van
Joost-Pieter Katoen. Ook bestuurlijk zijn er veranderingen gepland, die echter nog wel goedkeuring van de ledenvergadering op de theoriedag behoeven. Han La Poutré, Karin Aardal en Wim Hesselink zijn bereid om tot het bestuur toe te treden. De rolverdeling zal ook veranderen; Jaco van der Pol zal de nieuwe voorzitter worden en Han La Poutré de nieuwe secretaris van het bestuur. Met al deze nieuwe mensen
aan boord ziet de toekomst van de NVTI er rooskleurig uit!

Ik kijk al uit naar de theoriedag 2009 op 20 maart en ik hoop dat u ook in staat bent om te komen!

Joost N. Kok,
Voorzitter NVTI

Nederlandse Vereniging voor Theoretische Informatica

We are happy to invite you for the Theory Day 2009 of the NVTI. The Dutch Association for Theoretical Computer Science (NVTI) supports the study of theoretical computer science and its applications.

NVTI Theory Day 2009
Friday March 20, 2009, 9:30-16:45
Hoog Brabant, Utrecht (close to Central Station)

We have an interesting program with excellent speakers from The Netherlands and abroad, covering important streams in theoretical computer science. Below you will find the abstracts.

Lecturers:

Barbara Terhal (IBM Watson, NY)
Frank de Boer (CWI, U Leiden)
Mark Kas (NWO Physical Sciences)
E. Allen Emerson (U Texas, Austin)
Paul Vitanyi (CWI, U v Amsterdam)

It is possible to participate in the organized lunch, for which registration is required. Please register by E-mail (J.M.W.Lammerink@ewi.utwente.nl) or
by phone (053-4893767), no later than one week before the meeting (March 13, 2009). The costs of 15 Euro can be paid at the location. We just mention that in the direct vicinity of the meeting room there are plenty of nice lunch facilities as well.

The NVTI theory day 2009 is sponsored (financially or in kind) by
NWO (Netherlands Organisation for Scientific Research), Elseviers Science,
CWI (Dutch Center of Mathematics and Computer Science) and the Dutch research schools IPA (Institute for Programming Research and Algorithmics), OzsL (Dutch Graduate school in Logic) and SIKS (Dutch research school for Information and Knowledge Systems).

Please find the full program and abstracts of the lectures below.

Kind regards,

Jaco van de Pol,
NVTI secretary.

Program

9.30-10.00: Arrival with coffee

10.00-10.10: Opening

10.10-11.00: Lecture: Barbara Terhal (IBM Watson, NY)
Title: *Quantum Complexity Theory:*
Bringing Rigor to Computational Quantum Physics

11.00-11.30: Coffee/Tea

11.30-12.20: Lecture: Frank de Boer (CWI, U Leiden)
Title: *Tasks for Actors*

12.20-12.40: Dr. Mark Kas (NWO Physical Sciences)
Title: *Around the world (of Dutch computer science) in 8 years and 80 million euros*

12.40-14.10: Lunch (see above for registration)

14.10-15.00: Lecture: E. Allen Emerson (U Texas, Austin)
Title: *Model Checking: Theory, Themes, and Practice*

15.00-15.20: Coffee/Tea

15.20-16.10: Lecture:Paul Vitanyi (CWI, U v Amsterdam)
Title: *Universal Similarity*

16.10-16.40: Business meeting NVTI

Abstracts

Speaker: Barbara Terhal (IBM Watson, NY)
Title: *Quantum Complexity Theory: Bringing Rigor to Computational Quantum Physics*

In the past century physicists have developed many heuristic computational techniques for understanding and analyzing quantum physical systems. While these techniques often work very well in practice, a rigorous understanding of the efficiency and limitations of these methods, that is, a quantum complexity theory, has been lacking. We will discuss several results in the emerging field of quantum complexity theory which address the complexity of problems in quantum physics. In particular we will consider problems which are QMA-complete or "quantum NP-complete".

Speaker: Frank de Boer (CWI, U Leiden)
Title: *Tasks for Actors*

This lecture presents a modular method for a schedulability analysis of real time distributed systems. Distributed systems are naturally described in terms of actors, which constitute an asynchronous model for concurrent objects. An extension of the actor model is presented with real time and behavioral interfaces for specifying the order and timings of the messages an actor may send and receive. Scheduling policies additionally determine the order in which the received messages are processed.
It is shown how to analyze the behavioral interface of an actor to check that every received message is processed within the specified deadline (schedulability analysis). The overall modular analysis of a system of actors is based on a technique for testing the compatibility of the actual use of an actor and its behavioral interface.

Speaker: E. Allen Emerson (U Texas, Austin)
Title: *Model Checking: Theory, Themes, and Practice*

Model checking is an automatic method of verifying finite state concurrent programs. The use of temporal logic and related frameworks to specify correctness has greatly facilitated simply thinking about the verification problem. Despite early worries about the intractability of state explosion, nowadays it can often be ameliorated, permitting verification of enormously large systems in practice. We will discuss various themes underlying the utility of model checking including expressive specification, efficient reasoning, and simplification.

Speaker: Paul Vitanyi (CWI, U v Amsterdam)
Title: *Universal Similarity*

We survey a new area of parameter-free similarity distance measures useful in data-mining, pattern recognition, learning and automatic semantics extraction. Given a family of distances on a set of objects, a distance is universal up to a certain precision for that family if it minorizes every distance in the family between every two objects in the set, up to the stated precision (we do not require the universal distance to be an element of the family). We consider similarity distances for two types of objects: literal objects that as such contain all of their meaning, like genomes or books, and names for objects. The latter may have literal embodyments like the first type, but may also be abstract like "red" or "christianity." For the first type we consider a family of computable distance measures corresponding to parameters expressing similarity according to particular features between pairs of literal objects. For the second type we consider similarity distances generated by web users corresponding to particular semantic relations between the (names for) the designated objects. For both families we give universal similarity distance measures, incorporating all particular distance measures in the family. In the first case the universal distance is based on compression and in the second case it is based on Google page counts related to search terms. In both cases experiments on a massive scale give evidence of the viability of the approaches.

www.win.tue.nl/ipa/

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 & Engineering. Researchers from nine universities (University of Nijmegen, Leiden University, Technische Universiteit Eindhoven, University of Twente, Utrecht University, University of Groningen, Vrije Universiteit Amsterdam, University of Amsterdam, and Delft University), the CWI and Philips Research (Eindhoven) participate in IPA.

In 1997, IPA was formally accredited by the Royal Dutch Academy of Sciences (KNAW). This accreditation was extended in 2002 and 2007. In setting its agenda for 2007 - 2012, IPA chose five focus areas, where we expect important developments in the near future and want to stimulate collaboration. In the focus area:

Beyond Turing we want to explore novel paradigms of computation that incorporate concepts that are no longer adequately modeled by the classical Turing machine such as nonuniformity of memory, adaptivity and mobility.

Algorithms & models for life sciences we wish to apply algorithmic theory and formal models to contribute to the understanding of biological processes, entities and phenomena.

Hybrid systems we want to continue to contribute to the confluence of systems and control theory and computer science in integrated methods for modelling, simulation, analysis, and design of such systems.

Model-driven software engineering we want to study various fundamental aspects of the model-driven approach to software engineering.

Software analysis we want to make progress in the extraction of facts from source code and their analysis, to obtain instruments for measuring the various quality attributes of software.

For descriptions of these areas see `www.win.tue.nl/ipa/about.html`.

Activities in 2008

IPA has two multi-day events per year which focus on current topics, the Lentedagen and the Herfstdagen. In 2008, the Lentedagen were on Integrating Formal Methods and the Herfstdagen were dedicated to the focus area Software Analysis.

Lentedagen on Integrating Formal Methods *May 7 - 9, Hotel 't Paviljoen, Rhenen*
The IPA Lentedagen are an annual multi-day event, dedicated to a specific theme of current interest to the research community of IPA. This year's Lentedagen are dedicated to the integration of formal methods.

Formal methods (languages, tools, and techniques for the analysis and design of systems, with a sound, mathematical basis) come in many different flavors. One reason why this abundance of approaches persists is that there is still no single method that does it all: analyzing a complex or large system can require the application of different formal methods, modeling different relevant aspects of the system in isolation. For instance, algebraic data types for its data, process algebras or temporal logics for its behavior, duration calculus or timed automata for its timing. In the efforts to address this problem, two directions can be distinguished: one is to try to increase the scope of individual methods, extending them to cover more aspects within a single formal method. The other is to try to couple methods to benefit from their combined coverage. In this year's Lentedagen we focus on the second direction, the coupling of formal methods.

Methods (or "notations") can be coupled with different degrees of tightness, in their Ten Commandments Revisited, Bowen and Hinchey distinguish three levels:

- Viewpoints in a loose coupling: different notations present different system views, with each notation emphasizing a particular system aspect.
- Method integration in a closer coupling: several notations (both formal and informal or semiformal) combine with manual or automatic translation between notations. The idea is to provide an underlying semantics for the less formal notations, to enable well-understood graphical (or other) presentations, and to offer the benefits of formal verification.
- Integrated methods in a tight coupling: multiple notations combine within a single framework to give a uniform semantics to each notation.

Looking back over the past decade, the authors conclude that ground has only been gained on the first level in the practice of software engineering (i.e., the UML), but they acknowledge that progress has been made in research on the other levels. Within the IPA community both method integration and integrated methods are topics of considerable interest. Combining these under the heading of "Integrating Formal Methods", the Lentedagen aim to provide an overview of the work that is being done in and around IPA, addressing the integration of formal methods both at the level of formalisms and of the accompanying tools.

Einar Broch Johnsen (University of Oslo) and Hubert Garavel (INRIA Rhone-Alpes) were invited speakers. Abstracts, hand-outs and papers are available through the archive on the IPA-website:
`www.win.tue.nl/ipa/archive/springdays2008/`.

Herfstdagen on Software Analysis *November 24 - 28, Hotel Dennenhoeve, Nunspeet*
For the period 2007-2012, IPA has chosen five focus areas where it expects important developments in the field in the near future. Each year the Herfstdagen will be dedicated to one of these areas. This year's Herfstdagen are dedicated to Software Analysis.

Quality software can be characterized in many ways. Of course, software should first and foremost satisfy its functional requirements, but there are many non-functional quality aspects as well. For instance, availability, modifiability, performance, security, testability and usability are also important qualities of software. For each of these it is desirable to have analysis and measurement instruments to determine to what extent a software system satisfies them. Analysis starts with extracting relevant facts from the source code. This is a detailed step that is largely dependent on the programming languages that have been used during the construction of the source code and of the forms of analysis that are required. It is not uncommon that several languages have been used and in those cases cross-language fact extraction and analysis are necessary. Language-parametric techniques for fact extraction are highly desirable but mostly still lacking.

After fact extraction, direct analysis of these facts can be done for certain applications such as call graph analysis, dead code removal, analysis of information flows and the like. For other applications a more abstract model has to be constructed that describes certain domain-specific properties of the system that is implemented by the source code. In this area fall, for instance, protocol analysis, deadlock analysis and determining certain security properties. It is crucial to guarantee that the abstract model faithfully represents relevant properties of the original source code. Achieving scalability and usability of the involved validation techniques is a major challenge.

A final aspect to consider is the way in which the results of software analysis are presented. It is important to develop new methods for information visualization that will increase the usability of software analysis.

Software analysis is essential for getting insight in the quality aspects of software. This is true for old software that has evolved over the years, but also for new software that has to be accepted from a third party. Software analysis will therefore become more and more important as the outsourcing of software becomes more popular as well. It is also of particular importance in the area of open source software, where there is typically not a single producer that can be held responsible, but (by the very nature of the development process) the source code itself is available, providing extensive opportunities for analysis. Software analysis may reveal certain defects in software that have to be repaired. It is therefore also a prerequisite for refactoring, transformation, renovation and other forms of software improvement. The seamless integration of software analysis with software transformation is a rich area for research.

We feel that making progress in this area is both urgent and timely for several reasons: It is urgent because the ever increasing dependence of our society on software systems and the high-profile failures of some of those systems makes it mandatory to invest in techniques that can measure various quality attributes of software in order to prevent such failures. It is timely because progress in source-code analysis techniques enables the extraction of facts for more advanced analysis and validation than is currently done, and because improvements of validation techniques by way of algorithmic enhancements or concurrent execution enable the analysis of increasingly large problems.

This year's Herfstdagen aim to bring researchers from in and around IPA together, for an overview of current research in Software Analysis. The program for the event was composed by Arie van Deursen (TUD) and Paul Klint (UvA, CWI). The contributions to the program were clustered in the following sessions: Software Analysis, Repository Analysis, Co-evolution, Analysis for refactoring, Visual analysis, Model extraction, Dynamic analysis, Fault diagnosis, Security, Refactoring with formal methods, Static analysis via code query technologies, Language parametric analysis and transformation, and Type-related analysis.

Invited speakers were Ralf Lämmel (Universität Koblenz-Landau) and Rainer Koschke (Universität Bremen). Abstracts, hand-outs, and papers are available through the IPA website at `www.win.tue.nl/ipa/archive/falldays2008/`.

IPA organises Courses on each of its major research fields, Algorithms and Complexity, Formal Methods and Software Technology & Engineering. These courses intend to give an overview of the research of IPA in these fields, and are organized at regular intervals on a cyclic schedule. In 2008, the course on Algorithms and Complexity was held.

IPA Course on Formal Methods *June 23 - 27, TU/e, Eindhoven*
IPA organises courses on each of its major research fields, Algorithms and Complexity, Formal Methods and Software Technology. These three IPA Courses intend to give an overview of (part of) the research of IPA in this specific field. They are organised in a cycle of two years, and form a mandatory element in the education plan of IPA PhD-students (every student should attend at least two out of the three IPA courses).

The Formal Methods Course, which is hosted by IPA at the Technische Universiteit Eindhoven focussed on five main areas of Formal Methods research. One course day has been dedicated to each of these areas, in which lectures are combined with hands-on tool training. The program for this edition of the course was composed by Jaco van de Pol (UT). Topics and teachers were: *Timed*

Automata & UPPAAL, Frits Vaandrager (RU), *Model Checking with SPIN*, Theo C. Ruys (UT), *Theorem Proving & PVS*, Erik Poll (RU), *Behavioural Analysis using mCRL2*, Aad Mathijssen, Bas Ploeger, Tim Willemse, Frank Stappers (TU/e), and *Graph Transformations & GROOVE*, Arend Rensink (UT).

IPA Ph.D. Defenses in 2008

W. Pieters (RU, January 21)
La Volonté Machinale: Understanding the Electronic Voting Controversy
Promotor: prof.dr. B.P.F. Jacobs. Co-promotores: dr.ir. E. Poll, dr. M.J. Becker
IPA Dissertation Series 2008-01

M. Bravenboer (UU, January 21)
Exercises in Free Syntax: Syntax Definition, Parsing, and Assimilation of Language Conglomerates
Promotor: prof.dr. D.S. Swierstra. Co-promotor: dr. E. Visser
IPA Dissertation Series 2008-06

A.M. Marin (TUD, January 25)
An Integrated System to Manage Crosscutting Concerns in Source Code
Promotor: prof.dr. A. van Deursen
IPA Dissertation Series 2008-04

N.C.W.M. Braspenning (TU/e, February 18)
Model-based Integration and Testing of High-tech Multi-disciplinary Systems
Promotores: prof.dr.ir. J.E. Rooda, prof.dr. J.C.M. Baeten. Co-promotor: dr.ir. J.M. van de Mortel-Fronczak
IPA Dissertation Series 2008-05

M. Torabi Dashti (VUA, February 27)
Keeping Fairness Alive: Design and Formal Verification of Optimistic Fair Exchange Protocols
Promotores: prof.dr. W.J. Fokkink, prof.dr. J.C. van de Pol
IPA Dissertation Series 2008-07

A.L. de Groot (RU, March 6)
Practical Automaton Proofs in PVS
Promotor: prof.dr. F.V. Vaandrager. Co-promotor: dr. J. Hooman
IPA Dissertation Series 2008-02

I.S. Zapreev (UT, March 7)
Model Checking Markov Chains: Techniques and Tools
Promotores: prof.dr.ir. J.P. Katoen, prof.dr. H. Brinksma
IPA Dissertation Series 2008-11

I. Hasuo (RU, March 10)
Tracing Anonymity with Coalgebras
Promotor: prof.dr. B.P.F. Jacobs
IPA Dissertation Series 2008-09

G. Güleşir (UT, March 13)
Evolvable Behavior Specifications Using Context-Sensitive Wildcards
Promotor: prof.dr.ir. M. Akşit. Co-promotor: dr.ir. L.M.J. Bergmans
IPA Dissertation Series 2008-13

M. Bruntink (TUD, March 17)
Renovation of Idiomatic Crosscutting Concerns in Embedded Systems
Promotores: prof.dr. A. van Deursen, prof.dr. P. Klint
IPA Dissertation Series 2008-03

I.S.M. de Jong (TU/e, March 27)
Integration and Test Strategies for Complex Manufacturing Machines

Promotor: prof.dr.ir. J.E. Rooda. Co-promotor: dr.ir. J.M. van de Mortel-Fronczak
IPA Dissertation Series 2008-08

L.G.W.A. Cleophas (TU/e, April 1)
Tree Algorithms: Two Taxonomies and a Toolkit
Promotores: prof.dr. M.G.J. van den Brand, prof.dr. B.W. Watson. Co-promotor: dr.ir. C. Hemerik
IPA Dissertation Series 2008-10

M. Farshi (TU/e, April 8)
A Theoretical and Experimental Study of Geometric Networks
Promotor: prof.dr. M.T. de Berg. Co-promotor: dr. J. Gudmundsson
IPA Dissertation Series 2008-12

F.D. Garcia (RU, May 9)
Formal and Computational Cryptography: Protocols, Hashes and Commitments
Promotor: prof.dr. B.P.F. Jacobs. Co-promotor: dr. J.-J. Hoepman
IPA Dissertation Series 2008-14

P.E.A. Dürr (UT, June 26)
Resource-based Verification for Robust Composition of Aspects
Promotor: prof.dr.ir. M. Akşit. Co-promotor: dr.ir. L.M.J. Bergmans
IPA Dissertation Series 2008-15

E.M. Bortnik (TU/e, July 1)
Formal Methods in Support of SMC Design
Promotores: prof.dr.ir. J.E. Rooda, prof.dr. J.C.M. Baeten
IPA Dissertation Series 2008-16

C.M. Gray (TU/e, August 25)
Algorithms for Fat Objects: Decompositions and Applications
Promotor: prof.dr. M.T. de Berg
IPA Dissertation Series 2008-19

M. van der Horst (TU/e, September 4)
Scalable Block Processing Algorithms
Promotor: prof.dr.ir. C.H. van Berkel. Co-promotor: prof.dr. J.J. Lukkien
IPA Dissertation Series 2008-18

J.R. Calamé (UT, September 4)
Testing Reactive Systems with Data - Enumerative Methods and Constraint Solving
Promotores: prof.dr. J.C. van de Pol, prof.dr. W.J. Fokkink
IPA Dissertation Series 2008-20

E. Mumford (TU/e, September 8)
Drawing Graphs for Cartographic Applications
Promotor: prof.dr. M.T. de Berg. Co-promotor: dr. B. Speckmann
IPA Dissertation Series 2008-21

R.H. Mak (TU/e, September 9)
Design and Performance Analysis of Data-Independent Stream Processing Systems
Promotores: prof.dr. P.A.J. Hilbers, prof.dr.ir. C.H. van Berkel
IPA Dissertation Series 2008-17

A. Koprowski (TU/e, September 25)
Termination of Rewriting and Its Certification
Promotores: prof.dr. H. Zantema, prof.dr.ir. J.F. Groote
IPA Dissertation Series 2008-24

U. Khadim (TU/e, September 29)
Process Algebras for Hybrid Systems: Comparison and Development
Promotor: prof.dr. J.C.M. Baeten. Co-promotor: dr.ir. P.J.L. Cuijpers
IPA Dissertation Series 2008-25

J. Markovski (TU/e, October 2)
Real and Stochastic Time in Process Algebras for Performance Evaluation
Promotor: prof.dr. J.C.M. Baeten. Co-promotor: dr. E.P. de Vink
IPA Dissertation Series 2008-26

H. Kastenberg (UT, October 3)
Graph-Based Software Specification and Verification
Promotor: prof.dr. H. Brinksma. Co-promotor: prof.dr.ir. M. Akşit
IPA Dissertation Series 2008-27

I.R. Buhan (UT, October 23)
Cryptographic Keys from Noisy Data Theory and Applications
Promotor: prof.dr. P.H. Hartel. Co-promotor: dr.ir. R.N.J. Veldhuis
IPA Dissertation Series 2008-28

E.H. de Graaf (UL, October 29)
Mining Semi-structured Data, Theoretical and Experimental Aspects of Pattern Evaluation
Promotor: prof.dr. J.N. Kok. Co-promotor: dr. W.A. Kosters
IPA Dissertation Series 2008-22

R.S. Marin-Perianu (UT, November 6)
Wireless Sensor Networks in Motion: Clustering Algorithms for Service Discovery and Provisioning
Promotor: prof.dr. P.H. Hartel
IPA Dissertation Series 2008-29

R. Brijder (UL, December 3)
Models of Natural Computation: Gene Assembly and Membrane Systems
Promotor: prof.dr. G. Rozenberg. Co-promotor: dr. H.J. Hoogeboom
IPA Dissertation Series 2008-23

Activities in 2009

IPA is planning several activities for 2009, including the Lentedagen (April 15-17) which is organized in combination with ASCI and which will be dedicated to *Algorithms for data analysis and visualization*, the Course on Software Technology & Engineering (TU/e, Eindhoven), and the Herfstdagen (November) which will be dedicated to the focus area *Hybrid Systems* of IPA. More information on these events will appear on the IPA-website as dates and locations for these events are confirmed.

