Bruno Buchberger
John A. Campbell (Eds.)

LNAI 3249

Artificial Intelligence and Symbolic Computation

7th International Conference, AISC 2004
Linz, Austria, September 2004
Proceedings

Springer

Bruno Buchberger John A. Campbell (Eds.)

Artificial Intelligence and Symbolic Computation

7th International Conference, AISC 2004
Linz, Austria, September 22-24, 2004
Proceedings

Springer

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

Volume Editors

Bruno Buchberger
Johannes Kepler University
Research Institute for Symbolic Computation (RISC-Linz)
4040 Linz, Austria
E-mail: Buchberger@RISC.Uni-Linz.ac.at

John A. Campbell
University College London, Department of Computer Science
Gower Street, London WC1E 6BT, UK
E-mail: j.campbell@cs.ucl.ac.uk

Library of Congress Control Number: 2004112218

CR Subject Classification (1998): I.2.1-4, I.1, G.1-2, F.4.1

ISSN 0302-9743
ISBN 3-540-23212-5 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11318781 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3249
Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence (LNAI)

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

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

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

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

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

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

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

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

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

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

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

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004.

Vol. 3127: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.), Conceptual Structures at Work. XI, 403 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004.

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004.

Vol. 3097: D. Basin, M. Rusinowitch (Eds.), Automated Reasoning. XII, 493 pages. 2004.

Vol. 3071: A. Omicini, P. Petta, J. Pitt (Eds.), Engineering Societies in the Agents World. XIII, 409 pages. 2004.

Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004. XXV, 1208 pages. 2004.

Vol. 3068: E. André, L. Dybkjær, W. Minker, P. Heisterkamp (Eds.), Affective Dialogue Systems. XII, 324 pages. 2004.

Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), Programming Multi-Agent Systems. X, 221 pages. 2004.

Vol. 3066: S. Tsumoto, R. Słowiński, J. Komorowski, J.W. Grzymała-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004.

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic in Computer Science. X, 275 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004.

Vol. 3055: H. Christiansen, M.-S. Hacid, T. Andreasen, H.L. Larsen (Eds.), Flexible Query Answering Systems. X, 500 pages. 2004.

Vol. 3040: R. Conejo, M. Urretavizcaya, J.-L. Pérez-de-la-Cruz (Eds.), Current Topics in Artificial Intelligence. XIV, 689 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004.

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004.

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004.

Vol. 3020: D. Polani, B. Browning, A. Bonarini, K. Yoshida (Eds.), RoboCup 2003: Robot Soccer World Cup VII. XVI, 767 pages. 2004.

Vol. 3012: K. Kurumatani, S.-H. Chen, A. Ohuchi (Eds.), Multi-Agnets for Mass User Support. X, 217 pages. 2004.

Vol. 3010: K.R. Apt, F. Fages, F. Rossi, P. Szeredi, J. Váncza (Eds.), Recent Advances in Constraints. VIII, 285 pages. 2004.

Vol. 2990: J. Leite, A. Omicini, L. Sterling, P. Torroni (Eds.), Declarative Agent Languages and Technologies. XII, 281 pages. 2004.

Vol. 2980: A. Blackwell, K. Marriott, A. Shimojima (Eds.), Diagrammatic Representation and Inference. XV, 448 pages. 2004.

Vol. 2977: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (Eds.), Engineering Self-Organising Systems. X, 299 pages. 2004.

Vol. 2972: R. Monroy, G. Arroyo-Figueroa, L.E. Sucar, H. Sossa (Eds.), MICAI 2004: Advances in Artificial Intelligence. XVII, 923 pages. 2004.

Vol. 2969: M. Nickles, M. Rovatsos, G. Weiss (Eds.), Agents and Computational Autonomy. X, 275 pages. 2004.

Vol. 2961: P. Eklund (Ed.), Concept Lattices. IX, 411 pages. 2004.

Vol. 2953: K. Konrad, Model Generation for Natural Language Interpretation and Analysis. XIII, 166 pages. 2004.

Vol. 2934: G. Lindemann, D. Moldt, M. Paolucci (Eds.), Regulated Agent-Based Social Systems. X, 301 pages. 2004.

