Jacques Calmet   John A. Campbell (Eds.)

Artificial Intelligence and Symbolic Mathematical Computing

International Conference AISMC-1
Karlsruhe, Germany, August 3-6, 1992
Proceedings

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editors

Gerhard Goos
Universität Karlsruhe
Postfach 69 80
Vincenz-Priessnitz-Straße 1
D-76131 Karlsruhe, Germany

Juris Hartmanis
Cornell University
Department of Computer Science
4130 Upson Hall
Ithaca, NY 14853, USA

Volume Editors

Jacques Calmet
University of Karlsruhe
Postfach 69 80, D-76128 Karlsruhe, Germany

John A. Campbell
Dept. of Computer Science, University College London
Gower Street, London WC1E 6BT, United Kingdom

CR Subject Classification (1991): I.1-2, G.1-2

ISBN 3-540-57322-4 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57322-4 Springer-Verlag New York Berlin Heidelberg

Typesetting: Camera-ready by author
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

Foreword

The present volume reproduces the papers, updated in some cases, given at the first AISMC (Artificial Intelligence and Symbolic Mathematical Computations) conference which was held in Karlsruhe, August 3–6, 1992. This was the first conference to be devoted to such a topic after a long period when SMC made no appearance in AI conferences, though it used to be welcome in the early days of AI. Some conferences have been held recently on mathematics and AI or on applications of symbolic computing in AI; however, none covers similarly the topics of this conference.

Because of the novelty of the domain, authors were given longer allocations of time than usual in which to present their work. As a result, extended and fruitful discussions followed each paper. The introductory item in this book, which was not presented during the conference, reflects in many ways the flavour of these discussions and aims to set out the framework for future activities in this domain of research.

There was a unanimous demand from the attendees to hold conferences of this kind regularly. We plan to organize them biannually in even years.

This conference was organized mainly by the Institute of Algorithms and Cognitive Systems at the University of Karlsruhe. Sponsorships and financial supports from DFG, DEC Germany and IBM Germany are thankfully acknowledged.

We also express our gratitude to the members of the program committee and to the organizers, particularly to Karsten Homann and Ono Tjandra.

August 1993 — Jacques Calmet and John A. Campbell

Conference Committee: Jacques Calmet (Germany – Conference Chairman)
John A. Campbell (UK – Program Committee Chairman)
David Y.Y. Yun (USA - Publicity Chairman)

Program Committee:
L. Aiello (Italy)
J. Calmet (Germany)
J.A. Campbell (UK, Chairman)
E. Engeler (Switzerland)
J. Johnson (UK)
L. Joskowicz (USA)
M. Karpinski (Germany)
J.-L. Lassez (USA)
R. Lopez de Mantaras (Spain)
R. Loos (Germany)
A. Miola (Italy)
A.C. Norman (UK)
R. Parikh (USA)
M. Pohst (Germany)
Z.W. Ras (USA)
E. Sacks (USA)
E. Sandewall (Sweden)
S. Stifter (Austria)
D. Weld (USA)
D.Y.Y. Yun (USA)

Local Organizers: K. Homann, I.A. Tjandra, W. Fakler

Contents

Artificial Intelligence and Symbolic Mathematical Computations

J. CALMET
University of Karlsruhe

J.A. CAMPBELL
University College London

Abstract. This introductory paper summarizes the picture of the territory common to AI and SMC that has evolved from discussions following the presentation of papers given at the 1992 Karlsruhe conference. Its main objective is to highlight some patterns that can be used to guide both sketches of the state of the art in this territory and suggestions for future research activities.

The structure of the paper mirrors the emerging patterns of interaction between AI and symbolic mathematical computing (SMC). We begin with some historical considerations, to put the past relationship between the two subjects into perspective. 1971, the year in which the first major conference on computer algebra was held, can be regarded as the year in which the two subjects started to diverge significantly. The section that follows mentions the topics in AI that have never quite lost touch with SMC. We then consider the offerings that AI can make to SMC and vice versa, with a short intermediate section arguing for the idea of knowledge representation as a bridge between the two subjects. A concluding section looks at what we consider to be the most rewarding possibilities for future developments in research.

We make no claim that our treatment of the past, present or future is exhaustive. This introduction is best seen as a sketch of a survey paper. A more detailed version will appear elsewhere.

1 An Historical View

1.1 The first 15 years of modern AI

A watershed of AI was the summer workshop held at Dartmouth College in 1956, at which many of the founding fathers of the subject met to compare their work and to define an agenda for the near future [Mc]. During the 15 years that followed (until the time of the 1971 Los Angeles computer algebra conference), it was normal to find papers on SMC in AI sources and conferences. This is not surprising in retrospect, because of the mathematical backgrounds of the early AI researchers and because symbolic mathematics offered some problems of a manageable size where the criteria for successful solution were clear-cut.