Addresses

Visiting address
Technische Universiteit Eindhoven
Main Building HG 7.22
Den Dolech 2
5612 AZ Eindhoven
The Netherlands

Postal address
IPA, Fac. of Math. and Comp. Sci.
Technische Universiteit Eindhoven
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 `www.win.tue.nl/ipa/`

School for Information and Knowledge Systems (SIKS) in 2008

Richard Starmans (UU)

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 & Information Systems and Software Engineering. Its main concern is research and education in the field of information and computing sciences, more particular in the area of information and knowledge systems.The School currently concentrates on seven focus areas in the IKS field: Agent Technology, Computational Intelligence, Knowledge Representation and Reasoning, Web-based Information Systems, Enterprise Information Systems, Human Computer Interaction, and Data Management, Storage and Retrieval.

SIKS is an interuniversity research school that comprises 12 research groups from 10 universities and CWI. Currently, over 400 researchers are active, including 200 Ph.D.-students. The Vrije Universiteit in Amsterdam is SIKS' administrative university, and as off January 1 2006 Prof.dr. R.J. Wieringa (UT) was appointed scientific director. The office of SIKS is located at Utrecht University. SIKS received its first accreditation by KNAW in 1998 and was re-accredited in June 2003 for another period of 6 years. In December 2008 SIKS has applied for another accreditation period of six years.

Activities

We here list the main activities (co-)organized or (co-)financed by SIKS. We distinguish basic courses, advanced courses and other activities (including master classes, workshops, one-day seminars, conferences, summer schools, doctoral consortia and research colloquia)

Basic courses:

"Interactive Systems", May 19 - 20, 2008, NH Hotel, Best
Course director: Prof.dr. P. De Bra (TU/e), Prof.dr. G. van der Veer (VUA/OU)

"Combinatory Methods", May 21-22, 2008, NH Hotel, Best
Course director: Dr. N. Roos (UM)

"Research methods and methodology for IKS", 24-25-26 November, 2008, Zeist
Course directors: Dr. H. Weigand (UvT), Prof.dr. R. Wieringa (UT), Prof.dr. H. Akkermans (VUA) Prof.dr. J.-J.Ch. Meyer (UU), Dr. R.J.C.M. Starmans (UU).

"Mathematical methods for IKS", 08-09 December, 2008, Landgoed Huize Bergen, Vught
Course directors: Prof.dr. E.O. Postma (UvT), Prof.dr. T. Heskes (RUN)

"Knowledge modeling", 10-11 December 2008, Landgoed Huize Bergen, Vught
Course director: Dr. B. Bredeweg (UVA)

Advanced courses:

"Engineering Web-based Systems: a semantic perspective", March 03-04 2008, Landgoed Huize Bergen, Vught.
Course directors: Prof.dr. G.-J. Houben (TUD), Dr. S. Schlobach (VUA)

"Summer course on Datamining", 25-29 August 2008, Maastricht
Course directors: Dr. E. Smirnov (UM), Prof.dr. E.O. Postma (UM)

"Computational Intelligence", October 23 en 24, 2008, Woudschoten, Zeist
Course directors: Prof.dr. A.P.J.M. Siebes (UU), Dr. U. Kaymak (EUR), Dr. A. Feelders (UU)

"Business Process Management",November 06-07, 2008, Landgoed Huize Bergen, Vught
Course directors: Prof.dr.ir. W. van der Aalst (TU/e), Prof.dr. M. Reichert (Univ. Ulm)

"Multi Agent Systems: Theory, Technology and Applications", December 11-12, 2008, Delft
Course directors: Prof. dr. B. De Schutter (TUD), Prof. dr. C. Witteveen (TUD).

Other activities:

- Conference: Benelearn 08, May 19-20, 2008, Spa (Belgium)
- Conference: BNAIC 08, October 30-31, 2008, Enschede
- Conference: Dutch-Belgian Database Day 2008 (DBDBD), October 2008, Namur (Belgium)
- Conference: DIR 2008, April 14-15, 2008, Maastricht
- Conference: "SIKS-Conference on Enterprise Information Systems" (EIS), May 22-23, Tilburg
- Conference: "First International Conference on Human-Robot Personal Relationships", 12-13 June, Maastricht
- Masterclass: "Ontology-Driven conceptual modeling with applications", March 04-05, 2008, Enschede
- Masterclass: "SIKS-ICS Master class Distributed model based diagnosis and repair", 07 Oktober, Utrecht, 25 June, 2008
- NVTI Theory Day 2008, March 14, 2008, Utrecht
- SIKS Evaluation Days 2008, 02-03 June, Amsterdam
- SIKS-day 2008, October 02, 2008, Utrecht
- Seminar: "The Semantic Web", 09 April 2008, Utrecht
- Seminar: "SIKS/Yahoo Seminar: Searching and Ranking in Structured Text Repositories", 27 June, 2008, Enschede
- Seminar: UU-SIKS Seminars (2 times in 2008)
- SIKS-Agent Colloquia (10 times in 2008), Utrecht/Delft/Amsterdam
- SIKS-IKAT Colloquia (8 times), Maastricht
- Summerschool EASSS 2008, May, 05-09, 2008, Lisbon (Portugal)
- Symposium: The Dynamics of Knowledge and Interpretation, April 22, 2008, Nijmegen
- Symposium: "SIKS-MICC Symposium on Embodied Cognition", 25 June, 2008 Maastricht
- Symposium: "CAI-SIKS Symposium New Directions in Evolutionary Computation", 27 November 2008, Nijmegen
- Symposium: "RU-SIKS Symposium:Trends in Artificial Intelligence", 28 November 2008, Nijmegen
- Workshop: "Agent Based approaches in Economics", January 24, 2008, Rotterdam.
- Workshop: "Machine Learning and Multimodal Interaction", 8-10 September 2008 Utrecht
- Workshop: "ERIM-SIKS Workshop on Support Vector Machines and Classification",12 September 2008 Rotterdam
- Workshop "Image Processing for Artist Identification II", 20-21 Oktober, 2008, Amsterdam
- Workshop TiCC-SIKS Workshop on "Single Neuron Modeling", 02 December 2008, Tilburg
- Working Conference on Human Factors and Computational Models in Negotiation, 08-09 December,2009, Delft

Ph.D.-defenses in 2008

In 2008 35 researchers successfully defended their Ph.D.-thesis and published their work in the SIKS-dissertation Series.

2008-01
Katalin Boer-Sorbán (EUR)
Agent-Based Simulation of Financial Markets: A modular, continuous-time approach
Promotor: Prof. dr. A. de Bruin (EUR)
Copromotor: Dr. ir. U. Kaymak (EUR)
Promotie: 25 January 2008

2008-02
Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations
Promotor: Prof.dr. J. Treur (VU)
Promotie: 10 January 2008

2008-03
Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach
Promotor: Prof.dr. B.J. Wielinga (UvA)
Copromotor:Dr. M.W. van Someren (UvA)
Promotie: 31 January 2008

2008-04
Ander de Keijzer (UT)
Management of Uncertain Data - towards unattended integration
Promotor: Prof. dr. P.M.G. Apers (UT)
Copromotor:Dr. ir. M. van Keulen (UT)
Promotie: 01 februari 2008

2008-05
Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective
Promotor: Prof. Dr. R. J. Wieringa (UT)
Copromotor:Dr. M. U. Reichert (UT)
Promotie: 17 January 2008

2008-06
Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective
Promotor: Prof.dr.ir. Th.P. van der Weide (RUN)
Co-promotor: dr. P.J.F. Lucas (RUN)
Promotie: 18 April 2008

2008-07
Peter van Rosmalen (OU)
Supporting the tutor in the design and support of adaptive e-learning
Promotor: Prof. dr. E.J.R. Koper (OU)
Co-promotor: Prof. dr. P.B. Sloep (OU)
Promotie: 18 April 2008

2008-08
Janneke Bolt (UU)
Bayesian Networks: Aspects of Approximate Inference
Promotor: Prof.dr.ir L.C. van der Gaag (UU)
Promotie: 21 April 2008

2008-09
Christof van Nimwegen (UU)
The paradox of the guided user: assistance can be counter-effective

Promotor: Prof.dr. L. van den Berg (UU)
Co-promotor: Dr. H. van Oostendorp (UU)
Promotie: 31 March 2008

2008-10
Wouter Bosma (UT)
Discourse oriented summarization
Promotor: Prof. dr. ir. A. Nijholt (UT)
Co-promotor: Dr. M. Theune (UT)
Promotie: 27 March 2008

2008-11
Vera Kartseva (VU)
Designing Controls for Network Organizations: A Value-Based Approach
P Promotores: Prof. dr. Y.H. Tan (VU), Prof.dr.ir R. Paans (VU)
Co-promotor: Dr. J. Gordijn (VU)
Promotie: 28 May 2008

2008-12
Jozsef Farkas (RUN)
A Semiotically Oriented Cognitive Model of Knowledge Representation
Promotor: Prof. dr.ir. T.P. van der Weide (RUN)
Co-promotor: Dr. J.J. Sarbo (RUN)
Promotie: 23 April 2008

2008-13
Caterina Carraciolo (UVA)
Topic Driven Access to Scientific Handbooks
Promotor: Prof. dr.M. de Rijke (UVA)
Co-promotor: Dr. J. Kircz (HvA)
Promotie: 25 April 2008

2008-14
Arthur van Bunningen (UT)
Context-Aware Querying; Better Answers with Less Effort
Promotores: Prof. dr.P.M.G. Apers (UT), Prof.dr. L. Feng (Tsinghua University, China)
Co-promotor: Dr. M. Fokkinga (UT)
Promotie: 13 June 2008

2008-15
Martijn van Otterlo (UT)
The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.
Promotores: Prof. dr.ir A. Nijholt (UT), Prof.dr. J.-J.Ch. Meyer (UU)
Co-promotor: Dr. M. Poel (UT)
Referent: Dr. M. Wiering (RUG)
Promotie: 30 May 2008

2008-16
Henriette van Vugt (VU)
Embodied agents from a user's perspective
Promotores: Prof.dr. J. Kleinnijenhuis (VU) Prof.dr. G.C. van der Veer (VU)
Co-promotores: Dr. J. Hoorn (VU), Dr. E.A. Konijn (VU)
Promotie: 25 June 2008

2008-17
Martin Op 't Land (TUD)
Applying Architecture and Ontology to the Splitting and Allying of Enterprises

Promotor: Prof.dr. ir. J.L.G. Dietz (TUD)
Promotie: 13 June 2008

2008-18
Guido de Croon (UM)
Adaptive Active Vision
Promotores: Prof. dr. E.O. Postma (UM), Prof. dr. H.J. van den Herik (UM)
Promotie: 26 June 2008

2008-19
Henning Rode (UT)
From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search
Promotor: Prof.dr. P.M.G. Apers (UT)
Co-promotor: Dr. D. Hiemstra (UT)
Promotie: 27 June 2008

2008-20
Rex Arendsen (UVA)
Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven.
Promotor: prof. dr. T.M. van Engers (UvA)
Promotie: 07 October 2008

2008-21
Krisztian Balog (UVA)
People Search in the Enterprise
Promotor: Prof.dr. M. de Rijke (UVA)
Promotie: 30 September 2008

2008-22
Henk Koning (UU)
Communication of IT-Architecture
Promotores: Prof. dr. S. Brinkkemper (UU), Prof. dr. J.C. van Vliet (VU),
Co-promotor: Dr. R. Bos (UU)
Promotie: 24 September 2008

2008-23
Stefan Visscher (UU)
Bayesian network models for the management of ventilator-associated pneumonia
Promotor:Prof.dr. M.J.M Bonten (UU/ UMCU)
Co-promotores: Dr. P. Lucas (RUN), Dr. C.A.M. Schurink (EUR)
Promotie: 30 September 2008

2008-24
Zharko Aleksovski (VU)
Using background knowledge in ontology matching
Promotor: Prof. dr. F. van Harmelen (VU)
Co-promotor: Dr. W. ten Kate (VU)
Promotie: 05 September 2008

2008-25
Geert Jonker (UU)
Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency
Promotor: Prof.dr. J.- J. Ch. Meyer (UU)
Co-promotor: Dr. F. Dignum (UU)
Promotie: 06 October 2008

2008-26
Marijn Huijbregts (UT)
Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled
Promotor: Prof. dr. F.M.G. de Jong (UT)
Co-promotor: dr. R.J.F. Ordelman (UT)
Promotie: 21 November 2008

2008-27
Hubert Vogten (OU)
Design and Implementation Strategies for IMS Learning Design
Promotor: Prof.dr. E.J.R. Koper (OU)
Co-promotor: Dr. J.M. van Bruggen (OU)
Promotie: 07 November 2008

2008-28
Ildiko Flesch (RUN)
On the Use of Independence Relations in Bayesian Networks
Promotor: Prof. dr. ir. Th.P. van der Weide (RUN)
Co-promotor: Dr. P.J.F. Lucas (RUN))
Promotie: 27 November 2008

2008-29
Dennis Reidsma (UT)
Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans
Promotor: Prof. dr. ir. A. Nijholt (UT)
Co-promotor: Dr. ir. H. J. A. op den Akker (UT)
Promotie: 09 October 2008

2008-30
Wouter van Atteveldt (VU)
Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content
Promotor: Prof.dr. F. van Harmelen (VU), Prof.dr. J. Kleinnijenhuis (VU)
Co-promotor: Dr. S. Schlobach (VU)
Promotie: 11 November 2008

2008-31
Loes Braun (UM)
Pro-Active Medical Information Retrieval
Promotor: Prof.dr. H.J. van den Herik (UM), Prof.dr.ir. A. Hasman (UvA)
Co-promotor: Dr. F. Wiesman (UVA)
Promotie: 29 October 2008

2008-32
Trung H. Bui (UT)
Toward Affective Dialogue Management using Partially Observable Markov Decision Processes
Promotor: Prof. dr. ir. A. Nijholt (UT)
Co-promotor: Dr. J. Zwiers (UT)
Promotie: 09 October 2008

2008-33
Frank Terpstra (UVA)
Scientific Workflow Design; theoretical and practical issues
Promotor: Prof.dr. P.W.Adriaans (UVA)
Co-promotor: Dr. G.R. Meijer (UVA)
Promotie: 06 November 2008

2008-34
Jeroen de Knijf (UU)
Studies in Frequent Tree Mining
Promotor: Prof. dr. A.P.J.M. Siebes (UU)
Co-promotor: Dr. A.J. Feelders (UU)
Promotie: 19 November 2008

2008-35
Ben Torben Nielsen (UvT)
Dendritic morphologies: function shapes structure
Promotor: Prof. dr H.J. van den Herik (UvT), Prof.dr. E.O. Postma (UvT)
Co-promotor: Dr. K.P. Tuyls (TUE)
Promotie: 03 December 2008

Towards an Information Retrieval Theory of Everything*

Djoerd Hiemstra
University of Twente
http://www.cs.utwente.nl/~hiemstra

Abstract

I present three well-known probabilistic models of information retrieval in tutorial style: The binary independence probabilistic model, the language modeling approach, and Google's page rank. Although all three models are based on probability theory, they are very different in nature. Each model seems well-suited for solving certain information retrieval problems, but not so useful for solving others. So, essentially each model solves part of a bigger puzzle, and a united view on these models might be a first step towards an *Information Retrieval Theory of Everything.*

1 Introduction

Many applications that handle information on the internet would be completely inadequate without the support of information retrieval technology. How would we find information on the world wide web if there were no web search engines? How would we manage our email without spam filtering? Much of the development of information retrieval technology, such as web search engines and spam filters, requires a combination of *experimentation* and *theory.* Experimentation and rigorous empirical testing are needed to keep up with increasing volumes of web pages and emails. Furthermore, experimentation and constant adaptation of technology is needed in practice to counteract the effects of people that deliberately try to manipulate the technology, such as email spammers. However, if experimentation is not guided by theory, engineering becomes trial and error. New problems and challenges for information retrieval come up constantly. They cannot possibly be solved by trial and error alone. So, what is the theory of information retrieval?

*A more extensive overview of information retrieval theory, covering eight models is given in: Djoerd Hiemstra, Information Retrieval Models. In: Ayse Goker, John Davies, and Margaret Graham (eds.), *Information Retrieval: Searching in the 21st Century*, Wiley, 2009

There is not one convincing answer to this question. There are many theories, here called *formal models*, and each model is helpful for the development of some information retrieval tools, but not so helpful for the development of others. In order to understand information retrieval, it is essential to learn about these retrieval models. In this paper, I present three well-known probabilistic models of information retrieval in a tutorial style. But first, we will describe what exactly it is that these models model.

2 Terminology

An information retrieval system is a software program that stores and manages information on documents, often textual documents but possibly multimedia. The system assists users in finding the information they need. It does not explicitly return information or answer questions. Instead, it informs on the existence and location of documents that might contain the desired information. Some suggested documents will, hopefully, satisfy the user's information need. These documents are called *relevant* documents. A perfect retrieval system would retrieve only the relevant documents and no irrelevant documents. However, perfect retrieval systems do not exist and will not exist, because search statements are necessarily incomplete and relevance depends on the subjective opinion of the user. In practice, two users may pose the same query to an information retrieval system and judge the relevance of the retrieved documents differently: Some users will like the results, others will not.

There are three basic processes an information retrieval system has to support: the representation of the content of the documents, the representation of the user's information need, and the comparison of the two representations. The processes are visualized in Figure 1. In the figure, squared boxes represent data and rounded boxes represent processes.

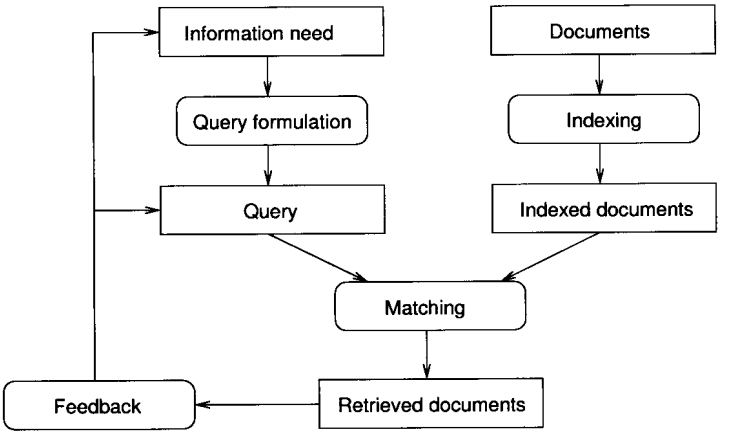


Figure 1: Information retrieval processes

Representing the documents is usually called the *indexing* process. The process takes place off-line, that is, the end user of the information retrieval system is not directly involved. The indexing process results in a representation of the document. Often, full text retrieval systems use a rather trivial algorithm to derive the index representations, for instance an algorithm that identifies words in an English text and puts them to lower case. The indexing process may include the actual storage of the document in the system, but often documents are only stored partly, for instance only the title and the abstract, plus information about the actual location of the document.

Users do not search just for fun, they have a need for information. The process of representing their *information need* is often referred to as the *query formulation* process. The resulting representation is the query. In a broad sense, query formulation might denote the complete interactive dialogue between system and user, leading not only to a suitable query but possibly also to the user better understanding his/her information need: This is denoted by the feedback process in Figure 1.