Vol. 2930: F. Winkler (Ed.), Automated Deduction in Geometry. VII, 231 pages. 2004.

Vol. 2926: L. van Elst, V. Dignum, A. Abecker (Eds.), Agent-Mediated Knowledge Management. XI, 428 pages. 2004.

Vol. 2923: V. Lifschitz, I. Niemelä (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 365 pages. 2004.

Vol. 2915: A. Camurri, G. Volpe (Eds.), Gesture-Based Communication in Human-Computer Interaction. XIII, 558 pages. 2004.

Vol. 2913: T.M. Pinkston, V.K. Prasanna (Eds.), High Performance Computing - HiPC 2003. XX, 512 pages. 2003.

Vol. 2903: T.D. Gedeon, L.C.C. Fung (Eds.), AI 2003: Advances in Artificial Intelligence. XVI, 1075 pages. 2003.

Vol. 2902: F.M. Pires, S.P. Abreu (Eds.), Progress in Artificial Intelligence. XV, 504 pages. 2003.

Vol. 2892: F. Dau, The Logic System of Concept Graphs with Negation. XI, 213 pages. 2003.

Vol. 2891: J. Lee, M. Barley (Eds.), Intelligent Agents and Multi-Agent Systems. X, 215 pages. 2003.

Vol. 2882: D. Veit, Matchmaking in Electronic Markets. XV, 180 pages. 2003.

Vol. 2871: N. Zhong, Z.W. Raś, S. Tsumoto, E. Suzuki (Eds.), Foundations of Intelligent Systems. XV, 697 pages. 2003.

Vol. 2854: J. Hoffmann, Utilizing Problem Structure in Planing. XIII, 251 pages. 2003.

Vol. 2843: G. Grieser, Y. Tanaka, A. Yamamoto (Eds.), Discovery Science. XII, 504 pages. 2003.

Vol. 2842: R. Gavaldá, K.P. Jantke, E. Takimoto (Eds.), Algorithmic Learning Theory. XI, 313 pages. 2003.

Vol. 2838: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Knowledge Discovery in Databases: PKDD 2003. XVI, 508 pages. 2003.

Vol. 2837: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Machine Learning: ECML 2003. XVI, 504 pages. 2003.

Vol. 2835: T. Horváth, A. Yamamoto (Eds.), Inductive Logic Programming. X, 401 pages. 2003.

Vol. 2821: A. Günter, R. Kruse, B. Neumann (Eds.), KI 2003: Advances in Artificial Intelligence. XII, 662 pages. 2003.

Vol. 2807: V. Matoušek, P. Mautner (Eds.), Text, Speech and Dialogue. XIII, 426 pages. 2003.

Vol. 2801: W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, J.T. Kim (Eds.), Advances in Artificial Life. XVI, 905 pages. 2003.

Vol. 2797: O.R. Zaïane, S.J. Simoff, C. Djeraba (Eds.), Mining Multimedia and Complex Data. XII, 281 pages. 2003.

Vol. 2792: T. Rist, R.S. Aylett, D. Ballin, J. Rickel (Eds.), Intelligent Virtual Agents. XV, 364 pages. 2003.

Vol. 2782: M. Klusch, A. Omicini, S. Ossowski, H. Laamanen (Eds.), Cooperative Information Agents VII. XI, 345 pages. 2003.

Vol. 2780: M. Dojat, E. Keravnou, P. Barahona (Eds.), Artificial Intelligence in Medicine. XIII, 388 pages. 2003.

Vol. 2777: B. Schölkopf, M.K. Warmuth (Eds.), Learning Theory and Kernel Machines. XIV, 746 pages. 2003.

Vol. 2752: G.A. Kaminka, P.U. Lima, R. Rojas (Eds.), RoboCup 2002: Robot Soccer World Cup VI. XVI, 498 pages. 2003.

Vol. 2741: F. Baader (Ed.), Automated Deduction – CADE-19. XII, 503 pages. 2003.

Vol. 2705: S. Renals, G. Grefenstette (Eds.), Text- and Speech-Triggered Information Access. VII, 197 pages. 2003.