Shortly after 1971, papers on symbolic computation disappeared almost completely from the places where AI material was published. Why was this? The answer, which we discuss below, has several parts, but a common strand of the explanation is that research on both sides of the fence was too successful.

The problems considered in early AI were basically search, computational logic (for theorem-proving and construction of plans, e.g. for carrying out searches, by steps similar to successive steps of a proof), pattern-matching, and symbolic mathematics. (Natural-language processing has also been a long-lived partner of AI, meeting up with AI at conferences but otherwise travelling mainly on a separate parallel track). What they had in common, and what distinguished them at the same time from early non-AI programming in Fortran and its predecessors, was that conventional programming dealt with algorithmic methods while AI relied on *heuristics.*

In the particular case of symbolic mathematics, it is not hard to find algorithmic applications (e.g. in differential calculus) that would involve novel uses of the computing technology of the 1950s. But the AI applications were heuristic. Good examples are Gelernter's geometrical theorem-prover [Ge] and Slagle's work on integral calculus [Sl]. These represented the state of the art in the early 1960s, when finding good heuristics and therefore making something new work for the first time was more important than thinking about regularities in what was found.

The best example of a change of emphasis towards looking at the regularities is Moses' [Mo] integral-calculus system SIN, which was publicized first in his PhD thesis at the beginning of 1967 and which was absorbed later into the development of the MACSYMA system for symbolic mathematics. There were basically two kinds of regularity. The one that was historically more important for the separation of AI and SMC was that the study of the heuristics collected for SIN and from other work on programs for integral calculus led to an *evolution away from heuristics and towards algorithms.* As integration in finite terms became more of a mathematical specialization and hence more algorithmic, it became less attractive as a topic for AI referees and hence AI conferences.

This story has been repeated for other areas of AI. It is clear from a reading of series of AI conference proceedings (the IJCAI series is the best source) from the late 1960s until today, that AI is a moving target: its contents change with time, and in particular some topics (e.g. symbolic mathematics) that were once popular disappear completely from later conferences. The paradox of AI is that "AI exports its successes". Success implies not only that something works for users but also that it works efficiently. Algorithms are usually more efficient than heuristics for the same job, so that there is a social pressure to turn good heuristics into good algorithms. And there seems to have been a tacit agreement, in many topics that were once focuses for AI research, that when something is algorithmic and no longer heuristic, it is not really suitable for reporting at AI conferences.

The second regularity that showed up in the heuristics collected for SIN, the system DENDRAL [Bc] for determination of molecular structure from experimental data, and several less well-known projects from the late 1960s onwards, was the *rule* structure, now standard in expert systems, for individual heuristics. In fact, it has been said that the SIN-MACSYMA system and DENDRAL were the first modern expert systems (through this is not an accurate picture of the former).

The general success of the rule-based (and therefore still heuristic) approach in AI forced attention in research towards the examination of rules, and towards a common labelling for rules and other schemes such as inheritance networks or semantic nets. Because they all held knowledge in some form, the label of "knowledge representation" emerged. The algorithmic direction of work on MACSYMA and similar software at the same time as this development led symbolic research. Hence the divorce proceedings with AI that started in 1971.

1.2 The first 20 years of SMC

It is convenient to date the origin of SMC to the two 1953 Master's theses of Kahrimanian [Ka] and Nolan [No], who wrote assembly-code programs to perform symbolic differentiation of simple mathematical expressions. These first examples were algorithmic and not heuristic.

After that piece of history, the first attempts to write software to solve serious problems were in celestial mechanics, and were associated with the group of A. Deprit [De]. These attempts, from the late 1950s, were motivated by the fact that accurate computations of trajectories of astronomical bodies required extremely long symbolic expressions, and that manual calculations could take a significant fraction of a mathematician's lifetime and could contain undetected errors. The computations typically involved substituting Poisson-series expressions in several symbolic variables into differential equations, and determination of solutions as series expansions.

The mainstream of symbolic computation during the 1960s was in application-oriented systems, either for general algebraic manipulations or specific extensions to high energy physics, celestial mechanics, general relativity or group theory. Many well-known modern systems such as REDUCE and CAYLEY started their development during this period.

As in the case of the 1951 work, there were no significant heuristic components in most of these systems. The SIN-MACSYMA activity was an isolated exception. Work on SMC could still be done and reported in circles close to AI because of the past honorary membership (dating from the choice of symbolic mathematics, mainly integral calculus, as a source of problems for AI after 1956) of the subject in AI. We have given part of the explanation for the post-1971 separation of AI and computer algebra in the sub-section above. The rest of the explanation is that the 1971 computer algebra conference and the later foundation of the ACM special-interest group SIGSAM and its Newsletter gave the work its own home; there was no further pressure to look for publication in outlets that were primarily devoted to AI.