The comparison of the query against the document representations is called the *matching* process. The matching process usually results in a ranked list of documents. Users will walk down this document list in search of the information they need. Ranked retrieval will hopefully put the relevant documents towards the top of the ranked list, minimizing the time the user has to invest in reading the documents. Simple but effective ranking algorithms use the frequency distribution of terms over documents, but also other statistics, such as the number of hyperlinks that point to the document. Ranking algorithms based on statistical approaches easily halve the time the user has to spend on reading documents. The theory behind ranking algorithms is a crucial part of information retrieval and the major theme of this paper.

3 Models of Information Retrieval

There are two good reasons for having models of information retrieval. The first is that models guide research and provide the means for academic discussion. The second reason is that models can serve as a blueprint to implement an actual retrieval system.

Mathematical models are used in many scientific areas with the objective to understand and reason about some behavior or phenomenon in the real world. One might for instance think of a model of our solar system that predicts the position of the planets on a particular date, or one might think of a model of the world climate that predicts the temperature given the atmospheric emissions of greenhouse gases. A model of information retrieval predicts and explains what a user will find relevant given the user query. The correctness of the model's predictions can be tested in a controlled experiment. In order to do predictions and reach a better understanding of information retrieval, models should be firmly grounded in intuitions, metaphors and some branch of mathematics. Intuitions are important because they help to get a model accepted as reason-

able by the research community. Metaphors are important because they help to explain the implications of a model to a bigger audience. For instance, by comparing the earth's atmosphere with a greenhouse, non-experts will understand the implications of certain models of the atmosphere. Mathematics are essential to formalise a model, to ensure consistency, and to make sure that it can be implemented in a real system. As such, a model of information retrieval serves as a blueprint which is used to implement an actual information retrieval system.

The following sections will describe three models of information retrieval rather extensively. Section 4 describes the classical probabilistic retrieval model, Section 5 describes the language modeling approach, and Section 6 describes the page rank model. Section 7 describes an approach to unify these three rather distinct models in a attempt to construct a "theory of everything". Section 8 concludes this paper.

4 The probabilistic retrieval model

Several approaches that try to model matching and ranking using *probability theory*. The notion of the probability of something, for instance the probability of relevance notated as $P(R)$, is usually formalized through the concept of an experiment, where an experiment is the process by which an observation is made. The set of all possible outcomes of the experiment is called the sample space. In the case of $P(R)$ the sample space might be $\{relevant, irrelevant\}$, and we might define the random variable R to take the values $\{0, 1\}$, where $0 = \text{irrelevant}$ and $1 = \text{relevant}$.

Let's define an experiment for which we take one document from the collection at random: If we know the number of relevant documents in the collection, say 100 documents are relevant, and we know the total number of documents in the collection, say 1 million, then the quotient of those two defines the probability of relevance $P(R=1) = 100/1{,}000{,}000 = 0.0001$. Suppose furthermore that $P(D_k)$ is the probability that a document contains the term k with the sample space $\{0, 1\}$, $(0 = \text{the document does not contain term } k,\ 1 = \text{the document contains term } k)$, then we will use $P(R, D_k)$ to denote the *joint probability distribution* with outcomes $\{(0, 0), (0, 1), (1, 0) \text{ and } (1, 1)\}$, and we will use $P(R|D_k)$ to denote the *conditional probability distribution* with outcomes $\{0, 1\}$. So, $P(R=1|D_k=1)$ is the probability of relevance if we consider documents that contain the term k.

Stephen Robertson and Karen Spärck-Jones based their probabilistic retrieval model on this line of reasoning (Robertson and Spärck-Jones 1976). They suggested to rank documents by $P(R|D)$, that is the probability of relevance R given the document's content description D. Note that D is here a vector of binary components, each component typically representing a term. In the probabilistic retrieval model the probability $P(R|D)$ has to be interpreted as follows: there might be several, say 10, documents that are represented by the same D. If 9 of them are relevant, then $P(R|D) = 0.9$. To make this work in

practice, we use Bayes' rule on the probability odds $P(R|D)/P(\overline{R}|D)$, where $\overline{R}$ denotes irrelevance. The odds allow us to ignore $P(D)$ in the computation while still providing a ranking by the probability of relevance. Additionally, we assume independence between terms given relevance.

$$\frac{P(R|D)}{P(\overline{R}|D)} = \frac{P(D|R)P(R)}{P(D|\overline{R})P(\overline{R})} = \frac{\prod_k P(D_k|R)P(R)}{\prod_k P(D_k|\overline{R})P(\overline{R})} \tag{1}$$

Here, D_k denotes the k^{th} component (term) in the document vector. The probabilities of the terms are defined as above from examples of relevant documents. A more convenient implementation of probabilistic retrieval uses the following three order preserving transformations. First, the documents are ranked by sums of logarithmic odds, instead of the odds themselves. Second, the a priori odds of relevance $P(R)/P(\overline{R})$ is ignored. Third, we subtract $\sum_k \log(P(D_k = 0|R)/P(D_k = 0|\overline{R}))$, i.e., the score of the empty document, from all document scores. This way, the sum over all terms, which might be millions of terms, only includes non-zero values for terms that are present in the document.

$$\text{matching-score}(D) = \sum_{\substack{k \in \text{match-} \\ \text{ing terms}}} \log \frac{P(D_k{=}1|R)\,P(D_k{=}0|\overline{R})}{P(D_k{=}1|\overline{R})\,P(D_k{=}0|R)} \tag{2}$$

In practice, terms that are not in the query are also ignored in Equation 2. Making full use of the probabilistic retrieval model requires two things: examples of relevant documents and long queries. Relevant documents are needed to compute $P(D_k|R)$, that is, the probability that the document contains the term k given relevance. Long queries are needed because the model only distinguishes term presence and term absence in documents and as a consequence, the number of distinct values of document scores is low for short queries. For a one-word query, the number of distinct probabilities is two (either a document contains the word or not), for a two-word query it is four (the document contains both terms, or only the first term, or only the second, or neither), for a three-word query it is eight, etc. Obviously, this makes the model inadequate for web search, for which no relevant documents are known beforehand and for which queries are typically short. However, the model is helpful in for instance spam filters. Spam filters accumulate many examples of relevant (no spam or 'ham') and irrelevant (spam) documents over time. To decide if an incoming email is spam or ham, the full text of the email can be used instead of just a few query terms.

5 Language modeling

Language models were applied to information retrieval by a number of researchers in the late 1990's (Ponte and Croft 1998; Hiemstra 1998; Miller et al. 1999). They originate from probabilistic models of language generation developed for automatic speech recognition systems in the early 1980's. Automatic

speech recognition systems combine probabilities of two distinct models: the acoustic model, and the language model. The acoustic model might for instance produce the following candidate texts in decreasing order of probability: "food born thing", "good corn sing", "mood morning", and "good morning". Now, the language model would determine that the phrase "good morning" is much more probable, i.e., it occurs more frequently in English than the other phrases. When combined with the acoustic model, the system is able to decide that "good morning" was the most likely utterance, thereby increasing the quality of the system.

For information retrieval, language models are built for each document. By following this approach, the language model of the NVTI newsletter you are reading now would assign an exceptionally high probability to the word "theory" indicating that this document would be a good candidate for retrieval if the query contains this word. Language models take the following starting point: Given D – the document is relevant – the user will formulate a query by using a term T with some probability $P(T|D)$. The probability is defined by the text of the documents: If a certain document consists of 100 words, and of those the word "good" occurs twice, then the probability of "good" given that the document is relevant is simply defined as 0.02. For queries with multiple words, we assume that query words are generated independently from each other, i.e., the conditional probabilities of the terms $T_1, T_2, \cdots$ given the document are multiplied:

$$P(T_1, T_2, \cdots | D) = \prod_i P(T_i|D) \tag{3}$$

> Note that the notation $P(\ldots)$ is overloaded. Any time we are talking about a different random variable or sample space, we are also talking about a different measure P. So, one equation might refer to several probability measures, all ambiguously referred to as P. Also note that random variables like D and T might have different sample spaces in different models. For instance, D in the language modeling approach is a random variable denoting "*this* is the relevant document", that has as possible outcomes the identifiers of the documents in the collection. However, D in the probabilistic retrieval model is a random variable that has as possible outcomes all possible document descriptions, which in this case are vectors with binary components d_k that denote whether a document is indexed by term k or not.

As a motivation for using the probability of the query given the document, one might think of the following experiment. Suppose we ask one million monkeys to pick a good three-word query for several documents. Each monkey will point three times at random to each document. Whatever word the monkey points to, will be the (next) word in the query. Suppose that 7 monkeys accidentally pointed to the words "information", "retrieval" and "model" for document 1, and only 2 monkeys accidentally pointed to these words for document 2. Then, document 1 would be a better document for the query "information retrieval model" than document 2.

The above experiment assigns zero probability to words that do not occur anywhere in the document, and because we multiply the probabilities of the single words, it assigns zero probability to documents that do not contain all of the words. For some applications this is not a problem. For instance for a web search engine, queries are usually short and it will rarely happen that no web page contains all query terms. For many other applications empty results happen much more often, which might be problematic for the user. Therefore, a technique called *smoothing* is applied: Smoothing assigns some non-zero probability to unseen events. One approach to smoothing takes a linear combination of $P(T_i|D)$ and a background model $P(T_i)$ as follows.

$$P(T_1, \cdots, T_n|D) = \prod_{i=1}^{n} (\lambda P(T_i|D) + (1-\lambda)P(T_i)) \tag{4}$$

The background model $P(T_i)$ might be defined by the probability of term occurrence in the collection, i.e., by the quotient of the total number of occurrences in the collection divided by the length of the collection. In the equation, λ is an unknown parameter that has to be set empirically. Linear interpolation smoothing accounts for the fact that some query words do not seem to be related to the relevance of documents at all. For instance in the query "capital of the Netherlands", the words "of" and "the" might be seen as words from the user's general English vocabulary, and not as words from the relevant document he/she is looking for. In terms of the experiment above, a monkey would either pick a word at random from the document with probability λ or the monkey would pick a word at random from the entire collection. A more convenient implementation of the linear interpolation models can be achieved with order preserving transformations that are similar to those for the probabilistic retrieval model (see Equation 2). We multiply both sides of the equation by $\prod_i (1-\lambda)P(T_i)$ and take the logarithm, which leads to:

$$\text{matching-score}(d) = \sum_{\substack{k \in \text{match-} \\ \text{ing terms}}} \log(1 + \frac{P(T_i|D)}{P(T_i)} \cdot \frac{\lambda}{1-\lambda}) \tag{5}$$

Language models are well-suited in situations that require searching for documents that are *similar* to a query. Instead of modeling what a relevant document looks like, as done by the probabilistic model, the language modeling approach simply returns the document that is most similar to the query. As such a language modeling approach is well-suited for search systems that get ad hoc, short queries, as is for instance the case in web search.

6 Google's page rank model

When Sergey Brin and Lawrence Page launched the web search engine Google in 1998 (Brin and Page 1998), it had two features that distinguished it from other web search engines: It had a simple no-nonsense search interface, and, it used

a radically different approach to rank the search results. Instead of returning documents that closely match the query terms, they aimed at returning high quality documents, i.e., documents from trusted sites. Google uses the hyperlink structure of the web to determine the quality of a page, called *page rank*. Web pages that are linked at from many places around the web are probably worth looking at: They must be *high quality* pages. If pages that have links from other high quality web pages, for instance DMOZ or Wikipedia[1], then that is a further indication that they are likely to be worth looking at. The page rank of a page d is defined as $P(D = d)$, i.e., the probability that d is relevant, exactly as it is used in the language modeling approach of Section 5 as well. It is defined as:

$$P(D{=}d) \;=\; (1{-}\lambda)\frac{1}{\#\text{pages}} \;+\; \lambda \sum_{i|i \text{ links to } d} P(D{=}i)P(D{=}d|D{=}i) \qquad (6)$$

If we ignore $^{(1-\lambda)}/_{\#\text{pages}}$ for the moment, then the page rank $P(D = d)$ is recursively defined as the sum of the page ranks $P(D = i)$ of all pages i that link to d, multiplied by the probability $P(D{=}d|D{=}i)$ of following a link from i to d. One might think of the page rank as the probability that a random surfer visits a page. Suppose we ask the monkeys from the previous section to surf the web from a randomly chosen starting point i. Each monkey will now click on a random hyperlink with the probability $P(D = d|D = i)$ which is defined as one divided by the number of links on page i. This monkey will end up in d. But other monkeys might end up in d as well: Those that started on another page that happens to link to d. After letting the monkeys surf a while, the highest quality pages, i.e., the best connected pages, will have most monkeys that look at it.

The above experiment has a similar problem with zero probabilities as the language modeling approach. Some pages might have no links pointing to them, so they will get a zero page rank. Others might not link to any other page, so you cannot leave the page by following hyperlinks. The solution is also similar to the zero probability problem in the language modeling approach: We smooth the model by some background model, in this case the background is uniformly distributed over all pages. With some unknown probability λ a link is followed, but with probability $1 - \lambda$ a random page is selected, which is like a monkey typing in a random (but valid) URL.

Page rank is a so-called static ranking function, that is, it does not depend on the query. It is computed once off-line at indexing time by iteratively calculating the page ranks of pages at time $t + 1$ from the page ranks calculated in the iteration at time t until they do not change significantly anymore. Once the page rank of every page is calculated it can be used during querying. One possible way to use page rank during querying is as follows: Select the documents that contain all query terms and rank those documents by their page rank. In practice, web search engines like Google use many more factors in their ranking than just page rank alone.

[1] see http://dmoz.org and http://wikipedia.org

7 Putting things together

There is no such thing as a dominating model or theory of information retrieval, unlike the situation in for instance the area of databases where the relational model is *the* dominating database model. In information retrieval, some models work for some applications, whereas others work for other applications. For instance, the probabilistic retrieval model of Section 4 might be a good choice if examples of relevant and non-relevant documents are available; language models in Section 5 are helpful in situations that require models of language similarity; and the page rank model of Section 6 is often used in situations that need modeling of more of less static relations between documents. Despite their differences, these three models all use probability theory. This brings up the questions: Could we combine the models in one coherent model? and, What would such a model look like?

Let's consider the scenario of searching scientific papers as for instance done by Citeseer, Google Scholar or Scopus[2], that is, given a text query, for instance "theory of information retrieval", the system should retrieve the most important research papers in the field. To find the most important research papers on the theory of information retrieval, we need a model that fulfills the following requirements: First and foremost, an important research paper should mention the query terms "theory", "information" and "retrieval" more often than we would expect in random texts. Second, the paper should be cited a lot, preferably by papers that are cited a lot themselves. Third, the paper should fulfill a number of simple criteria and intuitions that we have about good research papers, such as 1) it should be written recently; 2) it should be written in an ISI-rated journal; 3) it should contain examples; 4) it should contain real program code, etc.

1. The paper should mention the query terms

To fulfill the first requirement, i.e., to find papers that use similar language as our query, a language modeling approach would be appropriate. So, we assign the result of Equation 4 to every document. Note however that Equation 4 defines the probability of a query given a document, but obviously, the system should rank by the probability of the documents given the query. These two probabilities are related by Bayes' rule as follows.

$$P(D|T_1, T_2, \cdots, T_n) = \frac{P(T_1, T_2, \cdots, T_n|D)P(D)}{P(T_1, T_2, \cdots, T_n)} \tag{7}$$

The left-hand side of Equation 7 cannot be used directly because the independence assumption presented above assumes terms are independent given the document. So, in order to compute the probability of the document D given the query, we need to multiply Equation 4 by $P(D)$ and divide it by $P(T_1, \cdots, T_n)$. Again, as stated in the previous paragraph, the probabilities themselves are

[2]http://citeseerx.ist.psu.edu, http://scholar.google.com, and http://scopus.com

of no interest, only the ranking of the document by the probabilities is. And since $P(T_1, \cdots, T_n)$ does not depend on the document, ranking the documents by the numerator of the right-hand side of Equation 7 will rank them by the probability given the query. This shows the importance of $P(D)$, the marginal probability, or prior probability of the document, i.e., it is the probability that the document is relevant if we do not know the query (yet). But how to define $P(D)$ properly?

2. The paper should be cited a lot

In fact $P(D)$ defines a static ranking function as described earlier for the page rank model of Section 6. If one research paper cites another, the cited paper is endorsed by the citing paper. Interestingly, Brin and Page (1998) were inspired by citation analysis when developing page rank, and they call the hyperlink graph a *citation graph* in their paper. So, the page rank model would be an excellent model to fulfill requirement 2, and since the page rank score is really a document prior $P(D)$ it can be easily combined with the language modeling score as shown by Equation 7. So, in terms of the language modeling approach, static rankings are simply document priors, i.e., the a-priori probability of the document being relevant, that should be combined with the probability of terms given the document. Document priors can be easily combined with standard language modeling probabilities and are as such powerful means to improve the effectiveness in web search (Kraaij et al. 2002).

3. Intuitions about good research papers

Our third requirement lists a number of intuitions about the properties of a good research paper. Let's assume that these properties are easily detected for a paper. So, for every document we know if it is: 1) written recently (for instance after 2004), 2) published in an ISI-rated journal, 3) containing examples, 4) containing real code. However, are these 4 properties all equally important? If not, how important is each property? Could it be that some properties are not important at all? It might not surprise the reader at this point that the probabilistic retrieval model of Section 4 is able to answer these questions. As said, the probabilistic retrieval model needs examples of relevant and non-relevant documents, in this case examples of important research papers and unimportant research papers. Suppose we ask a group of users to use a basic version of our system for some time, and to rate the research papers found for each query as *important* or *not important*. Using this data, we can estimate for each document its probability of relevance given its 4 properties. This probability can be combined with the page rank model by using it in Equation 6 to replace $1/\#\text{pages}$. The resulting random surfer of this model follows citations with some probability λ, but selects a "random" page by the probability of relevance given the document properties with probability $1 - \lambda$.

Discussion

Let's again explain our model's implications by the analogy of monkeys. Suppose we ask the monkeys from the previous sections to follow random citations in research papers. Instead of starting from a completely random paper they are more likely to start from papers that have a high probability of relevance given their properties, and each time they are allowed to follow a citation, they might also (with probability $1 - \lambda$) return to the important papers. If, after letting them surf around for a while, the number of monkeys on each paper does not change significantly anymore, we ask them to stop following citations. Instead they pick three random words from the paper in which they ended up, one word at a time.

The process is defined in such a way that every research paper has a very small probability of having a monkey ending up there that also picked the words "theory", "information", and "retrieval". The document with the highest probability of this event is most likely to be an important paper on the theory of information retrieval.

8 Conclusion and further reading

This paper describes three information retrieval models in a tutorial style in order to explain the consequences of modeling assumptions. Once the reader is aware of the consequences of modeling assumptions, he or she will be able to choose a model of information retrieval that is adequate in new situations. Although each model is well-suited for certain applications and not so useful for others, their modeling assumptions do not necessarily contradict each other, and unified modeling approaches are certainly possible.

A much more elaborate version of this paper that covers eight models instead of only three can be found in the book by Goker, Davies, and Graham (2009). The book focuses on current trends and achievements in information retrieval. It provides a basis for understanding recent developments in the field and outlines directions for information search technologies in the near future and beyond. The book contains exercises, making it a good candidate for information retrieval courses in both undergraduate and graduate programs.

References

Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems 30*(1-7), 107–117.

Goker, A., J. Davies, and M. Graham (eds.) (2009). *Information Retrieval: Searching in the 21st Century*. Wiley.

Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *Proceedings of the Second European Conference*

on Research and Advanced Technology for Digital Libraries (ECDL), pp. 569–584.

Kraaij, W., T. Westerveld, and D. Hiemstra (2002). The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pp. 27–34.

Miller, D., T. Leek, and R. Schwartz (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp. 214–221.

Ponte, J. and W. Croft (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 275–281.

Robertson, S. and K. Spärck-Jones (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science 27*, 129–146.

Infinite streams

Jörg Endrullis*, Clemens Grabmayer†,
Dimitri Hendriks‡, and Jan Willem Klop§

February 6, 2009

Abstract

Infinite streams of data are interesting from various points of view: Theoretically, because they are a paradigm example for the application of coalgebraic techniques as well as infinitary rewriting techniques. Practically, because infinite streams arise naturally in several applications concerning data transmissions. And graphically, because they can be visualized by 'drawing turtles' as graphic trajectories that display curious patterns, from aesthetically beautiful to intractably chaotic. In this note we discuss some of these aspects.