Vol. 2703: O.R. Zaïane, J. Srivastava, M. Spiliopoulou, B. Masand (Eds.), WEBKDD 2002 - MiningWeb Data for Discovering Usage Patterns and Profiles. IX, 181 pages. 2003.

Vol. 2700: M.T. Pazienza (Ed.), Extraction in the Web Era. XIII, 163 pages. 2003.

Vol. 2699: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff, D.F. Gordon-Spears (Eds.), Formal Approaches to Agent-Based Systems. IX, 297 pages. 2002.

Vol. 2691: V. Mařík, J.P. Müller, M. Pechoucek (Eds.), Multi-Agent Systems and Applications III. XIV, 660 pages. 2003.

Vol. 2684: M.V. Butz, O. Sigaud, P. Gérard (Eds.), Anticipatory Behavior in Adaptive Learning Systems. X, 303 pages. 2003.

Vol. 2682: R. Meo, P.L. Lanzi, M. Klemettinen (Eds.), Database Support for Data Mining Applications. XII, 325 pages. 2004.

Vol. 2671: Y. Xiang, B. Chaib-draa (Eds.), Advances in Artificial Intelligence. XIV, 642 pages. 2003.

Vol. 2663: E. Menasalvas, J. Segovia, P.S. Szczepaniak (Eds.), Advances in Web Intelligence. XII, 350 pages. 2003.

Vol. 2661: P.L. Lanzi, W. Stolzmann, S.W. Wilson (Eds.), Learning Classifier Systems. VII, 231 pages. 2003.

Vol. 2654: U. Schmid, Inductive Synthesis of Functional Programs. XXII, 398 pages. 2003.

Vol. 2650: M.-P. Huget (Ed.), Communications in Multiagent Systems. VIII, 323 pages. 2003.

Vol. 2645: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XI, 320 pages. 2003.

Vol. 2639: G. Wang, Q. Liu, Y. Yao, A. Skowron (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. XVII, 741 pages. 2003.

Preface

AISC 2004, the 7th International Conference on Artificial Intelligence and Symbolic Computation, was the latest in the series of specialized biennial conferences founded in 1992 by Jacques Calmet of the Universität Karlsruhe and John Campbell of University College London with the initial title *Artificial Intelligence and Symbolic Mathematical Computing (AISMC)*. The *M* disappeared from the title between the 1996 and 1998 conferences. As the editors of the AISC 1998 proceedings said, *the organizers of the current meeting decided to drop the adjective 'mathematical' and to emphasize that the conference is concerned with all aspects of symbolic computation in AI: mathematical foundations, implementations, and applications, including applications in industry and academia.*

This remains the intended profile of the series, and will figure in the call for papers for AISC 2006, which is intended to take place in China. The distribution of papers in the present volume over all the areas of AISC happens to be rather noticeably mathematical, an effect that emerged because we were concerned to select the best relevant papers that were offered to us in 2004, irrespective of their particular topics; hence the title on the cover. Nevertheless, we encourage researchers over the entire spectrum of AISC, as expressed by the 1998 quotation above, to be in touch with us about their interests and the possibility of eventual submission of papers on their work for the next conference in the series.

The papers in the present volume are evidence of the health of the field of AISC. Additionally, there are two reasons for optimism about the continuation of this situation.

The first is that almost all the items in the list of useful areas for future research that the editors of the proceedings of the first conference in 1992 suggested in a 'state of the field' paper there are represented in AISC 2004. Many have of course been present in other AISC conferences too, but never so many as in this year's conference: theorem proving, expert systems, qualitative reasoning, Gröbner bases, differential and integral expressions, computational group theory, constraint-based programming, specification (implying verification), and instances of automated learning, for example. The only major items from the 1992 list that would be needed here to make up what poker players might call a full house are knowledge representation and intelligent user interfaces for mathematical tasks and mathematical reasoning – but while a word search in this volume may not find them, ingredients of both are undoubtedly present this year. (For a hint, see the next paragraph.)