1.3 MACSYMA as an expert system?

As we have said above, the SIN-MACSYMA work has been quoted as an early example of the development of an expert system (e.g. in [Ba]). To the extent that the heuristic knowledge in SIN could be read off as rules from the source code, this would be true. In practice it was not totally false, but one needed to be an expert excavator of LISP code to detect the traces of truth in the statement. Also, some of the heuristic information was more about pattern-matching than the use of rules. Finally, the architecture of the software was not set up to allow the rule-like material to be separated easily from the code that made use of it, i.e. there was no obvious way to distinguish a rule base in SIN from an inference engine.

DENDRAL has a much better claim to reflect the structure of an expert system, though this identification of the parts of the structure in the DENDRAL code would still not have been simple.

SMC did not invent expert systems, unfortunately. Nevertheless, it can claim to have come close. For example, in REDUCE and other early programs, the basic algorithm for symbolic differentiation must be supplemented by knowledge about the derivative $f'(x) = g(x)$ of any new function $f(x)$ that the programs have to differentiate. Each such pair is declared as a form of substitution, e.g. FOR ALL X LET DF(F(X),X)=G(X). The pairs are held in a format that allows the original meaning to be read off, and that distinguishes this knowledge from the code for the programs themselves.

This suggestion is a jeu d'esprit rather than a serious contribution to history - but it counters the belief in some quarters in AI that a particular symbolic computing system (MACSYMA) was almost the first expert system.

1.4 Why has a new convergence between AI and computer algebra been happening?

The original divorce between AI and computer algebra was driven by the fact that AI continued to focus on and develop its understanding of the notion of heuristics while symbolic computation became more and more algorithmic. The algorithmic emphasis was increased by a migration of professional mathematicians into research on computer algebra.

We have now reached a point where concentration on algorithmics rarely leads to qualitatively new capabilities in symbolic computing systems. On the other hand, users of the systems can still be seen doing pen-and-paper calculations, or trying alternative ways of expressing instructions to the systems, because some of their needs have not been met. Generally, the reason why the systems do not service these calculations (or service them well) is that the needs require heuristic treatment. For example, the control of intermediate expression swell is in general still a heuristic exercise.

Heuristic knowledge about symbolic mathematics exists in many places. The number of attempts to capture it systematically and put it to work via AI-influenced methods is not yet large, but there is an increasing trend towards doing such research. The contribution of H. Hong to this book is one excellent example of the trend. There are potential payoffs for AI, in the evaluation of its methods in a new area of application, and for users of symbolic computing systems, who can expect qualitatively new kinds of support for their calculations when the systems become "intelligent assistants".

From the AI side, some of its topics for research and applications have a mathematical flavor, after quite a long period since 1971 when there was not much contact between mathematics and AI. The AI topic that has had the greatest effect in this direction is qualitative reasoning, which typically requires the use of a mixture of formal operations and heuristics on mathematical entities such as equations. We pay more attention to qualitative reasoning in section 2.3.

Qualitative reasoning is simply the most visible instance of the more general effect that formal (mathematical) systems of description of phenomena in different corners of AI lead to non-trivial calculations, which symbolic computing systems can perform, either immediately or after relatively simple changes or additions. Another such instance [Wa], in which the systems can be used immediately, emerges from a method for proving geometrical theorems (a classical AI exercise) constructively by the manipulation of simultaneous linear equations. A further area in which future applications of symbolic computation in AI can be expected is in robotic path planning and varieties of the general "piano movers" problem.

In such topics, at present, the knowledge expressed in AI systems and software can be used to produce statements of problems that are then (most often after some manual translation or massaging) suitable as inputs to symbolic computing systems. A challenge for the near future is to integrate the two types of systems and cut out as much as practicable of the present human intervention.

There is a still further extension of the ideas expressed immediately above. It is increasingly common for conceptual structures in AI to be expressed in mathematical language, usually algebraic and quite often in terms of definitions that specify some unusual algebra. Here, one is interested not just in doing symbolic calculations, e.g. in the sense of path planning or qualitative reasoning, but in exploring the internal features of the algebra and its consequences for the AI problem that gave rise to it. In this sense, symbolic computing systems are possible components of an "AI algebraist's laboratory". Integrating them with other relevant software produced from AI research is basically the same kind of challenge that we have mentioned in the previous paragraph.

The final reason for the existence of converging paths between AI and SMC is that specialists in knowledge representation are becoming interested in capturing and using the considerable amount of heuristic knowledge that mathematicians possess about suitable symbolic structures (e.g. differential equations). Apart from the technical interest of doing this, there is the attraction that success will make it possible to build practical systems that combine operational mathematical knowledge with the calculational capabilities of present systems like MACSYMA and REDUCE. Any substantial progress in this area will give such systems their first significant boost in functionality since the early 1970s.