1 Background and context

The background and context of this note is the work done during the last three years in the framework of NWO BRICKS-project *Infinity*, a cooperation project of VU Amsterdam, CWI Amsterdam, and Utrecht University. The main objective of that work is to study infinite objects, typically given by recursion equations, from various points of view, to wit, proof theory (find complete proof systems and their relations), rewrite theory (in particular infinitary rewriting), and coalgebraic techniques. The present authors at the VU and the UU were especially concerned with the specialized area of infinite streams over the natural numbers or booleans $0, 1$. In particular we zoomed in on the problem of recognizing the well-definedness of recursive stream specifications, better known as 'productivity' after a proposal of Dijkstra [7], using familiar operators such as tail, zip, even, odd, etc. In fact we outlined a rather expressive format of such stream operators or functions, called the pure stream format, and proved decidability of productivity for recursive stream definitions staying within in the pure format.

*Dept. of Computer Science, Vrije Universiteit Amsterdam, e-mail: joerg@few.vu.nl.
†Dept. of Philosophy, Universiteit Utrecht, e-mail: clemens@phil.uu.nl.
‡Dept. of Computer Science, Vrije Universiteit Amsterdam, e-mail: diem@cs.vu.nl.
§Dept. of Computer Science, Vrije Universiteit Amsterdam, e-mail: jwk@cs.vu.nl.

2 History: streams in various disciplines

Infinite streams, also called infinite sequences, infinite words, or ω-words, are the subject of study in several disciplines. A landmark was the work of Axel Thue, who devised in 1906 infinite sequences of symbols avoiding certain simple patterns such as squares ww or cubes www where w is a finite word. He introduced the famous sequence 0110100110010110... that is obtained from the morphism $0 \to 01$, $1 \to 10$ with initial word 0. This sequence is cube-free and turned out to be ubiquitous indeed (see [1]), and was rediscovered by Marston Morse in 1921 in the mathematical context of dynamical systems and ergodic theory. The Thue–Morse sequence is also known to be an 'automatic sequence' (see [2]), and in particular it is a morphic sequence or D0L sequence. In the terminology of [2], the sequence is obtained by a 'substitution', another word for morphism.

So infinite streams (we will often just call them streams) arise in theoretical computer science, in particular in the areas of formal languages and combinatorics, and also in mathematics, with applications in dynamical systems and number theory. They also appear on the more practical side of computer science where functional programming languages reside [17].

3 Productivity

The notion of productivity (sometimes also referred to as liveness) was first mentioned by Dijkstra [7]. Since then several papers [18, 15, 5, 11, 17, 4] have been devoted to criteria ensuring productivity. Technically, the common essence of these approaches is a quantitative analysis, in terms of a quantitative input/output behaviour of a stream function f by a 'modulus of production' $\nu_f : \mathbb{N}^k \to \mathbb{N}$ with the property that the first $\nu_f(n_1, \ldots, n_k)$ elements of $f(t_1, \ldots, t_k)$ can be computed whenever the first n_i elements of t_i are defined. We have adopted and elaborated this approach; in [9, 10, 8] we describe a calculus by means of which we can compute composition, infimum and fixed points of periodically increasing production functions.

A general observation is that productivity is in some sense the infinitary analogon of finitary termination, a notion that is widely studied in term rewriting, and for which there nowadays exists an extensive tool technology.

Let us now consider two easy examples, that illustrate the essence of the productivity problem. A specification of the Thue–Morse stream is given in Figure 1. For this specification it is fairly easy to see, or even to prove, that it is productive in the sense that unfolding (by infinitary rewriting [12], to be precise) the definitions will never stagnate, but will 'always eventually' produce more elements of the stream. In short, the specification is productive.

Now let us consider a second example:

$$\mathsf{J} = 0 : 1 : \mathsf{even}(\mathsf{J}) \tag{1}$$

$$M = 0 : \mathsf{zip}(\mathsf{inv}(M), \mathsf{tail}(M))$$

$$\begin{aligned}
\mathsf{tail}(x : \sigma) &= \sigma & \mathsf{zip}(x : \sigma, \tau) &= x : \mathsf{zip}(\tau, \sigma) \\
\mathsf{even}(x : \sigma) &= x : \mathsf{odd}(\sigma) & \mathsf{inv}(0 : \sigma) &= 1 : \mathsf{inv}(\sigma) \\
\mathsf{odd}(x : \sigma) &= \mathsf{even}(\sigma) & \mathsf{inv}(1 : \sigma) &= 0 : \mathsf{inv}(\sigma)
\end{aligned}$$

Figure 1: *A pure specification for the Thue–Morse stream.*

where even is defined as before. Although at first glance this specification looks perfectly OK, it nevertheless is not productive: it only produces 4 elements and then continues with infinitely long unproductive calculations. It will produce an infinitary normal form, namely $0:1:0:0:\mathsf{even}^\omega$, but this is not good enough: we want infinite normal forms built from constructors only, not including unevaluated function calls.

These two examples are setting the scene. In more complicated specifications it is not easy to see whether they are productive or not. In fact, the notion is in general undecidable. So we have to find a restricted format for which productivity is still decidable, but such that it is still sufficiently expressive.

In [9, 8, 10] we have defined and analyzed such a restrictive yet expressive format. We will refrain here from stating the full technical definition, but a good first impression is given by the examples in this section. Basically, the specification is a layered one: in the top-layer, there is a recursive definition of the intended stream constant, like M; the recursive definition may actually be a system of recursive equations. In the second layer, a declaration of stream functions is given, like zip. In the third and last layer, we declare the data functions involved. The whole specification consists of orthogonal rewrite rules, thereby guaranteeing confluence, and other useful properties.

Now for this restricted format we do have decidability. And not only theoretical decidability, but practical decidability. We have developed a tool, available at `http://infinity.few.vu.nl/productivity/` that accepts such specifications, and yields the verdict "productive", or "not productive, only producing n elements". The tool produces a pdf file with the productivity analysis in full detail. An example for Thue–Morse is in Figure 2; an example for the unproductive stream specification J is in Figure 3.

Let us consider a variation on the specification for M given in Figure 1. There the use of the 'fine' definition of zip is crucial. If we replace the definition of zip by the 'coarser' one: $\mathsf{zip}^*(x : \sigma, y : \tau) \to x : y : \mathsf{zip}^*(\sigma, \tau)$, then the specification produces only one element. The reason is that in rewrite sequences starting from M, the second argument of zip^* will never match with a stream constructor. The constant M rewrites, in the limit, to an infinite term with no reducible expressions in it, and hence to an infinite normal form. However, as mentioned before, due to the stacking of unevaluated function calls, this is not a *constructor normal form* as required for productivity. The altered specification therefore is not productive. This shows that one has to be careful when replacing stream functions by variants that are 'extensionally equivalent'; the

$$
\begin{aligned}
[\mathsf{M}] &= \mu M.\bullet([\mathsf{zip}]([\mathsf{inv}](M), [\mathsf{tail}](M))) \\
&= \mu M.\bullet(\triangle(\mathsf{box}(\overline{-++}, \mathsf{box}(\overline{-+}, M)), \mathsf{box}(\overline{+-+}, \mathsf{box}(\overline{--+}, M)))) \\
&\to_{\mathsf{R}} \mu M.\bullet(\triangle(\mathsf{box}(\overline{-++}, M), \mathsf{box}(+\overline{-++}, M))) \\
&\to_{\mathsf{R}} \mu M.\mathsf{box}(+\overline{-+}, \triangle(\mathsf{box}(\overline{-++}, M), \mathsf{box}(+\overline{-++}, M))) \\
&\to_{\mathsf{R}} \mu M.\triangle(\mathsf{box}(+\overline{-+}, \mathsf{box}(\overline{-++}, M)), \mathsf{box}(+\overline{-+}, \mathsf{box}(+\overline{-++}, M))) \\
&\to_{\mathsf{R}} \mu M.\triangle(\mathsf{box}(\overline{+-+}, M), \mathsf{box}(++\overline{-++}, M)) \\
&\to_{\mathsf{R}} \triangle(\mu M.\mathsf{box}(\overline{+-+}, M), \mu M.\mathsf{box}(++\overline{-++}, M)) \\
&\to_{\mathsf{R}} \triangle(\mathsf{src}(\infty), \mathsf{src}(\infty)) \\
&\to_{\mathsf{R}} \mathsf{src}(\infty)
\end{aligned}
$$

Figure 2: *Output of our productivity decision tool: a computation yielding that evaluation of the stream constant* M *in the specification of Figure 1 can generate infinitely many data-elements, establishing the specification's productivity.*

$$
\begin{aligned}
[\mathsf{J}] &= \mu J.\bullet(\bullet(\mathsf{box}(\overline{-+-}, J))) \to_{\mathsf{R}} \mu J.\mathsf{box}(+\overline{-+}, \mathsf{box}(+\overline{-+}, \mathsf{box}(\overline{-+-}, J))) \\
&\to_{\mathsf{R}} \mu J.\mathsf{box}(++\overline{-+}, \mathsf{box}(\overline{-+-}, J)) \to_{\mathsf{R}} \mu J.\mathsf{box}(++\overline{-+-}, J) \to_{\mathsf{R}} \mathsf{src}(4)
\end{aligned}
$$

Figure 3: *For the specification* (1) *we obtain that* J *is not productive (only 4 elements can be evaluated).*

property of productivity is sensitive to such replacements, due to the 'intensional' aspect of such stream specifications.

4 Complexity

Another challenging cluster of questions concerns the logical complexity, in terms of the classical arithmetic and analytical hierarchy, of various notions involved in stream specifications. The arithmetical hierarchy classifies sets by the complexity of first-order formulas describing them, which in turn is defined as the number of quantifiers of the prenex normal form. The analytical hierarchy continues the classification using second-order formulas.

We present two of the main facts:

(i) *Productivity is* Π^0_2. The problem of deciding whether a given orthogonal term rewriting system is productive, is Π^0_2-complete — a level of the arithmetical hierarchy —, and thereby equivalent to the well-known uniform halting problem for Turing machines.

(ii) *Infinitary normalization is* Π^1_1. A term rewriting system is called *infinitary normalizing* if all (possibly transfinitely long) rewrite sequences end in a (possibly infinite) normal form; the counterpart of normalization when

considering infinite terms. The complexity of this property exceeds the arithmetical hierarchy and thereby classical first-order theory. Its precise complexity is Π^1_1, a level of the analytical hierarchy.

Both results are taken from work in progress. The result in item (i) has been obtained by Endrullis, Grabmayer and Hendriks, item (ii) by Endrullis, Geuvers and Zantema.

5 Comparing Streams

How can we compare streams? Not with respect to recursion-theoretic complexity, because the streams we are interested in are all computable. Some of the tools that come to mind are Kolmogorov complexity, and the technical notion known as 'subword complexity'. With respect to this measure morphic streams (such as Thue–Morse, Toeplitz) have complexity at most quadratic, whereas the subfamily of sturmian streams, to which the Fibonacci stream belongs, have the lowest possible complexity, namely $n + 1$.

We will consider another approach. As we have seen, the streams Thue–Morse and Toeplitz are strongly related, they are twin brothers, the one can be easily converted into the other. From Thue–Morse to Toeplitz this is just taking the differences of consecutive elements modulo 2, call this operation diff, and the other way around undiff is equally simple. (Actually, there are two undiff's, undiff_0 and undiff_1, depending on choosing the first element.) Both transformations can be easily defined in our framework.

Another way of defining these transformation operations such as diff is by means of a finite state transducer (FST), in which we can read in, starting at the unique root, an infinite word σ, on the way recording how each symbol of σ, depending on the current state that we reached in the FST, is transformed into a finite (possibly empty) word.

An example is M and M/3, where M is the Morse stream and M/3 is the stream obtained by taking every third element. Then it is not hard to find FST's to transform M into M/3 and M/3 back into M. We therefore define that the 'degree' of M and M/3 is the same. An interesting result obtained by Sebastian Stern in his recent master thesis [16] at the VU, is that every arithmetical subsequence (in fact some more) of Morse either is eventually periodic, or is still equivalent to Morse, in that it can be transformed back to Morse.

To make a connection with the turtle trajectories: the turtles can be seen as FST's where stream symbols are uniformly translated into machine instructions such as turn, etc.

To wind up this story, we get a partial order of degrees of streams, where a degree is an equivalence class of streams modulo the equivalence obtained by streams being interconvertible via FST's, see Figure 4. The trivial degree is the degree 0 of eventually periodic streams. We get an ordering between degrees in a straightforward way. Of special interest to us are minimal non-trivial degrees (call them 'prime' degrees), that have the property that there is

no non-trivial degree less than that degree. Our favourite conjecture is that the degree of Morse is such a prime degree. Prime degrees are not hard to find, e.g. one is the degree generated by the stream 11010010001000010000001 . . ..

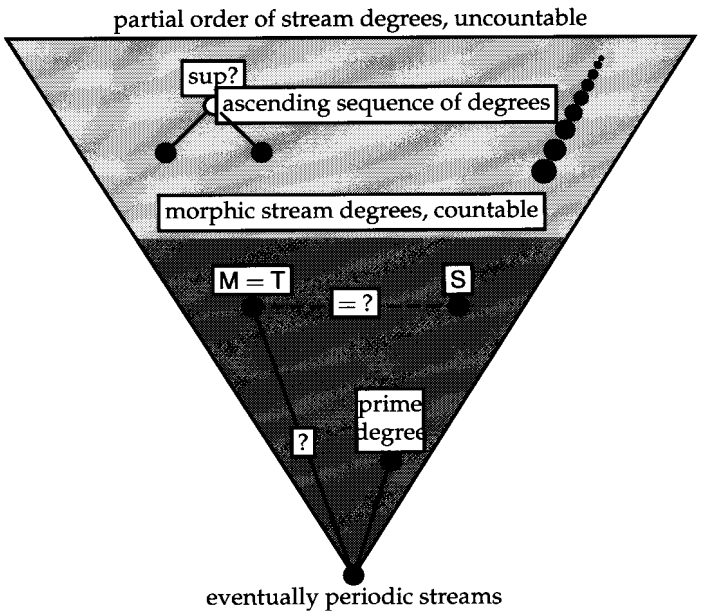


Figure 4: *The partial order of degrees of streams.*

6 Graphics

Recently, a surprising connection was discovered by Holdener and Ma [14], who showed that there is a strong connection beween the Thue–Morse stream and the famous snowflake of Helge von Koch [13], also from 1906. The discovery was done by using 'turtle graphics', an endeavour that was used in the 1980's for didactical purposes: let a turtle with a writing head and a tiny memory draw a trajectory in the plane according to a simple program describing the elementary drawing steps. In the present case, Holdener and Ma gave the turtle as program the Thue–Morse sequence, where a 0 was interpreted as: draw a line segment of one unit length in the direction in which the write head is positioned, 0 turn the write head over $\frac{\pi}{3}$. The first 50 or so steps a trajectory is drawn that crosses itself or overlaps itself, but does not evoke many associations. However, when the drawing is continued, sometimes scaling back the figure when it spills over the edges of the screen, a most remarkable phenomenon appears: the trajectory starts to resemble the Koch snowflake. And indeed, in the limit, one obtains precisely the snowflake. The limiting process is interesting, in that it uses the Hausdorff metric.

Subsequently, it was pointed out by Allouche [3] that such a connection between Thue–Morse and Koch was already implicit present in the work of

F.M. Dekking [6], in the terminology of exponential sums. An even simpler rendering was noticed by the present authors: the Toeplitz stream T, that is the stream of 'first differences' of the Thue–Morse stream, considered as a turtle program, gives the Koch snowflake right away; see Figure 10. The drawing instructions are clear from the figure. Note that the Koch snowflake nicely connects the Thue–Morse stream and the Toeplitz stream: the first can be read off above the snowflake, the second below the snowflake.

Another nice exercise is to consider the Sierpinski curve, see Figure 5, with

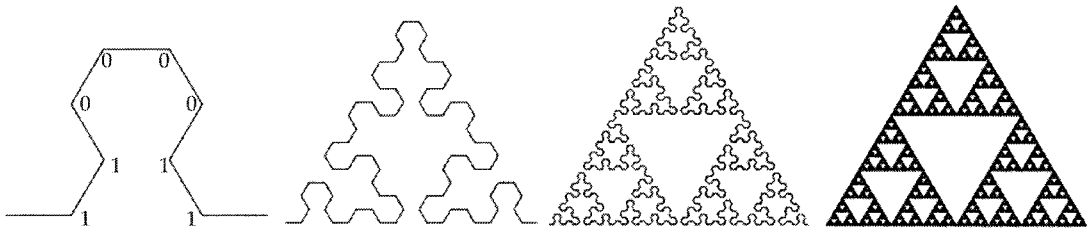


Figure 5: Construction of the Sierpinski triangle.

its associated 0-1-stream, and find a neat definition of that stream. It turns out that the Sierpinski stream, as we will call it, belongs to a familiar family of streams, namely the Toeplitz streams, that are in a simple sense 'self-generating'. Triggered by these turtle drawings, we generated several thousands of such drawings, for various streams and various drawing instructions.

Several streams yield chaotic pictures, depending on the turtle instructions. For the Fibonacci stream defined by the morphism $0 \to 01$, $1 \to 0$ starting on 0, we often find regular, aesthetically pleasing patterns, see Figure 9.

An example of a stream whose turtle trajectories are chaotic is the Kolakoski stream. This stream K is identical to the sequence of its 'run-lengths', that is, $K = 22\ 11\ 2\ 1\ 22\ 1\ 22\ 11\ldots$, where the run-lengths of the alternating blocks of similar symbols are $2\ 2\ 1\ 1\ 2\ 1\ 2\ 2\ldots$. The Kolakoski stream can also be specified in the format of [8], as follows:

$$K = f_2(2 : \mathsf{tail}(K))$$

$$f_1(1:\sigma) = 1 : f_2(\sigma) \qquad f_1(2:\sigma) = 1:1:f_2(\sigma)$$

$$f_2(1:\sigma) = 2 : f_1(\sigma) \qquad f_2(2:\sigma) = 2:2:f_1(\sigma)$$

For the turtle trajectory for K we find very chaotic patterns, see Figure 6. This stream is the subject of many open problems.

7 Problems

We conclude by mentioning a number of areas for further developments:

(i) *Integration into functional languages environments*: In collaboration with people from the functional programming community, we want to examine whether the results on automated recognition of stream productivity [9, 10] can be used to improve compilers.

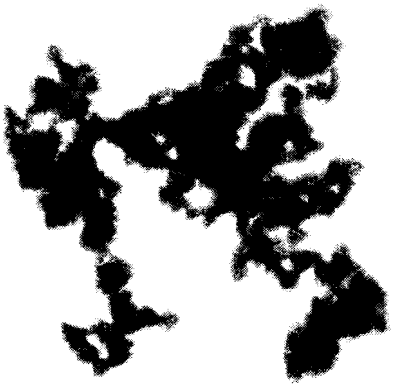


Figure 6: *A turtle trajectory for the Kolakoski sequence* K *for a prefix of* $2 \cdot 10^6$ *entries.*

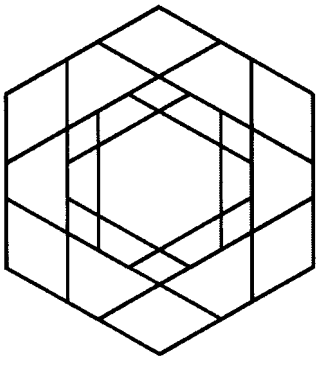


Figure 7: *A turtle trajectory for the Toeplitz stream, which can be obtained by the morphism* $0 \to 11$, $1 \to 10$ *on the initial word* 1.

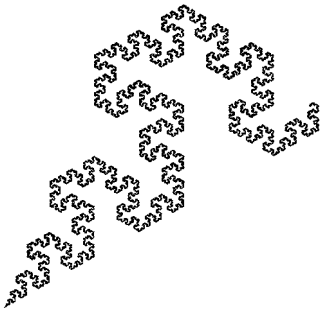


Figure 8: *A turtle trajectory for the Mephisto Waltz, the stream which can be obtained from the morphism* $0 \to 001$, $1 \to 110$ *on the initial word* 0.

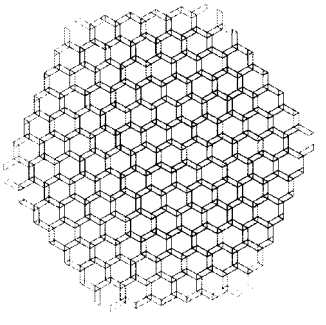


Figure 9: *A turtle trajectory for the Fibonacci stream.*

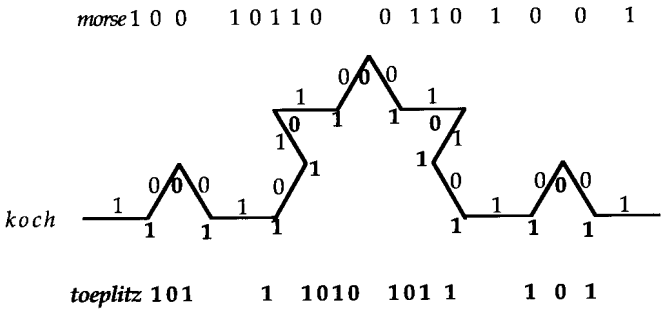


Figure 10: *Construction of the Koch snowflake from the Toeplitz stream.*

(ii) *Relation between productivity and unique definedness*: There is a close connection between productivity of a stream specification $\mathcal{S}$ and the property of $\mathcal{S}$ to have a unique solution. It is easy to see that productivity implies unique solvability, but the converse direction fails in general. Here we mention a recent interesting result by Zantema giving a criterion for unique solvability in terms of finitary termination.