The second of our reasons for an optimistic view of AISC is the maturation of a scientific proposal or prediction that dates back to 1985. In founding the Journal of Symbolic Computation in that year, one of us proposed that SC should encompass both exact mathematical algorithmics (computer algebra) and automated reasoning. Only in recent years has an integration and interaction of these two fields started to materialize. Since 2001 in particular, this has given rise to the MKM (mathematical knowledge management) 'movement', which considers seriously the automation of the entire process of mathematical theory exploration. This is now one of the most promising areas for the application of AI methods in general (for invention or discovery of mathematical concepts, problems, theorems and algorithms) to mathematics/SC and vice versa.

We are happy to be continuing the fruitful collaboration with Springer which started with the first AISMC conference in 1992 and which permitted the publication of the proceedings in the Lecture Notes in Computer Science (LNCS 737, 958, 1138) series from 1992 to 1996 and the Lecture Notes in Artificial Intelligence (LNAI 1476, 1930, 2385) series subsequently.

We, the AISC steering committee, and the organizers of the conference, are grateful to the following bodies for their financial contributions towards its operation and success: Linzer Hochschulfonds, Upper Austrian Government, FWF (Austrian Science Foundation), Raiffeisenlandesbank Upper Austria, Siemens Austria, IBM Austria, and CoLogNET.

Our thanks are also due to the members of the program committee and several additional anonymous referees, and to those who ensured the effective running of the actual conference and its Web sites.

In this latter connection, we administered the submission and selection of papers for AISC 2004 entirely through special-purpose conference software for the first time in the history of AISC, using the START V2 conference manager described at **www.softconf.com**. This contributed substantially to the efficiency of the whole process, and allowed us to respect an unusually tight set of deadlines. We appreciate the prompt and helpful advice on using this software that we received from Rich Gerber whenever we needed it.

The effectiveness of the final stage of production of this volume was due mainly to the intensive work of Theodoros Pagtzis. We express our gratitude to him.

July 2004

Bruno Buchberger
John Campbell

Organization

AISC 2004, the 7th international conference on Artificial Intelligence and Symbolic Computation, was held at Schloss Hagenberg, near Linz, during 22–24 September 2004.

The Research Institute for Symbolic Computation (RISC) of the Johannes-Kepler Universitat Linz, and the Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences (Österreichische Akademie der Wissenschaften), Linz were jointly responsible for the organization and the local arrangements for the conference.

Conference Direction

Conference Chair	Bruno Buchberger (Johannes-Kepler-Universität Linz)
Program Chair	John Campbell (University College London, UK)

Local Arrangements

Local Organizers	Betina Curtis Ramona Pöchlinger
Websites and Publicity	Camelia Kocsis Koji Nakagawa Florina Piroi
Logic Programming Tutorial	Klaus Trümper (University of Texas at Dallas)

Program Committee

Luigia Carlucci Aiello	(Università 'La Sapienza', Rome)
Michael Beeson	(San José State University)
Belaid Benhamou	(Université de Provence)
Bruno Buchberger	(Johannes-Kepler-Universität Linz)
Jacques Calmet	(Universität Karlsruhe)
Bill Farmer	(McMaster University)
Jacques Fleuriot	(University of Edinburgh)
Laurent Henocque	(Université de la Méditerranée)
Tetsuo Ida	(Tsukuba University)
Michael Kohlhase	(Universität Bremen)
Steve Linton	(University of St Andrews)
Aart Middeldorp	(Universität Innsbruck)
Eric Monfroy	(Université de Nantes)

John Perram	(Syddansk Universitet)
Jochen Pfalzgraf	(Universität Salzburg)
Zbigniew Ras	(University of North Carolina)
Eugenio Roanes-Lozano	(Universidad Complutense de Madrid)
Tomas Recio	(Universidad de Cantabria)
Volker Sorge	(University of Birmingham)
John Stell	(University of Leeds)
Carolyn Talcott	(SRI International)
Dongming Wang	(Université de Paris 6)
Wolfgang Windsteiger	(Johannes-Kepler-Universität Linz)

Sponsoring Institutions

Linzer Hochschulfonds
Upper Austrian Government
FWF (Austrian Science Foundation)
Raiffeisenlandesbank Upper Austria
Siemens Austria
IBM Austria
CoLogNET

Table of Contents

Short Presentations

The Algorithmization of Physics: Math Between Science and Engineering*

Markus Rosenkranz

Johann Radon Institute for Computational and Applied Mathematics
Austrian Academy of Sciences, A-4040 Linz, Austria
Markus.Rosenkranz@oeaw.ac.at

Abstract. I give a concise description of my personal view on symbolic computation, its place within mathematics and its relation to algebra. This view is exemplified by a recent result from my own research: a new symbolic solution method for linear two-point boundary value problems. The essential features of this method are discussed with regard to a potentially novel line of research in symbolic computation.

1 Physics: The Source and Target of Math

What is the *nature of mathematics*? Over the centuries, philosophers and mathematicians have proposed various different answers to this elusive and intriguing question. Any reasonable attempt to systematically analyze these answers is a major epistemological endeavor. The goal of this presentation is more modest: I want to give you a personal (partial) answer to the question posed above, an answer that highlights some aspects of mathematics that I consider crucial from the perspective of symbolic computation. At the end of my presentation, I will substantiate my view by a recent example from my own research.

According to [4], humankind has cultivated the art of *rational problem solving* in a fundamental three-step rhythm:

1. *Observation:* The problem of the real world is specified by extracting relevant data in an abstract model.
2. *Reasoning:* The model problem is solved by suitable reasoning steps, carried out solely in the abstract model.
3. *Action:* The model solution is applied in the real world by effectuating the desired result.

In this view, *mathematics* is not limited to any particular objects like numbers or figures; it is simply "reasoning in abstract models" (item number 2 in the enumeration above). For highlighting its place in the overall picture, let us take up the example of physics – of course, one can make similar observations for other disciplines like chemistry, biology, economics or psychology.

We can see physics as a *natural science* that deals with observations about matter and energy (item number 1 in the three-step rhythm). In doing so, it

* This work is supported by the Austrian Science Foundation FWF in project F1322.

B. Buchberger and J.A. Campbell (Eds.): AISC 2004, LNAI 3249, pp. 1–7, 2004.

extracts various patterns from the mass of empirical data and tabulates them in natural laws. It is here that it comes in contact with mathematics, which provides a rich supply of abstract structures for clothing these laws. In this process, physicists have often stimulated deep mathematical research for establishing the concepts asked for (e.g. distributions for modeling point sources), sometimes they have also found ready-made material in an unexpected corner of "pure mathematics" (e.g. Rogers-Ramanujan identities for kinetic gas theory); most of the time, however, we see a parallel movement of mutual fertilization.

As a branch of *technical engineering*, physics is utilized for constructing the machines that we encounter in the world of technology (item number 3 in the three-step rhythm). Engineers are nowadays equipped with a huge body of powerful applied mathematics – often hidden in the special-purpose software at their disposal – for controlling some processes of nature precisely in the way desired at a certain site (e.g. the temperature profile of a chemical reactor). If we are inclined to look down to this "down-to-earth math", we should not forget that it is not only the most prominent source of our money but also the immediate reason for our present-day prosperity.

Of course, the above assignment 1 $\sim$ natural sciences (e.g. theoretical physics), 2 $\sim$ formal science (i.e. mathematics), 3 $\sim$ technical sciences (e.g. engineering physics) must be understood *cum grano salis*: Abstract "mathematical" reasoning steps are also employed in the natural and technical sciences, and a mathematician will certainly benefit from understanding the physical context of various mathematical structures. Besides this, the construction of models is also performed within mathematics when powerful concepts are invented for solving math-internal problems in the same three-step rhythm (e.g. 1 $\sim$ extension fields, 2 $\sim$ Galois groups, 3 $\sim$ solvability criterion).

2 Algebraization: The Commitment to Computing

The above view of mathematics prefers its dynamical side (problem solving) over its static one (knowledge acquisition), but actually the two sides are intimately connected (knowledge is needed for solving problems, and problems are the best filter for building up relevant knowledge bases). The *dynamic view of mathematics* is also the natural starting point for symbolic computation, as I will explicate in the next section. In fact, one can see symbolic computation as its strictest realization, deeply embedded in the overall organism of less constructive or "structural" mathematics.