(iii) *Connection with finitary termination*: There is a large arsenal of termination methods in the finitary setting, which can be applied in various ways (for example by the result mentioned in (ii)) to stream specifications.

(iv) *Connection with with lambda calculus theory of Ω-free Böhm Trees*, i.e. totally defined Böhm Trees.

(v) *Graphical aspects*: We aim to study the connections between the graphical aspects of turtle renderings of streams, and the complexity properties of the streams. At present this is 'terra incognita' for us.

References

[1] J.-P. Allouche and J. Shallit. The Ubiquitous Prouhet–Thue–Morse Sequence. In *Sequences and Their Applications: Proceedings of SETA '98*, pages 1–16. Springer–Verlag, 1999.

[2] J.-P. Allouche and J. Shallit. *Automatic Sequences: Theory, Applications, Generalizations*. Cambridge University Press, New York, 2003.

[3] J.-P. Allouche and G. Skordev. Von Koch and Thue–Morse revisited, 2006.

[4] W. Buchholz. A Term Calculus for (Co-)Recursive Definitions on Stream-like Data Structures. *Annals of Pure and Applied Logic*, 136(1-2):75–90, 2005.

[5] Th. Coquand. Infinite Objects in Type Theory. In H. Barendregt and T. Nipkow, editors, *TYPES*, volume 806, pages 62–78. Springer-Verlag, Berlin, 1994.

[6] F.M. Dekking. On the distribution of digits in arithmetic sequences. In *Séminaire de Théorie des Nombres de Bordeaux*, volume 12, pages 3201–3212, 1983.

[7] E.W. Dijkstra. On the Productivity of Recursive Definitions, 1980. EWD749, available at `http://www.cs.utexas.edu/users/EWD/`.

[8] J. Endrullis, C. Grabmayer, and D. Hendriks. Data-Oblivious Stream Productivity. In *Logic for Programming, Artificial intelligence and Reasoning 2008*, number 5330 in LNCS, pages 79–96. Springer, 2008.

[9] J. Endrullis, C. Grabmayer, D. Hendriks, A. Isihara, and J.W. Klop. Productivity of Stream Definitions. In *Proceedings of FCT 2007*, number 4639 in LNCS, pages 274–287. Springer, 2007.

[10] J. Endrullis, C. Grabmayer, D. Hendriks, A. Isihara, and J.W. Klop. Productivity of Stream Definitions. Technical Report Preprint 268, Logic Group Preprint Series, Department of Philosophy, Utrecht University, 2008. Accepted for publication in a forthcoming special issue of TCS. Available at `http://www.phil.uu.nl/preprints/lgps/`.

[11] J. Hughes, L. Pareto, and A. Sabry. Proving the Correctness of Reactive Systems Using Sized Types. In *POPL '96*, pages 410–423, 1996.

[12] J.W. Klop and R.C. de Vrijer. Infinitary Normalization. In S. Artemov, H. Barringer, A.S. d'Avila Garcez, L.C. Lamb, and J. Woods, editors, *We Will Show Them: Essays in Honour of Dov Gabbay (2)*, pages 169–192. College Publications, 2005.

[13] H. von Koch. Une méthode géométrique élementaire pour l'étude de certaines questions de la theorie des courbes planes. In *Acta Math.*, volume 30, pages 145–174, 1906.

[14] J. Ma and J.A. Holdener. When Thue–Morse Meets Koch. In *Fractals: Complex Geometry, Patterns, and Scaling in Nature and Society*, volume 13, pages 191–206, 2005.

[15] B.A. Sijtsma. On the Productivity of Recursive List Definitions. *ACM Transactions on Programming Languages and Systems*, 11(4):633–649, 1989.

[16] S. Stern. The Thue–Morse Sequence. Master's thesis, Vrije Universiteit Amsterdam, 2008.

[17] A. Telford and D. Turner. Ensuring Streams Flow. In *AMAST*, pages 509–523, 1997.

[18] W.W. Wadge. An Extensional Treatment of Dataflow Deadlock. *TCS*, 13:3–15, 1981.

SEMIDEFINITE PROGRAMMING APPROXIMATIONS FOR STABLE SETS, COLOURING, AND CUTS IN GRAPHS

MONIQUE LAURENT

ABSTRACT. In the recent years semidefinite programming has become a widely used tool for designing better efficient approximation algorithms for hard combinatorial optimization problems. In this note we review some constructions for nested semidefinite programming relaxations to 0/1 linear programming problems and some of their main properties. In particular we indicate how to get more compact semidefinite programs, thus more amenable to practical computation, using symmetry reduction. We illustrate these techniques on some basic combinatorial problems dealing with stable sets, cuts and colouring in graphs.

1. INTRODUCTION

Combinatorial optimization deals typically with the problem of finding an optimum object within a finite (but huge) collection of objects. Think for instance of time-tabling, or scheduling problems, or various problems on graphs, like (1) finding a shortest path between two vertices, or (2) finding a shortest tour traversing all edges, or (3) finding a shortest tour traversing all vertices, or (4) finding a maximum set of pairwise non-adjacent vertices, or (5) colouring the vertices with as few colours as possible in such a way that adjacent vertices receive distinct colours, or (6) partitioning the vertices into two classes so as to maximize the number of edges cut by the partition, etc. As is well known, while efficient (polynomial time) algorithms exist for the first two graph problems (1)-(2), the last four problems (3)-(6) are NP-hard.

A typical approach to attack such hard problems is to represent the objects over which one wishes to optimize by 0/1 vectors and to formulate the problem at hand as a 0/1 linear programming (LP) problem, of the form

$$\max\ c^T x \ \text{ s.t. } Ax \leq b \ \text{ and } x \in \{0,1\}^n, \tag{1}$$

where A is an $m \times n$ matrix, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. Without the integrality constraint on the variable x, (1) would be an LP problem, thus solvable in polynomial time. However, adding the integrality condition makes the problem NP-hard. With P denoting the convex hull of the feasible solutions to (1), the objective is then to find a relaxation of the polytope P which is efficient and tight, meaning that one can compute over it in polynomial time and that the returned solution is close enough to the optimum over P.

Linear programming relaxations of P have been traditionnally considered. However, they are sometimes not powerful enough and semidefinite programming, an extension of linear programming where vector variables are replaced by matrix variables constrained to be positive semidefinite, turns out to be a powerful technique for designing better and efficient approximation algorithms for some problem classes. Semidefinite programming is also widely used in other areas like system and control theory (cf. e.g. [4]), but we focus here on its application to combinatorial optimization.

We now recall a few basic facts about semidefinite programs and refer to [23, 47, 49] and references therein for details. Let $\mathcal{S}_n$ denote the set of $n \times n$ symmetric matrices. A matrix $X \in \mathcal{S}_n$ is positive semidefinite, written as $X \succeq 0$, if $u^T X u \geq 0$ for all $u \in \mathbb{R}^n$ or, equivalently, if there exist

vectors $v_1, \ldots, v_n \in \mathbb{R}^n$ for which $X = v_i^T v_j$ $(\forall i, j)$. The standard form of a semidefinite program (SDP) reads

$$\sup \langle C, X \rangle \text{ s.t. } \langle A_j, X \rangle = b_j \ (j = 1, \ldots, m) \text{ and } X \succeq 0, \tag{2}$$

where $C, A_1, \ldots, A_m \in \mathcal{S}_n$, $b \in \mathbb{R}^m$ are given, $X \in \mathcal{S}_n$ is the matrix variable, and $\langle C, X \rangle = \mathrm{Tr}(C^T X)$ is the standard inner product. The program (2) is a convex program and, when C, A_j are diagonal matrices, it reduces to a classic linear program. As one can test in polynomial time whether a given rational matrix is positive semidefinite (e.g. using Gaussian elimination), semidefinite programs can be solved in polynomial time to any fixed precision using the ellpsoid method (cf. [15]). Algorithms based on the ellipsoid method are however not practical since their running time is prohibitively high. Instead, interior-point algorithms are widely used in practice; they return an approximate solution (to any given precision) in polynomially many iterations and their running time is efficient in practice for medium size problems. As most currently available interior-point algorithms are designed to take advantage of block-diagonal matrices, it is computationnally easier to solve an SDP involving many small blocks rather than one large matrix. One way to get such block-diagonal SDP's is by exploiting the symmetry structure of the problem at hand (cf. Section 4); moreover one can also design the SDP relaxation to force such a symmetry structure (cf. Section 3.3).

Contents. We present in Section 2 some basic SDP bounds for stable sets, colouring and maximum cut problems, and discuss their properties. Then we see in Section 3 how to construct hierarchies of SDP relaxations leading to the full representation of the combinatorial problem at hand. In Section 4 we indicate how symmetry reduction can be used to get more compact SDP programs, and Section 5 contains a few pointers to further related results.

2. Basic semidefinite bounds for stable sets, coloring and maximum cuts

We have chosen to illustrate the use of semidefinite programming in combinatorial optimization on the following combinatorial problems: maximum stable sets, minimum graph coloring, and maximum cuts in graphs. For these problems, some milestone results have been obtained in the recent years, that spurred an intense research activity and a wealth of results for other combinatorial optimization problems; we refer e.g. to [12, 23, 34, 38] and references therein for a detailed exposition. In this section we introduce some basic SDP relaxations which have led to interesting results for our three problems.

2.1. Basic SDP bounds for stable sets and colouring via Lovász' theta number.

Let $G = (V, E)$ be a graph; V is the set of vertices (or nodes) (typically $V = [1, n]$) and E is the set of edges, which are pairs of distinct vertices, then said to be *adjacent* in G. A *stable* (or *independent*) set in G is a set of vertices that are pairwise non-adjacent, and a *(vertex) colouring* of G is an assignment of colours to the vertices in such a way that adjacent vertices receive distinct colours, in other words, a partition of V into stable sets. Then the stable set problem asks for the *stability number* $\alpha(G)$, defined as the maximum cardinality of a stable set in G, while the graph colouring problem asks for the *chromatic number* $\chi(G)$, defined as the minimum number of colours in a vertex colouring. Both problems are NP-hard [11]. Note that

$$\chi(G) \geq \omega(G), \tag{3}$$

where $\omega(G)$ is the largest cardinality of a *clique* (i.e., a set of pairwise adjacent vertices) in G. Obviously, $\omega(G) = \alpha(\bar{G})$, where $\bar{G}$ is the complement of G, with the same set V of vertices and two distinct vertices being adjacent in $\bar{G}$ precisely when they are not adjacent in G.

For some graphs the inequality (3) is strict. For instance, it is strict for any circuit C_n of odd length $n \geq 5$, as $\omega(C_n) = 2 < \chi(C_n) = 3$, and for the complement of C_n as well. However there are many interesting classes of graphs for which equality $\omega(G) = \chi(G)$ holds. This is the case e.g. for bipartite graphs, line graphs of bipartite graphs, comparability graphs, chordal graphs, and for their complements as well. In fact the class of graphs for which equality $\omega(G) = \chi(G)$ holds not only for G but also for all its induced subgraphs (i.e. all those graphs that can be obtained by deleting vertices in G) turns out to be very interesting; following Berge, graphs in this class are called *perfect graphs*. Thus C_n and its complement $\bar{C}_n$ are not perfect for odd $n \geq 5$. Berge conjectured in 1962 that a graph is perfect if and only if its complement is perfect, which was proved a few years later by Lovász [35]. Berge also conjectured that a graph is perfect if and only if it does not contain any odd circuit of length at least 5, or its complement, as an induced subgraph; this was proved only recently by Chudnovsky et al. [6] and is known as the *strong perfect graph theorem*. It is intriguing to determine the complexity of computing $\alpha(G)$ and $\chi(G)$ for perfect graphs. As we indicate below this can be done in polynomial time, but to show this one has to use semidefinite programming.

Lovász [37] introduced his celebrated *theta number* $\vartheta(G)$, which serves as bound for both $\alpha(G)$ and $\chi(G)$. The theta number is defined via the semidefinite program

$$\vartheta(G) := \max \ \mathrm{Tr}(JX) \ \text{ s.t. } \ \mathrm{Tr}(X) = 1, \ X_{ij} = 0 \ (ij \in E), \ X \succeq 0, \tag{4}$$

where J denotes the all-ones matrix. A basic property of the theta number is that it satisfies the so-called *sandwich inequality*

$$\alpha(G) \leq \vartheta(G) \leq \chi(\bar{G}), \ \text{ or equivalently, } \ \omega(G) \leq \vartheta(\bar{G}) \leq \chi(G). \tag{5}$$

Indeed, if $\mathbf{1}_S \in \{0,1\}^V$ is the incidence vector of a stable set S in G, then $X := \mathbf{1}_S(\mathbf{1}_S)^T/|S|$ is feasible for (4) with objective value $|S|$, which gives $\alpha(G) \leq \vartheta(G)$. On the other hand, if X is a feasible solution to (4) and $V = C_1 \cup \ldots \cup C_k$ is a partition into $k := \chi(\bar{G})$ cliques of G, then

$$0 \leq \sum_{h=1}^{k} (k\mathbf{1}_{C_h} - \mathbf{1}_V)^T X (k\mathbf{1}_{C_h} - \mathbf{1}_V) = k^2 \mathrm{Tr}(X) - k\mathbf{1}_V^T X \mathbf{1}_V = k(k - \mathrm{Tr}(JX)),$$

which implies $\mathrm{Tr}(JX) \leq k$ and thus $\vartheta(G) \leq \chi(\bar{G})$. Another basic property of the theta number is

$$\vartheta(G)\bar{\vartheta}(G) \geq |V|, \ \text{ with equality if } G \text{ is vertex-transitive.} \tag{6}$$

Here, $\bar{\vartheta}$ is the 'complementary' graph parameter defined by $\bar{\vartheta}(G) := \vartheta(\bar{G})$ for any graph G. This convention is also used later for other graph parameters.

For perfect graphs, equality holds throughout in (5), which implies $\alpha(G) = \vartheta(G)$ and $\chi(G) = \vartheta(\bar{G})$. As the theta number can be computed in polynomial to any fixed precision, the stability number and the chromatic number can be computed in polynomial time for perfect graphs. Moreover, a maximum stable set and a minimum coloring can also be computed in polynomial time for a perfect graph G (by iterated computations of the theta number of certain induced subgraphs of G). These computations thus rely on using semidefinite programming and as of today no alternative efficient algorithm is known.

Lovász' original motivation for introducing the theta number was to bound the *Shannon capacity* of a graph G, which is defined as

$$\Theta(G) := \lim_{k \to \infty} \alpha(G^k)^{\frac{1}{k}}.$$

Here G^k denotes the product of k copies of G, with vertex set V^k and with two disctinct vertices $(u_1, \ldots, u_k)$ and $(v_1, \ldots, v_k)$ being adjacent in G^k if $u_h = v_h$ or $u_h v_h \in E$ for each position $h = 1, \ldots, k$. If we view V as an alphabet and adjacent vertices $u, v \in V$ as letters that can be

confounded, then $\alpha(G^k)$ is the maximum number of words of length k that cannot be confounded, since for any two of them there is a position h where their hth letters are not confoundable. One can verify that $\alpha(G^k) \geq \alpha(G)^k$ and $\vartheta(G^k) \leq \vartheta(G)^k$, which implies

$$\alpha(G) \leq \Theta(G) \leq \vartheta(G).$$

Therefore, when G is perfect, $\Theta(G) = \vartheta(G)$ can thus be computed via semidefinite programming. Lovász could also compute the Shannon capacity of the circuit C_5 using the theta number. He showed that $\Theta(C_5) = \sqrt{5}$, which follows from $\alpha(C_5^2) \geq 5$ (easy to verify) and $\vartheta(C_5) = \sqrt{5}$; the latter follows using (6) since C_5 is vertex-transitive and isomorphic to its complement. The exact value of the Shannon capacity of C_n is not known for odd $n \geq 7$. (Cf. [27] for more details.)

Although it permits to compute the stability and chromatic numbers of perfect graphs, the theta number can be a poor approximation for general graphs. Indeed the gap between $\alpha(G)$ and $\vartheta(G)$ (or $\vartheta(G)$ and $\chi(\bar{G})$) can be as large as $|V|^{1-\epsilon}$ (for any $\epsilon > 0$) [9]. Nevertheless much work has been done on approximating $\alpha(G)$ and $\chi(G)$ with SDP. For example, it is shown in [21] how to colour in randomized polynomial time a k-colourable graph with $O(n^{1-\frac{3}{k+1}}\sqrt{\log n})$ colours [21] using variations of the SDP (4) combined with sophisticated rounding techniques, and the behaviour of the theta number for Erdös-Renyi graphs is analysed in [7].

2.2. **The Goemans-Williamson approximation algorithm for maximum cuts.** Another successful application of semidefinite programming to combinatorial optimization is the celebrated 0.878-approximation algorithm of Goemans and Williamson [14] for the max-cut problem, which we sketch below.

Given a graph $G = (V, E)$ ($|V| = n$) with edge weights $w \in \mathbb{R}_+^E$, the *cut* determined by $S \subseteq V$ is the set of edges having their endnodes in distinct classes of the partition $(S, V \setminus S)$. The *max-cut problem* asks for the maximum weight of a cut in G, denoted as $\mathrm{mc}(G)$. While a minimum weight nonempty cut can be found in polynomial time (using flow algorithms), the max-cut problem is NP-hard [11].

Erdös proposed in 1967 the following simple 1/2-approximation algorithm. Colour the vertices $v_1, \ldots, v_n$ of G with two colours blue and red as follows: First colour v_1 blue. If $v_1, \ldots, v_i$ are coloured, colour v_{i+1} blue if the total weight of the edges joining v_{i+1} to the red vertices in $\{v_1, \ldots, v_i\}$ is more than the total weight of the edges joining v_{i+1} to the blue vertices in this set, and colour v_{i+1} red otherwise. Then the cut given by this partition of V has weight at least $w(E)/2$ and thus at least $\mathrm{mc}(G)/2$. There is an even easier randomized 1/2-approximation algorithm. Namely colour randomly each node blue or red independently, with probability 1/2. The probability that an edge belongs to the cut determined by this partition is 1/2 and thus the expected weight of this cut is $w(E)/2$. Can one do better than 1/2?

It is unfortunately not known how to do better using LP relaxations. Consider, for instance, the classic LP relaxation for max-cut obtained from the triangle inequalities: $x_{ij} - x_{ik} - x_{jk} \leq 0$ and $x_{ij} + x_{ik} + x_{jk} \leq 2$ for $i, j, k \in V$. While this LP relaxation permits to solve max-cut exactly for graphs with no K_5-minor (including planar graphs) [3], it is however a weak relaxation for general graphs, as the integrality gap can be as large as $2 - \epsilon$ ($\forall\ \epsilon > 0$) [41].

However, an improved approximation algorithm was designed by Goemans and Williamson [14] using semidefinite programming. For this, it is convenient to model the max-cut problem using ± 1-valued variables as

$$\mathrm{mc}(G) = \max \sum_{ij \in E} w_{ij}(1 - x_i x_j)/2 \quad \text{s.t.} \quad x \in \{\pm 1\}^n. \tag{7}$$

Observe that the matrix $X := xx^T$ satisfies the constraints: $X \succeq 0$, $X_{ii} = 1$ $(i \in V)$, and $\text{rank}(X) = 1$. If we omit the rank condition, then we find the semidefinite relaxation

$$\text{sdp}(G) := \max \sum_{ij \in E} w_{ij}(1 - X_{ij})/2 \text{ s.t. } X \succeq 0,\ X_{ii} = 1\ (i \in V), \tag{8}$$

which can be solved in polynomial time (to any fixed precision). Let X be an optimum solution to (8). Goemans and Williamson [14] propose the following randomized rounding procedure for constructing a good cut from X. Say X is the Gram matrix of $v_1, \ldots, v_n \in \mathbb{R}^n$, i.e. $X_{ij} = v_i^T v_j$ $\forall i, j \in V$. Consider a random unit vector $r \in \mathbb{R}^n$. The hyperplane with normal r splits V into two sets, depending on the sign of $r^T v_i$ $(i \in V)$, giving $x \in \{\pm 1\}^n$ with $x_i = 1$ iff $r^T v_i \geq 0$. As the probability that an edge ij lies in the cut determined by this partition is equal to $\frac{1}{\pi} \arccos(v_i^T v_j)$, the expected weight of this cut is equal to

$$\sum_{ij \in E} w_{ij} \frac{\arccos(v_i^T v_j)}{\pi} = \sum_{ij \in E} w_{ij} \frac{1 - v_i^T v_j}{2} \frac{2}{\pi} \frac{\arccos(v_i^T v_j)}{1 - v_i^T v_j} \geq \alpha_{\text{GW}}\ \text{sdp}(G) \geq\ 0.878567\ \text{mc}(G),$$

after setting $\alpha_{\text{GW}} := \min_{0 < \vartheta \leq \pi} \frac{2}{\pi} \frac{\vartheta}{1 - \cos \vartheta}$ and observing that $\alpha_{\text{GW}} >\ 0.878567$. This implies $\text{mc}(G) \geq \alpha_{\text{GW}}\ \text{sdp}(G)$. This randomized algorithm can be derandomized to yield in polynomial time a deterministic cut achieving the same performance ratio.