Within symbolic computation, I will focus on *computer algebra* in this presentation. Strictly speaking, this means that we restrict our interest to algebraic structures (domains with functional signatures and equational axioms like rings). But this is not a dogmatic distinction, rather a point of emphasis; e.g. fields are also counted among the algebraic structures despite their non-equational axiom on reciprocals. In some sense computer algebra is the most traditional branch of symbolic computation since rewriting functions along equational chains is maybe the most natural form of "computation with symbols". What is more important, though, is the axiomatic description of the algebraic structures in use.

Judging from our present understanding, it may seem obvious that one should proceed in this way. Looking at the historical development, however, we perceive a movement of *increasing abstraction* that has not stopped at the point indicated above [9]. We can see four distinct stages in this abstraction process:

1. *Concrete Algebra* (Weber, Fricke): Virtually any ring considered was a subring of the integers or of the complex polynomials or of some algebraic number field (similar for other domains). Various "identical" results were proved separately for different instances.
2. *Abstract Algebra* (Noether, van der Waerden): Rings are described by their axioms; the classical domains mentioned above are subsumed as examples. All the proofs are now done once, within "ring theory".
3. *Universal Algebra* (Graetzer, Cohn): Classes of algebraic structures are considered collectively; rings are just one instance of an algebraic structure. Results like the homomorphism theorem can be proved for generic algebraic structures that specialize to rings, groups, and the like.
4. *Category Theory* (MacLane, Eilenberg): Categories are any collection of objects (like algebraic or non-algebraic structures) connected through arrows (like homomorphisms in algebraic structures). The objects need not have a set-theoretic texture (as the carriers of structures have).

The role of mathematics as reasoning in abstract models becomes very clear in the process of algebraization: the mathematical models are now specified precisely by way of axioms and we need no longer rely on having the same intuition about them. Let me detail this by looking at one of the most fundamental structures used in physics – the notion of the *continuum*, which provides a scale for measuring virtually all physical quantities. Its axiomatization as the complete ordered field of reals needed centuries of focused mathematical research culminating in the categoricity result. Proceeding with computer algebra, we would now strip off the topological aspects (the "complete ordered" part) from the algebraic ones (the "field" part) and then study its computable subfields (finite extensions of the rationals).

If one models physical quantities by real numbers, analyzing their mutual dependence amounts to studying real functions (real analysis), and the natural laws governing them are written as differential equations. Their application to specific situations is controlled by adjusting some data like various parameters and initial/boundary conditions. Since the latter are the most frequent data in physical problems [10], *boundary value problems* (BVPs) will serve as a fitting key example in the last section of this presentation.

3 Algorithmization: The Realization of Computing

In order to actually "compute" the solution of a problem in an abstract model, algebraization alone is not enough. We have already observed this in the above example of the continuum: The field of real numbers is regarded as an algebraic domain, but it is clearly uncomputable because of its uncountable carrier. In view of these facts, Buchberger [7, 3, 5] has defined *symbolic computation* (in particular: computer algebra) as that part of algorithmic mathematics (in particular: algorithmic algebra) which solves problems stated in non-algorithmic domains by translating them into isomorphic algorithmic representations.

Let us look at three immediate examples:

1. The traditional *definition of polynomials* introduces them as certain (infinite!) congruence classes over some term algebra in the variety of unital commutative rings [13]. The modern definition starts from a monoid ring [12], which turns out to be an isomorphic translation of the former, basically encoding the canonical representatives of the congruence classes.
2. As another example, consider the cardinal question of *ideal membership* in algebraic geometry: As infinite sets, ideals cannot directly be treated by algorithmic methods; representing them via Gröbner bases allows a finitary description and a solution of the ideal membership problem [1, 2, 6].
3. Finally, let us consider an example from a traditional domain of symbolic computation that does not belong to computer algebra, namely *automated theorem proving.* It is based on translating the non-algorithmic (semantic) concept of consequence into the algorithmic (syntactic) concept of deducibility, the isomorphism being guaranteed by Gödel's Completeness Theorem.

Returning to the example of boundary value problems introduced above, we should also note that there are actually two famous approaches for algorithmization: symbolic computation takes the path through algebraization, whereas *numerical computation* goes through approximation. Simplifying matters a bit, we could say that symbolic computation hunts down the algebraic structure of the continuum, numerical computation its topological structure[1]. But while industrial mathematics is virtually flooded with numerical solvers (mostly of finite-element type), it is strange to notice that computer algebra systems like Maple or Mathematica do not provide any command for attacking those BVPs that have a symbolic solution. My own research on symbolic functional analysis can be seen as an endeavor to change this situation.

4 Symbolic Functional Analysis: Conquering a New Territory for Computing

I will now sketch my own contribution to the exciting process of conquering more and more territory through algebraization and algorithmization. As mentioned before, it deals with certain *boundary value problems.* More precisely, we are

[1] The ideal algorithmic approach to problem solving would be to combine the best parts of both symbolic and numerical computation, which is the overall objective of a 10-year special research project (SFB) at Johannes Kepler University. My own research there takes place in the subproject [17] on symbolic functional analysis; for some details, see the next section.

given a function f in $C^\infty[0,1]$ say[2], and we want to find a solution u in $C^\infty[0,1]$ such that

$$\begin{aligned} Tu &= f, \\ B_0 u = u_0, \ldots, B_{n-1}u &= u_{n-1}\ . \end{aligned} \tag{1}$$

Here T is a linear differential operator like $T = x^2D^2 - 2e^xD + 1$ and $B_0, \ldots, B_{n-1}$ are boundary operators like $u \mapsto u'(0) - 2u(1)$, whose number n should coincide with the order of T. Furthermore, we require the boundary conditions to be such that the solution u exists and is unique for every choice of f; in other words, we consider only regular BVPs.

In my own understanding, BVPs are the prototype for a new kind of problem in symbolic computation: Whereas "computer algebra" focuses on algorithmically solving for numbers (typical example: Gröbner bases for triangularizing polynomial systems) and "computer analysis" does the same for functions (typical example: differential algebra for solving differential equations), the proper realm of "computer operator-theory" or *symbolic functional analyis* would be solving for operators. See [17] for more details on this three-floor conception of the algebraic part of symbolic computation.

Why are BVPs an instance of *solving for operators*? The reason is that the forcing function f in (1) is understood as a symbolic parameter: One wants to have the solution u as a term that contains f as a free variable. In other words, one needs an operator G that maps any given f to u. For making this explicit, let us rewrite the traditional formulation (1) as

$$\begin{aligned} TG &= 1, \\ B_0 G = 0, \ldots, B_{n-1}G &= 0\ . \end{aligned} \tag{2}$$

Here 1 and 0 denote the identity and zero operator, respectively. Note also that I have passed to homogeneous boundary conditions (it is always possible to reduce a fully inhomogeneous problem to such a semi-inhomogeneous one).

The *crucial idea* of my solution method for (2) is to model the above operators as noncommutative polynomials and to extract their algebraically relevant properties into a collection of identities. For details, please refer to my PhD thesis [14]; see also [16, 15]. The outcome ot this strategical plan is the noncommutative polynomial ring $\mathbb{C}\langle\{D, A, B, L, R\} \cup \{\lceil f \rceil | f \in \mathfrak{F}^\#\}\rangle$ together with a collection of 36 polynomial equalities. The indeterminates D, A, B, L, R and $\lceil f \rceil$ stand for differentiation, integral, cointegral, left boundary value, right boundary value and the multiplication operator induced by f; here f may range over any so-called analytic algebra (a natural extension of a differential algebra), e.g. the exponential polynomials. The 36 polynomial identities express properties like the product rule of differentiation, the fundamental theorem of calculus, and integration by parts.

[2] The smoothness conditions can be dispensed with by passing to distributions on $[0,1]$. Of course one can also choose an arbitrary finite interval $[a,b]$ instead of the unit interval. See [15] for details.