Much research has been done trying to improve the Goemans-Williamson approximation algorithm for max-cut and to extend and apply it to other problems. Although improved algorithms could be designed for special graph classes, no better approximation ratio could yet be shown for the general max-cut problem. It has in fact been shown that α_{GW} is the best possible approximation ratio for max-cut that can be achieved in polynomial time (if P $\neq$ NP) under the Unique Games Conjecture [22]. On the negative side, Håstad [20] proved that no polynomial time approximation algorithm exists for max-cut with performance guarantee better than $16/17 \sim 0.94117$ (if P $\neq$ NP). We now indicate a natural extension to ± 1-quadratic programming:

$$\text{m}(A) := \max_{x \in \{\pm 1\}^n} x^T A x, \tag{9}$$

and its SDP relaxation:

$$\text{sdp}(A) := \max_{u_1, \ldots, u_n \in S^{n-1}} \sum_{i,j=1}^{n} A_{ij} u_i^T u_j = \max \text{Tr}(AX) \text{ s.t. } X \succeq 0,\ X_{ii} = 1\ (i \in V), \tag{10}$$

where $A \in \mathbb{R}^{n \times n}$ and S^{n-1} is the unit sphere in $\mathbb{R}^n$. When $A = L_w$, the Laplacian matrix of (G, w) (with $(L_w)_{ii} := \sum_{j | ij \in E} w_{ij}$, $(L_w)_{ij} := -w_{ij}$ if $ij \in E$ and 0 otherwise), we find the programs (7) and (8), in which case $\text{m}(L_w)/\text{sdp}(L_w) \geq 0.8785$. For general $A \succeq 0$, the above analysis can be adapted to show the inequality: $\text{m}(A) \geq \frac{2}{\pi}\ \text{sdp}(A)$ [40]. The constant of Grothendieck is a fundamental tool in functional analysis, which corresponds to the integrality gap of the following analogues of the programs (9)-(10):

$$\text{m}_G(A) := \max_{x \in \{\pm 1\}^n,\ y \in \{\pm 1\}^m} y^T A x = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} x_i y_j,$$

$$\text{sdp}_G(A) := \max_{u_1, \ldots, u_n, v_1, \ldots, v_m \in S^{n+m-1}} \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} u_i^T v_j,$$

where $A \in \mathbb{R}^{m \times n}$. Grothendieck (1953) showed the existence of a smallest constant K_G satisfying $\text{sdp}_G(A) \leq K_G \text{m}_G(A)$ for all $A \in \mathbb{R}^{m \times n}$. It is known that $K_G \geq 1.68$ [42] and $K_G \leq \frac{\pi}{2 \ln(1 + \sqrt{2})} \sim$

1.782 (cf. [1]). This has been used e.g. for deriving an efficient approximation algorithm for the cut-norm of matrices [1], and an extension to quadratic forms on graphs can be found in [2].

3. Hierarchies of semidefinite programming relaxations

We saw above how to define in a natural way a semidefinite relaxation for the maximum stable set problem (via the SDP (4)) and for the max-cut problem (via the SDP (8)). Several procedures have been proposed for constructing nested (increasingly tight) SDP relaxations (cf. [28, 39, 45], also [29, 34] and references therein). We briefly sketch some of these constructions.

3.1. **The N_+-operator of Lovász-Schrijver.** Lovász and Schrijver [39] introduced the N_+-operator which, given a convex set $K \subseteq [0,1]^V$, produces the convex set $N_+(K)$ consisting of the vectors $x \in \mathbb{R}^V$ for which

$$(11) \quad \exists X \text{ s.t. } x = \mathrm{diag}(X),\ \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0,\ \frac{Xe_j}{x_j} \in K \text{ if } x_j \neq 0,\ \frac{x - Xe_j}{1 - x_j} \in K \text{ if } x_j \neq 1.$$

Let K_I denote the convex hull of the integer points in K, thus the polytope we are interested in. We have:

$$K_I \subseteq N_+(K) \subseteq K.$$

Indeed, if x is an integer point in K, then $X := xx^T$ satisfies the conditions in (11), which shows the left inclusion, and the right inclusion follows easily after noting that $Xe_j = 0$ if $x_j = 0$ and $Xe_j = x$ if $x_j = 1$, and x, X are as in (11).

Defining iteratively $N_+^t(K) := N_+(N_+^{t-1}(K))$ for $t \geq 2$, $N_+^1(K) = N_+(K)$, we obtain a hierarchy of SDP relaxations of K_I, which finds K_I in $n = |V|$ iterations, i.e. $N_+^n(K) = K_I$. An important algorithmic property is that, for any *fixed* t, if one can optimize in polynomial time over K, then the same holds for $N_+^t(K)$.

As in [39], consider for instance the case when $K = \mathrm{FR}(G)$, the fractional stable set polytope of G, defined by the constraints $0 \leq x_i \leq 1$ $(i \in V)$ and $x_i + x_j \leq 1$ $(ij \in E)$. Then K_I is the convex hull of the incidence vectors of all stable sets in G. In fact, $K_I = N_+^t(K)$ for any $t \geq \alpha(G)$. Moreover, for $t = 1$, $N_+(K)$ gives an SDP bound for $\alpha(G)$ which is at least as strong as the theta number, and $N_+(K) = K_I$ when G is a perfect graph. As a clique and a stable set share at most one node, the clique inequalities $\sum_{i \in C} x_i \leq 1$ (C clique in G) are valid for K_I. They are in fact also valid for $N_+(K)$. However, if we omit positive semidefiniteness in (11) (leading to the N-operator), then $t = \omega(G) - 2$ iterations of the N-operator are needed before $N^t(K)$ satisfies all clique inequalities. Thus adding the semidefinite condition definitely helps!

3.2. **A simple recipe for SDP relaxations.** Here is a simple construction for SDP relaxations, leading to refinements of the hierarchy of Lovász and Schrijver. The basic idea is to 'lift' a vector $x \in \{0,1\}^V$ to a higher dimensional vector $y = (x_I := \prod_{i \in I} x_i)_{I \in \mathcal{A}}$ containing the products of the x_i's indexed by a given subset $\mathcal{A}$ of the full power set $\mathcal{P}(V)$. Then the matrix $Y = yy^T$ obviously satisfies the constraints: (i) $Y \succeq 0$, (ii) $Y_{\emptyset,\emptyset} = 1$, and (iii) the (I,J)-entry $Y_{I,J}$ depends only on the union $I \cup J$ (as $Y_{I,J} = x_{I \cup J}$). A matrix indexed by $\mathcal{A}$ and satisfying (iii) is of the form $M_{\mathcal{A}}(z) := (z_{I \cup J})_{I,J \in \mathcal{A}}$ for some $z \in \mathbb{R}^{\mathcal{A} \cup \mathcal{A}}$ (where $\mathcal{A} \cup \mathcal{A} = \{I \cup J \mid I, J \in \mathcal{A}\}$), and is known as a *(combinatorial) moment matrix*. Moreover, if x satisfies a linear constraint $\sum_{i \in V} a_i x_i \leq a_0$, then the lifted vector y satisfies $(a_0 - \sum_{i \in V} a_i x_i) yy^T \succeq 0$, yielding the *localizing constraint* $M_{\mathcal{A}}(az) \succeq 0$, after setting $(az)_I := a_0 z_I - \sum_{i \in V} a_i z_{I \cup \{i\}}$ for all I.

The next lemma implies that, when imposing the 'full' SDP constraints $M_{\mathcal{P}(V)}(z) \succeq 0$ combined with the 'full' localizing constraints, we obtain an exact (SDP or LP) representation of the original 0/1 problem.

Lemma 1. *The following assertions are equivalent for* $z \in \mathbb{R}^{\mathcal{P}(V)}$.

(i) $M_{\mathcal{P}(V)}(z) \succeq 0$.

(ii) $\sum_{S' | S \subseteq S' \subseteq V} (-1)^{|S' \setminus S|} y_{S'} \geq 0$ *for all* $S \subseteq V$.

(iii) $z = \sum_{S \subseteq V} \lambda_S z_S$ *for some* $\lambda_S \geq 0$, *after setting* $z_S := (\prod_{i \in I} (\mathbf{1}_S)_i)_{I \subseteq V} \in \{0,1\}^{\mathcal{P}(V)}$.

If we choose $\mathcal{A} = \mathcal{P}_t(V)$, the collection of all subsets of V with size at most t, then we obtain the SDP hierarchy of Lasserre [28]. If we choose $\mathcal{A} = \mathcal{P}(U)$ for all $U \in \mathcal{P}_t(V)$, then we obtain the hierarchy of Sherali-Adams [45] (thus an LP hierarchy, in view of Lemma 1). In fact, the Lasserre hierarchy refines both the Lovász-Schrijver and the Sherali-Adams hierarchies [29].

Applying this to the fractional stable set polytope $K = \mathrm{FR}(G)$, the Lasserre relaxation of order t gives the upper bound

$$\mathrm{las}^{(t)}(G) := \max_{z \in \mathbb{R}^{\mathcal{P}_{2t}(V)}} \sum_{i \in V} z_i \ \text{ s.t. } \ z_\emptyset = 1, \ M_{\mathcal{P}_t}(z) \succeq 0, \ z_{ij} = 0 \ (ij \in E) \tag{12}$$

for the stability number $\alpha(G)$, while the analogous program

$$\psi_{\mathrm{las}}^{(t)}(G) := \min_{z \in \mathbb{R}^{\mathcal{P}_{2t}(V)}} z_\emptyset \ \text{ s.t. } \ M_{\mathcal{P}_t(V)}(z) \succeq 0, \ z_i = 1 \ (i \in V), \ z_{ij} = 0 \ (ij \in E) \tag{13}$$

yields a lower bound for the chromatic number $\chi(G)$. In fact, $\psi_{\mathrm{las}}^{(t)}(G)$ remains bounded above by the *fractional chromatic number*:

$$\chi_f(G) := \min \sum_{S \text{ stable}} \lambda_S \ \text{ s.t. } \ \sum_{S \text{ stable}} \lambda_S \mathbf{1}_S = \mathbf{1}_V, \ \lambda_S \geq 0. \tag{14}$$

That is, $\psi_{\mathrm{las}}^{(t)} \leq \chi_f \leq \chi$ for all t. Moreover, the pair $(\mathrm{las}^{(t)}, \psi_{\mathrm{las}}^{(t)})$ is equal to $(\vartheta, \bar{\vartheta})$ for $t = 1$, to (α, χ_f) for $t \geq \alpha$, and it satisfies the analogue of the reciprocity relation (6) for any $t \geq 1$ [16, 17].

This construction also applies to max-cut (and to (9)). However, as variable x is now ± 1-valued (instead of 0, 1-valued), one should now require that the (I, J)-th entry in a moment matrix $M_{\mathcal{A}}(z)$ depends only on the symmetric difference $I \Delta J$ (instead of the union $I \cup J$). At order $t = 1$ we find the SDP relaxation (2) (or (10)); geometric properties of these relaxations can be found in [31].

3.3. **A more economical block-diagonal hierarchy.** The SDP (12) involves a matrix of size $O(n^{2t})$, which is thus too costly to compute for large n or t. More economical block-diagonal variations of this hierarchy are proposed in [19], based on the following idea: Instead of the full matrix $M_{\mathcal{P}_t(V)}(z)$, consider several smaller blocks arising from principal submatrices of it. Namely, for any $T \subseteq V$ of size $t-1$, consider the matrices $M(T; z)$ indexed by $\bigcup_{S \subseteq T} \mathcal{A}_S$, where $\mathcal{A}_S := \{S, S \cup \{i\} \mid i \in V\}$. These matrices need only the components of z indexed by $\mathcal{P}_{t+1}(V)$ (as opposed to $\mathcal{P}_{2t}(V)$ in (12)) and each $M(T; z)$ has a very symmetric block-structure. For instance, for $T = \{1, 2\}$,

$$M(T; z) = \begin{pmatrix} A_\emptyset(z) & A_1(z) & A_2(z) & A_{12}(z) \\ A_1(z) & A_1(z) & A_{12}(z) & A_{12}(z) \\ A_2(z) & A_{12}(z) & A_2(z) & A_{12}(z) \\ A_{12}(z) & A_{12}(z) & A_{12}(z) & A_{12}(z) \end{pmatrix} \succeq 0 \Longleftrightarrow \begin{cases} A_\emptyset(z) - A_1(z) - A_2(z) + A_{12}(z) \succeq 0 \\ A_1(z) - A_{12}(z) \succeq 0 \\ A_2(z) - A_{12}(z) \succeq 0 \\ A_{12}(z) \succeq 0 \end{cases}$$

(where blocks are indexed by $\mathcal{A}_\emptyset, \mathcal{A}_{\{1\}}, \mathcal{A}_{\{2\}}, \mathcal{A}_{\{1,2\}}$). A simple adaptation of Lemma 1 implies that each $M(T;z)$ can be block-diagonalized, namely

$$M(T;z) \succeq 0 \Longleftrightarrow \sum_{S' | S \subseteq S' \subseteq T} (-1)^{|S' \setminus S|} A_{S'}(z) \succeq 0 \quad \forall S \subseteq T. \tag{15}$$

In this way we obtain a *block-diagonal hierarchy*, where the condition '$M_{\mathcal{P}_t(V)}(z)$' is replaced by '$M(T;z) \succeq 0$ for all $T \subseteq V$ with $|T| = t-1$'; the localizing conditions can be analogously replaced by block-diagonal conditions. This block-diagonal hierarchy can be seen as an explicit analogue of the Lovász-Schrijver hierarchy, which is simpler (not recursive) and more economical (as it uses less variables and constraints); cf. [19] for details.

Let $\ell^{(t)}$ denote the block-diagonal variation of the parameter $\text{las}^{(t)}$ from (12), thus defined by

$$\ell^{(t)}(G) = \max \sum_{i \in V} z_i \text{ s.t. } z_\emptyset = 1,\ M(T;z) \succeq 0\ (T \subseteq V,\ |T| = t-1),\ z_{ij} = 0\ (ij \in E). \tag{16}$$

Then, $\ell^{(1)} = \vartheta$, and there is still convergence in $\alpha(G)$ steps to the stability number. Analogously, let $\psi^{(t)}$ denote the block-diagonal variation of $\psi^{(t)}_{\text{las}}$ in (13). In view of (15), $\ell^{(t)}$ (or $\psi^{(t)}$) can be computed via an SDP with $\binom{n}{t-1}2^{t-1}$ matrices of order $n+1$ and $O(n^{t+1})$ variables, as compared to one matrix of order $O(n^t)$ and $O(n^{2t})$ variables for the parameter (12).

The parameter $\ell^{(t)}$ $(t = 2, 3)$ has been tested in [19] on the class of Paley graphs P_q; P_q has vertex set the field $\mathbb{F}_q$ on q elements ($q = 1 \mod 4$), with an edge ij iff $i - j$ is a square in $\mathbb{F}_q$. Here are some numerical values: For $q = 401$, $\vartheta \sim 20.02$, $\ell^{(2)} \sim 12.7$, $\ell^{(3)} \sim 10$, $\alpha = 9$; for $q = 809$, $\vartheta \sim 28.4$, $\ell^{(2)} \sim 17.3$, $\ell^{(3)} \sim 13.4$, $\alpha = 11$ (the computation involves symmetry reduction, cf. [16, 19] for details). The bound $\ell^{(2)}$ was also computed for the class of Hamming graphs, enabling to get better upper bounds for the coding problem; details are given in Section 4.

3.4. **SDP hierarchies converging to the chromatic number.** We saw above some construction of SDP bounds for the chromatic number, however these bounds remain below the fractional chromatic number. We now give a 'recipe' for constructing bounds that may go beyond the fractional chromatic number. Following [16, 17], consider the operator Ψ that maps a graph parameter β to the graph parameter Ψ_β defined by

$$\Psi_\beta(G) := \min_{t \in \mathbb{N}} t \text{ s.t. } \beta(G \square K_t) = |V(G)|,$$

where $G_t := G \square K_t$ is the Cartesian product of G and the complete graph K_t. Then, Ψ is monotone non-increasing on the interval $[\frac{|V|}{\chi}, \bar{\chi}]$ and maps it onto the interval $[\omega, \chi]$. In particular, $\Psi_\alpha = \chi$, which is a well known reduction of graph colouring to the stable set problem. Moreover, Ψ maps the whole interval $[\frac{|V|}{\chi}, \alpha]$ onto the single parameter χ and the whole interval $[\frac{|V|}{\omega}, \bar{\chi}]$ onto ω; as an application, no parameter in these intervals can be computed in polynomial time unless P=NP.

An important property of Ψ is that it maps any hierarchy converging to α to a hierarchy converging to the *chromatic number* χ. While the block-diagonal hierarchy $\psi^{(t)}$ remains below χ_f, we have $\Psi_{\ell^{(t)}}(G) = \chi(G)$ for $t \geq |V(G)|$. Moreover,

$$\Psi_{\ell^{(t)}} \geq \psi^{(t)} \geq \frac{|V|}{\ell^{(t)}}.$$

We have tested the parameter $\Psi_{\ell^{(t)}}$ on Kneser graphs. For these graphs it is known that the fractional chromatic number is close to the clique number and thus any bound for the chromatic number which remains below the fractional chromatic number is of no use.

The Kneser graph $K(n,r)$ has vertex set the set of words of weight r in $\{0,1\}^n$, with two words adjacent if their Hamming distance is $2r$. It is known that $\alpha = \vartheta = \binom{n-1}{r-1}$, $\omega = \lfloor n/r \rfloor$, $\chi_f = n/r$,

and $\chi = n - 2r + 2$ [36]. But numerical tests confirm that the parameter $\Psi_{\ell^{(2)}}$ indeed goes beyond the fractional chromatic number (cf. [16, 18]). Here are some numerical values: For $K(12,3)$, $\chi_f = 4 < \Psi_{\ell^{(2)}} = 5 < \Psi_{\ell_+^{(2)}} = 6$; for $K(36,6)$, $\chi_f = 6 < \Psi_{\ell^{(2)}} = 7 < \Psi_{\ell_+^{(2)}} = 9$. (Here $\ell_+^{(2)}$ is the strengthening of $\ell^{(2)}$ obtained by adding non-negativity constraints on the variables.) See the next section for some comments on the computation of $\Psi_{\ell^{(2)}}$ for Kneser graphs.

4. Symmetric reduction and some computational results

We briefly indicate here how symmetry can be used to block-diagonalize SDP's and get more compact programs. (See e.g. [24, 46] for details.) Consider the SDP (2) where the matrices C, A_j are indexed by a set V, $|V| = n$. Assume that (2) is invariant under action of a group $\mathcal{G}$ of permutations of V. That is, if X is feasible for (2) and $\sigma \in \mathcal{G}$, then the matrix $\sigma(X) := (X_{\sigma(i),\sigma(j)})_{i,j\in V}$ is again feasible for (2). As (2) is a convex program, we may restrict w.l.o.g. the matrix variable X to lie in the algebra

$$\mathcal{A}_{\mathcal{G}} := \{X \mid \sigma(X) = X \ \forall \sigma \in \mathcal{G}\}$$

of invariant matrices under action of $\mathcal{G}$. Then, the number of distinct entries in X is at most the dimension d of $\mathcal{A}_{\mathcal{G}}$, which is given by the number of distinct orbits of $V \times V$ under action of $\mathcal{G}$. Thus there is already a gain in the number of variables when the dimension is small. Moreover, one can also reduce the size of the matrices in the SDP. As explained in [25], one can indeed replace the SDP by an equivalent SDP involving $d \times d$ matrices. One can do even better. Indeed the algebra $\mathcal{A}_{\mathcal{G}}$ is a matrix $*$-algebra, which means that it is closed under addition, multiplication, scalar multiplication, and transposition. Hence, by Wedderburn's theorem, it can be *block-diagonalized*. That is, there exists a unitary matrix U and $s, n_1, k_1, \ldots, n_s, k_s \in \mathbb{N}$ such that the set $U^*\mathcal{A}_{\mathcal{G}}U = \{U^*XU \mid X \in \mathcal{A}_{\mathcal{G}}\}$ consists of all the block-diagonal matrices with blocks $A_1 \in \mathbb{C}^{n_1\times n_1}$ (repeated k_1 times), ..., $A_s \in \mathbb{C}^{n_s\times n_s}$ (repeated k_s times), so that $d = \sum_{i=1}^s n_i^2$. In this way one can replace the SDP (2) involving one $n \times n$ matrix X by an equivalent SDP involving s smaller matrices $A_1, \ldots, A_s$. This reduction is particularly interesting when the n_i's are small compared to n.

Let us indicate how this can be applied to compute the block-diagonal bound $\ell^{(2)}$ from (16) for Hamming graphs. Given $\mathcal{D} \subseteq [1,n]$, the Hamming graph $G = H(n,\mathcal{D})$ has vertex set $\{0,1\}^n$, with an edge between two words when their Hamming distance lies in $\mathcal{D}$. Computing the stability number of Hamming graphs is a basic question in coding theory. Hamming graphs have a rich automorphism group $\mathcal{G}$, arising from all permutations of the n coordinates combined with 'switchings' (replacing all words i by $i \oplus i_0$, for a given word i_0).

Consider first the theta number of $H(n,\mathcal{D})$, given by the SDP (4). Then the matrix X is indexed by $\{0,1\}^n$ and is invariant under action of $\mathcal{G}$ precisely when the entries $X_{i,j}$ depend only on the Hamming distance between i,j. The corresponding algebra $\mathcal{A}_{\mathcal{G}}$ is known as the Bose-Mesner algebra, which is commutative with dimension $n+1$. Thus it can be diagonalized, and the theta number of $H(n,\mathcal{D})$ can in fact be computed via an LP of size n, which gives the bound of Delsarte [8].

Consider now the second order bound $\ell^{(2)}(H(n,\mathcal{D}))$ in the block-diagonal hierarchy (16). Note first that in (16) it suffices to consider just *one* of the matrices $M(\{i_0\}; z)$ (for example for the zero word $i_0 = \mathbf{0}$) because the graph $H(n,\mathcal{D})$ is vertex-transitive. Next note that the matrix $M(\{i_0\}; z)$ has the block form

$$\begin{pmatrix} 1 & a^T & b^T \\ a & A & B \\ b & B & B \end{pmatrix} \tag{17}$$

(blocks being indexed by $\emptyset$, $\{0,1\}^n$, and $\{\{\mathbf{0}, i\} \mid i \in \{0,1\}^n\}$). Here, A can be assumed to be invariant under action of $\mathcal{G}$ and thus lies in the Bose-Mesner algebra. Matrix B can be assumed to be invariant under action of the subgroup $\mathcal{G}_0$ of $\mathcal{G}$ consisting of its elements fixing $\mathbf{0}$, i.e. $B_{i,j}$ depends only on the weights of i, j and $i \oplus j$. The corresponding algebra $\mathcal{A}_{\mathcal{G}_0}$ is known as the Terwilliger algebra of the Hamming scheme, whose dimension is equal to $O(n^3)$. Schrijver [44] has computed the explicit block-diagonalization for the Terwilliger algebra. Thus the condition $B \succeq 0$ (of size 2^n) can be replaced by requiring that several blocks be positive semidefinite, of sizes $n_1, \ldots, n_s$ with $\sum_i n_i^2 = O(n^3)$. Thanks to this symmetry reduction, one is able to compute the parameter $\ell^{(2)}(H(n, \mathcal{D}))$ for n up to 28, which gives upper bounds on the stability number of $H(n, \mathcal{D})$, improving the bound of Delsarte (cf. numerical results in [26, 32, 44]).

The computation of the parameter $\Psi_{\ell^{(2)}}$ for Kneser graphs $G = K(n, r)$ deserves a comment. In order to compute $\Psi_{\ell^{(2)}}(G)$, we need to compute $\ell^{(2)}(G \square K_t)$ for several values of t. Again it suffices to consider in (16) *one* matrix $M(\{u\}; z)$ for just one node u of $G \square K_t$. This matrix has again the shape (17), but A and B are now indexed by $V(G \square K_t)$, of cardinality $t|V|$. However, one can exploit the invariance under the permutations of K_t to replace the condition $M(\{u\}; z) \succeq 0$ by a new SDP condition involving four matrices, of sizes $2|V(G)|+1, 2|V(G)|, |V(G)|, |V(G)|$. Thus, the parameter t which determines the size of $M(\{u\}; z)$ comes now as a numerical parameter within these four matrices. Next one can exploit the invariance under the automorphism group of the Kneser graph to further reduce these four matrices (this involves again the Terwilliger algebra). We refer to [16, 18] for details on the symmetry reduction and for computational results.

5. Final remarks

We have sketched how to use semidefinite programming for designing hierarchies of convex relaxations for 0/1 linear programming problems. Note that an analogous construction applies more generally to polynomial optimization problems, where the objective is a polynomial function and the constraints are polynomial equations or inequalities; cf. e.g. the survey [33] for details.

There are many interesting aspects that have been investigated about these hierarchies of SDP relaxations. A first example is the notion of *rank* of a hierarchy, defined as the smallest number of iterations needed to find the original 0/1 polytope. For instance, the rank of the N_+-operator has been studied e.g. for the max-cut, stable set and traveling salesman problems. For instance, the rank of the N_+ operator for the stable set polytope of G is at most $\alpha(G)$ and it can be equal to $\alpha(G)$ (cf. [13]). As another example, it is shown in [30] that the rank of the Lasserre hierarchy for the cut polytope of K_n is at least $\lceil n/2 \rceil$, and equality is conjectured.

An intriguing question is how to use higher order relaxations to get better approximation algorithms. This is by no way easy. As mentioned earlier, for max-cut, one cannot hope to do better than with the order 1 relaxation if the unique game conjecture holds. The following negative result is shown in [10] dealing with LP hierarchies for max-cut: The integrality gap of the basic LP relaxation by triangle inqualities can be as large as $2 - \epsilon$ ($\forall \epsilon > 0$), and this is still true for the LP relaxation obtained after t iterations of the Sherali-Adams procedure, or after adding all valid inequalities for the cut polytope on at most t nodes, for any fixed t. Moreover, it is shown in [43] that after $\Omega_\epsilon(n)$ iterations of the Lovász-Schrijver N operator, one gets an LP relaxation with the same duality gap $2 - \epsilon$. If one recalls that the basic order 1 SDP relaxation for max-cut permits to achieve the Goemans-Williamson approximation ratio 1/0.878, then this is a good illustration of the strength of the positive semidefinite condition.

On the positive side, the authors of [5] are able to exploit higher order relaxations in the Lasserre hierarchy for the maximum independent set in a 3-uniform hypergraph. Namely, for fixed $\gamma > 0$,

if there is an independent set of size at least γn, then they can construct one of size $n^{\Omega(\gamma^2)}$ using the relaxation of order $\Theta(1/\gamma^2)$.

One may also want to exploit the structure of the problem at hand, like sparsity, in order to get more economical relaxations. For instance, when G has small tree-width, one can design sparse relaxations leading to a compact (LP or SDP) representation of the problem (cf. [33, Sec. 8.2], [48]).

References

[1] N. Alon and A. Naor. Approximating the cut-norm via Grothendieck's inequality. *SIAM Journal on Computing* **35**:787–803, 2006.

[2] N. Alon, K. Makarychev, Y. Makarychev, and A. Naor. Quadratic forms on graphs. *Inventiones Mathematicae* **163**(3):499-522, 2005.

[3] F. Barahona and A.R. Mahjoub. On the cut polytope. *Mathematical Programming*, **36**:157–173, 1986.

[4] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Volume 15 of Studies in Applied Mathematics, SIAM, 1994.

[5] E. Chlamtac and G. Singh. Improved approximation guarantees through higher levels of SDP hierarchies. APPROX 2008.

[6] M. Chudnovsky, N.Robertson, P.Seymour, and R.Thomas. The strong perfect graph theorem. *Annals of Mathematics* **164**:51–229, 2006.

[7] A. Coja-Oghlan. The Lovász number of random graphs. *Combinatorics, Probability and Computing* **14**:439–465, 2005.

[8] P. Delsarte. *An Algebraic Approach to the Association Schemes of Coding Theory*. [Philips Research Reports Supplements (1973) No. 10] Philips Research Laboratories, Eindhoven, 1973.

[9] U. Feige. Randomized graph products, chromatic numbers, and the Lovász ϑ-function. *Combinatorica* 17:79–90, 1997.

[10] W. Fernandez de la Vega and C. Kenyon-Mathieu. Linear programming relaxations of maxcut. SODA 2007, pages 53–61.

[11] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco, W.H. Freeman & Company, Publishers, 1979.

[12] M.X. Goemans. Semidefinite Programming in Combinatorial Optimization. *Mathematical Programming* **79**:143–161, 1997.

[13] M.X. Goemans and L. Tunçel. When does the positive semidefiniteness constraint help in lifting procedures? *Mathematics of Operations Research* **26**:796–815, 2001.

[14] M.X. Goemans and D. Williamson. Improved approximation algorithms for maximum cuts and satisfiability problems using semidefinite programming. *Journal of the ACM* **42**:1115–1145, 1995.

[15] M. Grötschel, L. Lovász and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, Springer, 1988.

[16] N. Gvozdenović. *Approximating the Stability Number and the Chromatic Number of a Graph via Semidefinite Programming*. Ph.D. thesis, Univ. Amsterdam, 2008.

[17] N. Gvozdenović and M. Laurent. The operator Ψ for the Chromatic Number of a Graph. *SIAM Journal on Optimization* **19**(2):572-591, 2008.

[18] N. Gvozdenović and M. Laurent. Computing semidefinite programming lower bounds for the (fractional) chromatic number via block-diagonalization. *SIAM Journal on Optimization* **19(2)**:592–615, 2008.

[19] N. Gvozdenović, M. Laurent and F. Vallentin. Block-diagonal semidefinite programming hierarchies for 0/1 programming. *Operations Research Letters* **37**:27–31, 2009.

[20] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, ACM, New York, pp. 1–10, 1997.

[21] D. Karger, R. Motwani, and M. Sudan. Approximate graph colouring by semidefinite programming. *Journal of the Association for Computing Machinery* **45**:246–265, 1998.

[22] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, ACM, New York, pp. 767–775, 2002.

[23] E. de Klerk. *Aspects of Semidefinite Programming - Interior Point Algorithms and Selected Applications*. Kluwer, 2002.

[24] E. de Klerk. Exploiting special structure in semidefinite programming: a survey of theory and applications. Preprint, 2008.Available at `http://www.optimization-online.org/DB_HTML/2008/0/2107.html`

[25] E. de Klerk, D. Pasechnik, A. Schrijver. Reduction of symmetric semidefinite programs using the regular *-representation. *Mathematical Programming*, **109**(2-3):613–624, 2007.
[26] E. De Klerk, D.V. Pasechnik. A note on the stability number of an orthogonality graph. *European Journal of Combinatorics* **28**:1971–1979, 2007.
[27] D.E. Knuth. The sanwich theorem. *Electronic Journal of Combinatorics* **1**:1–48, 1994.
[28] J.B. Lasserre. An explicit exact SDP relaxation for nonlinear $0-1$ programs. In K. Aardal and A.M.H. Gerards, eds., *Lecture Notes in Computer Science* **2081**:293–303, 2001.
[29] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research* **28(3)**:470–496, 2003.
[30] M. Laurent. Lower bound for the number of iterations in semidefinite relaxations for the cut polytope. *Mathematics of Operations Research*, **28**(4):871–883, 2003.
[31] M. Laurent. Semidefinite relaxations for Max-Cut. In *The Sharpest Cut: The Impact of Manfred Padberg and His Work*. M. Grötschel (ed.), pages 257-290, *MPS-SIAM Series in Optimization* **4**, 2004.
[32] M. Laurent. Strengthened semidefinite programming bounds for codes. *Mathematical Programming* **109(2-3)**:239-261, 2007.
[33] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging Applications of Algebraic Geometry*, Vol. **149** of *The IMA Volumes in Mathematics and its Applications*, M. Putinar and S. Sullivant (eds.), Springer, pages 157-270, 2009.
[34] M. Laurent and F. Rendl. Semidefinite Programming and Integer Programming. In *Handbook on Discrete Optimization*, K. Aardal, G. Nemhauser, R. Weismantel (eds.), pp. 393-514, Elsevier B.V., 2005.
[35] L. Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics* **2**:253–267, 1972.
[36] L. Lovász. Kneser's conjecture, chromatic number and homotopy. *Journal of Combinatorial Theory, Series A* **25**(3):319–324, 1978.
[37] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory* **IT-25**:1–7, 1979.
[38] L. Lovász. Semidefinite programs and combinatorial optimization. In *Recent Advances in Algorithms and Combinatorics*, B.A. Reed and C.L. Sales (eds.), Springer, 2003.
[39] L. Lovász and A. Schrijver. Cones of matrices and set-functions and $0-1$ optimization. *SIAM Journal on Optimization* **1**:166–190, 1991.
[40] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software* **9**:141–160, 1998.
[41] S. Poljak and Zs. Tuza. On the expected relative error of the polyhedral approximation of the max-cut. *Operations Research Letters* **16**:191–198, 1994.
[42] J.A. Reeds. A new lower bound on the real Grothendieck constant, 1991. Avalaible at `http://www.dtc.umn.edu/~reedsj/bound2.dvi`
[43] G. Schoenebeck, L. Trevisan, and M. Tulsiani. Tight integrality gaps for Lovász-Schrijver LP relaxations of vertex cover and max cut. ECCC, Report 132, 2006.
[44] A. Schrijver. New code upper bounds from the Terwilliger algebra and semidefinite programming. *IEEE Transactions on Information Theory* **51**:2859–2866, 2005.
[45] H.D. Sherali and W.P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics* **3**:411–430, 1990.
[46] F. Vallentin. Symmetry in semidefinite programs. *Linear Algebra and its Applications* **430**:360–369, 2009.
[47] L. Vandenberghe and S. Boyd. Semidefinite Programming. *SIAM Review* **38(1)**:49–95, 1996.
[48] M.J. Wainwright and M.I. Jordan. Treewidth-based conditions for exactness of the Sherali-Adams and Lasserre relaxations. Univ. California, Berkeley, Technical Report 671, 2004.
[49] H. Wolkowicz, R. Saigal, L. Vandeberghe (eds.). *Handbook of Semidefinite Programming*, Kluwer, 2000.

CENTRUM VOOR WISKUNDE EN INFORMATICA (CWI), KRUISLAAN 413, 1098 SJ AMSTERDAM, THE NETHERLANDS
E-mail address: `monique@cwi.nl`

Practising logic through the web

Freek Wiedijk

Institute for Computing and Information Sciences
Radboud University Nijmegen
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands

Abstract. We present the ProofWeb system for practising natural deduction in predicate logic and for remotely working with the Coq proof assistant. The ProofWeb system can be used for free, both for trying it out as a guest as well as for hosting computer labs for logic and proof assistant courses, on the Nijmegen ProofWeb server `http://proofweb.cs.ru.nl/`.

A computer lab for a logic course

Suppose you are a lecturer at a computer science department who has to teach the introductory logic course. That course teaches natural deduction in three standard logics:

- Propositional logic
- Predicate logic
- Predicate logic with equality

You would like the computer to be used by your students for their logic exercises. What should you use?

In the Netherlands many of these courses currently use the Jape system from the UK [3]. It allows students to check simple deductions with a computer program that has a pleasant point-and-click interface. However, the Jape system has restrictions. It is not possible to save an unfinished deduction and later continue working on it. It is not possible to have types for variables. It is not possible to reason about time in the deductions, or to reason arithmetically in other contexts for that matter. This all makes the use of Jape inconvenient for larger examples. Also, with Jape there is no easy way to run a course in an centrally organized manner. Each student has to install Jape on their own system, it is hard to keep track of what the students are doing with it, and grading students' work is labor intensive.

Here we present the ProofWeb system, as a better system to be used for computer labs for logic courses. ProofWeb was developed as a joint effort between the Radboud University Nijmegen and the Free University Amsterdam, in a small education innovation project funded by the Surf Foundation called *Web-deductie voor het onderwijs in formeel denken.* ProofWeb already has been, and still is, used in about a dozen logic courses in various universities.

The Coq proof assistant as a web application

Proof assistants are systems for developing and checking deductions on the computer. These systems are both used for verification of the correctness of software and hardware, as well as for checking the correctness of mathematics. There are several proof assistants to choose from (important examples are PVS, HOL, Isabelle, ACL2, The B Method, Twelf, and in some sense even Jape). One of the best proof assistants currently available is the Coq system developed at INRIA in France [4, 1]. Coq has been used for many impressive projects, like the validated Compcert C compiler which compiles a large subset of C to assembly for the PowerPC family of processors, getting a performance similar to the gcc compiler with the first level of optimization turned on [10, 2], and the formal proof of the Four Color Theorem by Georges Gonthier [6].

Before the ProofWeb project, Henk Barendregt had for a long time already been asking for a web interface to the Coq system, to allow people to work with Coq without first having to go through the trouble of installing it on their computer. There had been a project in which mathematicians from Russia were supposed to contribute (they would be paid a fixed price for each lemma proved), but before they had Coq running correctly on their systems the project already was over. If a Coq environment had been available on the web this might have gone differently. But Henk wanted a web interface to Coq for a more ambitious goal: to have the whole world help with encoding all of mathematics in Coq. This would be an important step to make that come about.

Cezary Kaliszyk, one of the PhD students in Henk's group in Nijmegen, happened to be an expert on Web interfaces. His PhD research was on making proof assistants more friendly, but his master's thesis had been about Web interfaces. When he heard about Henk's wish, in a Christmas holiday he whipped together a simple but nice web interface to the Coq system. This interface later was extended into the ProofWeb system. Although various people were involved in that project, Cezary remained the sole developer of the system.

The ProofWeb system

The ProofWeb system has the following distinctive properties:

- With ProofWeb the students work with the Coq proof assistant. Their input is not being pre-processed by the ProofWeb interface; i.e., the text that ProofWeb users are typing are actual Coq proof scripts. With ProofWeb the students are working with an industrial strength interactive theorem prover, with the full power of that system available to them from the start.
- ProofWeb shows deductions in a style that matches as closely as possible the deductions the way they are presented in elementary logic courses. Although ProofWeb shows the Coq presentation of the state of the proof, there is also a display that really looks like the diagrams from logic textbooks. In fact, the ProofWeb system intentionally was made fully compatible with a good logic textbook: *Logic in Computer Science: Modelling and Reasoning about*

Systems by Michael Huth and Mark Ryan [7]. If a course uses this book then ProofWeb is a good choice for the lab work of that course. And if a course wants to use ProofWeb for the lab work, then this book is a good textbook to be used for the non-computer part of the course.

- ProofWeb comes with an extensive manual [8] that both summarizes natural deduction for predicate logic and presents all the details of working with the ProofWeb system. This manual can be downloaded as a PDF file from the ProofWeb web site.
- ProofWeb comes with a collection of more than a hundred simple logic exercises to be worked with the ProofWeb system. Courses can have their own set of exercises, but this set is a good default choice for an introductory logic course.
- To use ProofWeb there is no need to install special software. In fact ProofWeb does not even use a plug-in. All that one needs is a compatible web browser. ProofWeb users can access ProofWeb from anywhere on the internet. (For example, if necessary, students might go into an internet café to finish their homework.)
- ProofWeb has good interfaces for both students and teachers to manage their courses. Students will see a list of exercises, and the status of these exercises. Teachers will see a list of students, and the status of those students.

ProofWeb for teaching Coq

ProofWeb can be used for teaching logic to undergraduate students, but it also can be used for teaching proof assistants to graduate students. In fact, about half of the current ProofWeb courses are type theory courses that teach the Coq system.

In such courses the exercises do not have the shape of having to prove a single first order formula. Instead they are long files with many lemmas, where the students have to fill in the proofs of those lemmas. Generally students then will have to complete one such an exercise per week.

Fitch-style natural deduction

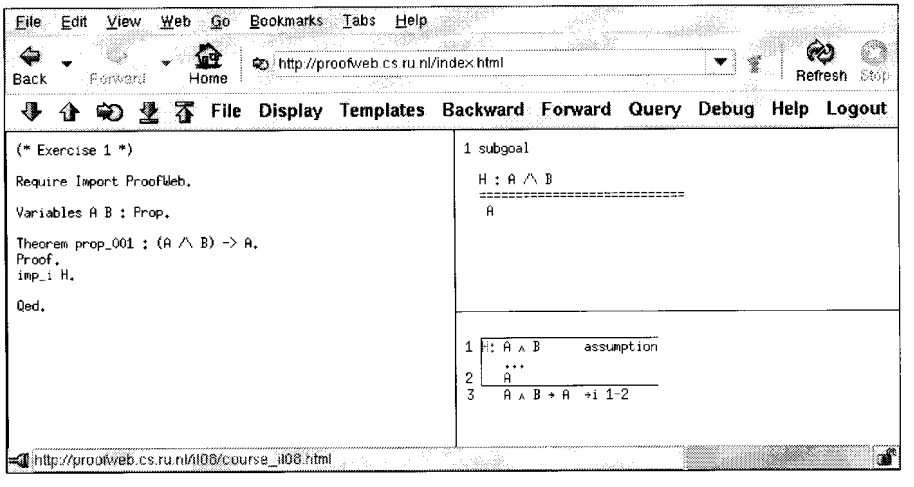
There are two styles of presenting natural deduction proofs on paper. The more commonly taught style was introduced by Frederic Fitch. These proofs consists of lines that are grouped together either by boxes around them or by having 'flags' with assumptions in the margin. (Using flags is typical of the way this kind of proof is taught at the Technical University of Eindhoven. This style originated in the Automath project from the sixties and seventies.)

Fitch-style proofs are the style of natural deduction presented in the book by Huth and Ryan that ProofWeb especially was designed to match. However, Fitch-style proofs do not exactly have the structure of Coq proofs. For this reason, in the ProofWeb project one of the major difficulties was to get Fitch-style deduction to work well with the Coq system [9].

Here is a Fitch-style proof as shown in ProofWeb:

1	H1:	∃x, (P x ∨ ¬Q a)	assumption
2	H2:	Q a	assumption
	b		
3	H3:	P b ∨ ¬Q a	assumption
4	H4:	P b	assumption
5		∃x, P x	∃i 4
6	H5:	¬Q a	assumption
7		⊥	¬e 6,2
8		∃x, P x	⊥e 7
9		∃x, P x	∨e 3,4-5,6-8
10		∃x, P x	∃e 1,3-9
11		Q a → ∃x, P x	→i 2-10
12		∃x, (P x ∨ ¬Q a) → Q a → ∃x, P x	→i 1-11

When doing the proof, this picture will be growing in the lower-right pane of the ProofWeb window. Here is what the ProofWeb interface looks like halfway down a simpler example:



Both the Coq view of the proof as well as the textbook display of the proof are present simultaneously. To our surprise when working on exercises like this, students mostly completely ignored the Coq proof state in the upper-right pane and just looked at the lower-right.

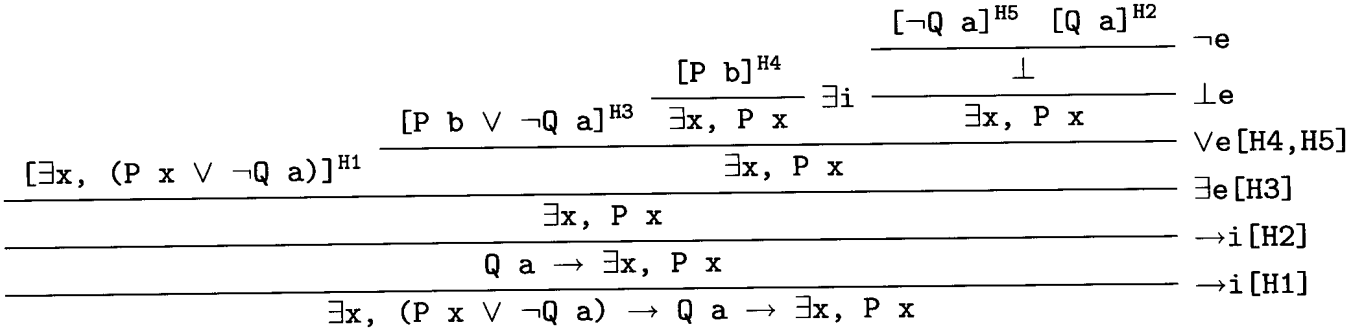
Gentzen-style natural deduction

The other style of natural deduction that is commonly used was introduced by Gerhard Gentzen. This is the style that is used in the textbook *Logic and Structure* by Dirk van Dalen [5]. (ProofWeb is not completely compatible with this book – we decided that internal consistency was important, so we had our Gentzen-style proof display be influenced by the Huth and Ryan conventions – but it comes very close.)

In the case of Gentzen-style natural deduction the proofs look like trees, growing upward from the statement that is proved. When doing this kind of

proof on paper one generally gets space problems fast, but in a computer one does not have this problem, as in that case one has scroll bars.

Here is the same example proof, but this time presented in Gentzen style:



The ProofWeb code that generated both proofs (which just are different displays of the same proof) was:

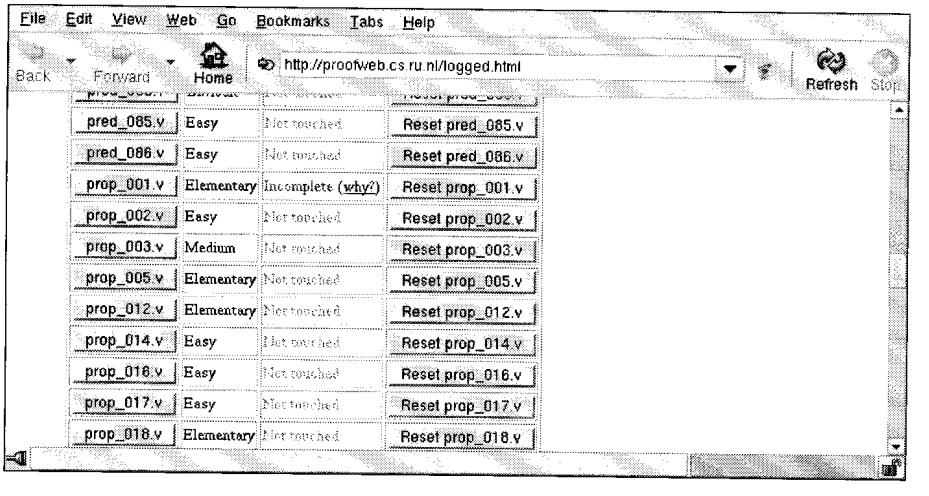
```
Require Import ProofWeb.

Variable P Q : D -> Prop.
Variable a : D.

Theorem example :
  exi x, (P(x) \/ ~Q(a)) -> Q(a) -> exi x, P(x).
Proof.
imp_i H1.
imp_i H2.
exi_e (exi x, (P(x) \/ ~Q(a))) b H3.
exact H1.
dis_e (P(b) \/ ~Q(a)) H4 H5.
exact H3.
exi_i b.
exact H4.
fls_e.
neg_e (Q(a)).
exact H5.
exact H2.
Qed.
```

This is an actual Coq input script. The commands occurring in this script can be selected from menu's in the ProofWeb interface, so the students do not need to know these commands by heart. Also, these commands are extensively explained in the ProofWeb manual.

The student's view of ProofWeb

When a student follows a course that uses the ProofWeb system, he or she will go to the ProofWeb server that hosts the course. The student will then select the

course from a menu, login in to the system by entering username and password, and then will be presented with the list of exercises:



Each exercise will have four possible statuses, which are color coded:

Gray	Not touched
Red	Incomplete
Orange	Correct
Green	Solved

The goal for the student will be to work the exercises until the traffic lights in his list are all green. (The orange status means that the proof is correct, but that the student used proof steps that are not allowed for the course. For example Coq's powerful automation will not be allowed when doing exercises in elementary logic.)

The teacher's view of ProofWeb

The teacher can login for the course too, but with the teacher's interface. He or she then will be presented with a table that lists all the students, with for each student a count of how many of that student's exercises there are of each of the four different colors. The teacher also can login as if he were a specific student, to grade exercises or maybe help the student with finding a solution.

Apart from this, the teacher interface also allows the teacher to add student logins or change passwords. Finally there is a button for downloading all files for the course as a big tar file, for archival purposes at the end of a course.

The MathWiki project

ProofWeb was primarily developed for logic education, but in Nijmegen we have more ambitious goals for it.

Recently, the project *MathWiki: a Web-based Collaborative Authoring Environment for Formal Proofs* was funded in NWO's vrije competitie, to develop a

system to be called MathWiki. This system will be a cross between a Wikipedia for mathematics, and the system that Henk was dreaming of where all the world would help build a Coq version of all of mathematics. An aspect of the MathWiki project is that it is not supposed to be just about the Coq proof assistant. Eventually many proof assistants will be available through our interface.

A prototype of MathWiki already was developed and a web page about this work can be found on the ProofWeb server.

Trying it?

If you want to try ProofWeb: it is completely free. Currently there are three way of using the system:

- First you can access it as a guest user. For this you do not even need to register. Just click the Guest login button. It is probably useful to first look through the ProofWeb manual to know what to do next.
- Second you can host courses on the Nijmegen ProofWeb server. For this, just send an email message to `proofweb@cs.ru.nl`.
- Third, you might not trust someone else with your students' data. In that case you might download the ProofWeb server and install it on a machine of your own. At the moment this has not been done much, and you will probably need some help from Nijmegen with that, which of course we will be happy to provide.

We hope ProofWeb will be useful both for logic teaching, as well as for exposing more students to proof assistants. If you have any questions about ProofWeb, just send mail to

`proofweb@cs.ru.nl`

Or you should surf to

`http://proofweb.cs.ru.nl/`

and take a look for yourself at the system that we developed.

References

1. Yves Bertot and Pierre Castéran. *Coq'Art: The Calculus of Inductive Constructions.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.
2. Sandrine Blazy, Zaynah Dargaye, and Xavier Leroy. Formal Verification of a C Compiler Front-end. In *FM 2006: Int. Symp. on Formal Methods*, volume 4085 of *Lecture Notes in Computer Science*, pages 460–475. Springer, 2006.
3. Richard Bornat and Bernard Sufrin. Jape's Quiet Interface. In N. Merriam, editor, *User Interfaces for Theorem Provers (UITP '96)*, Technical Report, pages 25–34. University of York, 1996.
4. Coq Development Team. *The Coq Proof Assistant Reference Manual*, 2008.
5. Dirk van Dalen. *Logic and Structure.* Springer, 4th edition, 2004.

6. Georges Gonthier. A computer-checked proof of the Four Colour Theorem. ⟨`http://research.microsoft.com/~gonthier/4colproof.pdf`⟩, 2006.
7. Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems.* Cambridge University Press, 2nd edition, 2004.
8. Cezary Kaliszyk, Femke van Raamsdonk, Freek Wiedijk, Hanno Wupper, Maxim Hendriks, and Roel de Vrijer. Deduction using the ProofWeb system. Technical Report ICIS-R08016, Institute for Computing and Information Sciences, University of Nijmegen, 2008.
9. Cezary Kaliszyk and Freek Wiedijk. Merging procedural and declarative proof. To be published in the proceedings of TYPES 2008, 2009.
10. Xavier Leroy. Formal Certification of a Compiler Back-end, or: Programming a Compiler with a Proof Assistant. In *POPL'06*, Charleston, South Carolina, USA, 2006.

Statuten

Artikel 1.
1. De vereniging draagt de naam: "Nederlandse Vereniging voor Theoretische Informatica".
2. Zij heeft haar zetel te Amsterdam.
3. De vereniging is aangegaan voor onbepaalde tijd.
4. De vereniging stelt zich ten doel de theoretische informatica te bevorderen haar beoefening en haar toepassingen aan te moedigen.

Artikel 2.
De vereniging kent gewone leden en ereleden. Ereleden worden benoemd door het bestuur.

Artikel 3.
De vereniging kan niet worden ontbonden dan met toestemming van tenminste drievierde van het aantal gewone leden.

Artikel 4.
Het verenigingsjaar is het kalenderjaar.

Artikel 5.
De vereniging tracht het doel omschreven in artikel 1 te bereiken door
a. het houden van wetenschappelijke vergaderingen en het organiseren van symposia en congressen;
b. het uitgeven van een of meer tijdschriften, waaronder een nieuwsbrief of vergelijkbaar informatiemedium;
c. en verder door alle zodanige wettige middelen als in enige algemene vergadering goedgevonden zal worden.

Artikel 6.
1. Het bestuur schrijft de in artikel 5.a bedoelde bijeenkomsten uit en stelt het programma van elk van deze bijeenkomsten samen.
2. De redacties der tijdschriften als bedoeld in artikel 5.b worden door het bestuur benoemd.

Artikel 7.
Iedere natuurlijke persoon kan lid van de vereniging worden. Instellingen hebben geen stemrecht.

Artikel 8.
Indien enig lid niet langer als zodanig wenst te worden beschouwd, dient hij de ledenadministratie van de vereniging daarvan kennis te geven.

Artikel 9.
Ieder lid ontvangt een exemplaar der statuten, opgenomen in de nieuwsbrief van de vereniging. Een exemplaar van de statuten kan ook opgevraagd worden bij de secretaris. Ieder lid ontvangt de tijdschriften als bedoeld in artikel 5.b.

Artikel 10.
Het bestuur bestaat uit tenminste zes personen die direct door de jaarvergadering worden gekozen, voor een periode van drie jaar. Het bestuur heeft het recht het precieze aantal bestuursleden te bepalen. Bij de samenstelling van het bestuur dient rekening gehouden te worden met de wenselijkheid dat vertegenwoordigers van de verschillende werkgebieden van de theoretische informatica in Nederland in het bestuur worden opgenomen. Het bestuur kiest uit zijn midden de voorzitter, secretaris en penningmeester.

Artikel 11.
Eens per drie jaar vindt een verkiezing plaats van het bestuur door de jaarver-

gadering. De door de jaarvergadering gekozen bestuursleden hebben een zittingsduur van maximaal twee maal drie jaar. Na deze periode zijn zij niet terstond herkiesbaar, met uitzondering van secretaris en penningmeester. De voorzitter wordt gekozen voor de tijd van drie jaar en is na afloop van zijn ambtstermijn niet onmiddellijk als zodanig herkiesbaar. In zijn functie als bestuurslid blijft het in de vorige alinea bepaalde van kracht.

Artikel 12.

Het bestuur stelt de kandidaten voor voor eventuele vacatures. Kandidaten kunnen ook voorgesteld worden door gewone leden, minstens een maand voor de jaarvergadering via de secretaris. Dit dient schriftelijk te gebeuren op voordracht van tenminste vijftien leden. In het geval dat het aantal kandidaten gelijk is aan het aantal vacatures worden de gestelde kandidaten door de jaarvergadering in het bestuur gekozen geacht. Indien het aantal kandidaten groter is dan het aantal vacatures wordt op de jaarvergadering door schriftelijke stemming beslist. Ieder aanwezig lid brengt een stem uit op evenveel kandidaten als er vacatures zijn. Van de zo ontstane rangschikking worden de kandidaten met de meeste punten verkozen, tot het aantal vacatures. Hierbij geldt voor de jaarvergadering een quorum van dertig. In het geval dat het aantal aanwezige leden op de jaarvergadering onder het quorum ligt, kiest het zittende bestuur de nieuwe leden. Bij gelijk aantal stemmen geeft de stem van de voorzitter (of indien niet aanwezig, van de secretaris) de doorslag.

Artikel 13.

Het bestuur bepaalt elk jaar het precieze aantal bestuursleden, mits in overeenstemming met artikel 10. In het geval van aftreden of uitbreiding wordt de zo ontstane vacature aangekondigd via mailing of nieuwsbrief, minstens twee maanden voor de eerstvolgende jaarvergadering. Kandidaten voor de ontstane vacatures worden voorgesteld door bestuur en gewone leden zoals bepaald in artikel 12. Bij aftreden van bestuursleden in eerste of tweede jaar van de driejarige cyclus worden de vacatures vervuld op de eerstvolgende jaarvergadering. Bij aftreden in het derde jaar vindt vervulling van de vacatures plaats tegelijk met de algemene driejaarlijkse bestuursverkiezing. Voorts kan het bestuur beslissen om vervanging van een aftredend bestuurslid te laten vervullen tot de eerstvolgende jaarvergadering. Bij uitbreiding van het bestuur in het eerste of tweede jaar van de cyclus worden de vacatures vervuld op de eerstvolgende jaarvergadering. Bij uitbreiding in het derde jaar vindt vervulling van de vacatures plaats tegelijk met de driejaarlijkse bestuursverkiezing. Bij inkrimping stelt het bestuur vast welke leden van het bestuur zullen aftreden.

Artikel 14.

De voorzitter, de secretaris en de penningmeester vormen samen het dagelijks bestuur. De voorzitter leidt alle vergaderingen. Bij afwezigheid wordt hij vervangen door de secretaris en indien ook deze afwezig is door het in jaren oudste aanwezig lid van het bestuur. De secretaris is belast met het houden der notulen van alle huishoudelijke vergaderingen en met het voeren der correspondentie.

Artikel 15.

Het bestuur vergadert zo vaak als de voorzitter dit nodig acht of dit door drie zijner leden wordt gewenst.

Artikel 16.

Minstens eenmaal per jaar wordt door het bestuur een algemene vergadering bijeengeroepen; één van deze vergaderingen wordt expliciet aangeduid met de naam van jaarvergadering; deze vindt plaats op een door het bestuur te bepalen

dag en plaats.

Artikel 17.

De jaarvergadering zal steeds gekoppeld zijn aan een wetenschappelijk symposium. De op het algemene gedeelte vaan de jaarvergadering te behandelen onderwerpen zijn

a. Verslag door de secretaris;

b. Rekening en verantwoording van de penningmeester;

c. Verslagen van de redacties der door de vereniging uitgegeven tijdschriften;

d. Eventuele verkiezing van bestuursleden;

e. Wat verder ter tafel komt. Het bestuur is verplicht een bepaald punt op de agenda van een algemene vergadering te plaatsen indien uiterlijk vier weken van te voren tenminste vijftien gewone leden schriftelijk de wens daartoe aan het bestuur te kennen geven.

Artikel 18.

Deze statuten kunnen slechts worden gewijzigd, nadat op een algemene vergadering een commissie voor statutenwijziging is benoemd. Deze commissie doet binnen zes maanden haar voorstellen via het bestuur aan de leden toekomen. Gedurende drie maanden daarna kunnen amendementen schriftelijk worden ingediend bij het bestuur, dat deze ter kennis van de gewone leden brengt, waarna een algemene vergadering de voorstellen en de ingediende amendementen behandelt. Ter vergadering kunnen nieuwe amendementen in behandeling worden genomen, die betrekking hebben op de voorstellen van de commissie of de schriftelijk ingediende amendementen. Eerst wordt over elk der amendementen afzonderlijk gestemd; een amendement kan worden aangenomen met gewone meerderheid van stemmen. Het al dan niet geamendeerde voorstel wordt daarna in zijn geheel in stemming gebracht, tenzij de vergadering met gewone meerderheid van stemmen besluit tot afzonderlijke stemming over bepaalde artikelen, waarna de resterende artikelen in hun geheel in stemming gebracht worden. In beide gevallen kunnen de voorgestelde wijzigingen slechts worden aangenomen met een meerderheid van tweederde van het aantal uitgebrachte stemmen. Aangenomen statutenwijzigingen treden onmiddellijk in werking.

Artikel 19.

Op een vergadering worden besluiten genomen bij gewone meerderheid van stemmen, tenzij deze statuten anders bepalen. Elk aanwezig gewoon lid heeft daarbij het recht een stem uit te brengen. Stemming over zaken geschiedt mondeling of schriftelijk, die over personen met gesloten briefjes. Uitsluitend bij schriftelijke stemmingen worden blanco stemmen gerekend geldig te zijn uitgebracht.

Artikel 20.

a. De jaarvergadering geeft bij huishoudelijk reglement nadere regels omtrent alle onderwerpen, waarvan de regeling door de statuten wordt vereist, of de jaarvergadering gewenst voorkomt.

b. Het huishoudelijk reglement zal geen bepalingen mogen bevatten die afwijken van of die in strijd zijn met de bepalingen van de wet of van de statuten, tenzij de afwijking door de wet of de statuten wordt toegestaan.

Artikel 21.

In gevallen waarin deze statuten niet voorzien, beslist het bestuur.