Resource Guided Concurrent Deduction

Christoph Benzmueller*; Mateja Jamnik*; Manfred Kerber*; Volker Sorge†
*School of Computer Science, The University of Birmingham
Edgbaston, Birmingham B15 2TT, United Kingdom
†Fachbereich Informatik (FB 14), Universitaet des Saarlandes
D-66041 Saarbruecken, Germany
C.E.Benzmuller|M.Jamnik|M.Kerber@cs.bham.ac.uk; sorge@ags.uni-sb.de

1 Motivation

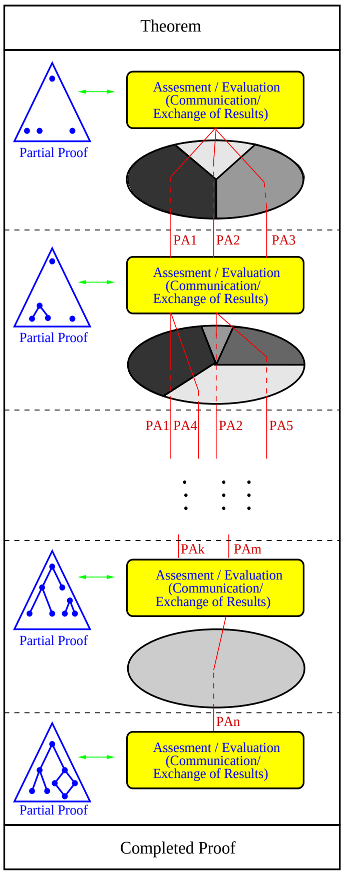
Our poster proposes an architecture for resource guided concurrent mechanised deduction which is motivated by some findings in cognitive science. Our architecture particularly reflects Hadamard's "Psychology of Invention" [Hadamard44]. In his study Hadamard describes the predominant role of the unconsciousness when humans try to solve hard mathematical problems. He explains this phenomenon by its most important feature, namely that it can make (and indeed makes) use of concurrent search (whereas conscious thought cannot be concurrent), see p. 22 Hadamard (1944): *"Therefore, we see that the unconscious has the important property of being manifold; several and probably many things can and do occur in it simultaneously. This contrasts with the conscious ego which is unique. We also see that this multiplicity of the unconscious enables it to carry out a work of synthesis."* That is, in Hadamard's view, it is important to follow different lines of reasoning simultaneously in order to come to a successful synthesis.

Human reasoning has been described in traditional AI (e.g., expert systems) as a process of applying rules to a working memory of facts in a recognise-act cycle. In each cycle one applicable rule is selected and applied. While this is a successful and appropriate approximation for many tasks (in particular for well understood domains), it seems to have some limitations, which can be better captured by an approach that is not only cooperative but also concurrent. And Minsky (1985) gives convincing arguments that the mind of a single person can and should be considered as a society of agents. Put in the context of mathematical reasoning this indicates that it is necessary to go beyond the traditional picture of a single reasoner acting on a working memory – even for adequately describing the reasoning process of a single human mathematician.

There are two major approaches to automated theorem proving, machine-oriented methods like the resolution method (with all its ramifications) and human-oriented methods. Most prominent amongst the human-oriented methods is the proof planning approach first introduced by Bundy (1988). In our poster we argue that an integration of the two approaches and the simultaneous pursuit of different lines in a proof can be very beneficial. One way of integrating the approaches is to consider a reasoner as a collection of specialised problem solvers, in which machine-oriented methods and planning play different rôles.

2 System Architecture

The architecture (for further details see Benzmüller et al. (1999)) that we describe here allows a number of proof search attempts to be executed in parallel. Each specialised subsystem may try a different proof strategy to find the proof of a conjecture. Hence, a number of different proof strategies are used at the same time in the proof search. However, following all the available strategies simultaneously would quickly consume the available system resources consisting of computation time and memory space. In order to prevent this, and furthermore, to guide the proof search we developed and employ a resource management concept in proof search. Resource management is a technique which distributes the available resources amongst the available subsystems (cf. Zilberstein (1995)). Periodically, it assesses the state of the proof search process, evaluates the progress, chooses a promising direction for further search and redistributes the available resources accordingly. If the current search direction becomes increasingly less promising then backtracking to the previous points in the search space is possible. Hence, only successful or promising proof attempts are allowed to continue searching for a proof. This process is repeated until a proof is found, or some other terminating condition is reached. An important aspect of our architecture is that in each evaluation phase the global proof state is updated, that is, promising partial proofs and especially solved subproblems are reported to a special plan server that maintains the progress of the overall proof search attempt. Furthermore, interesting results may be communicated between the subsystems (for instance, an open subproblem may be passed to a theorem prover that seems to be more appropriate). This communication is supported by the shells implemented around the specialised problem solvers. The resource management mechanism analyses the theorem and decides which subsystems, i.e., which provers, should be launched and what proportion of the resources needs to be assigned to a particular prover. The mechanism is also responsible for restricting the amount of information exchange between subsystems, so that not all of the resources are allocated to the communication. The Figure to the right demonstrates this concurrent resource management based proof planning architecture. The involved planning agents are represented by PA$_n$ and the wals indicate the ammount of resources assigned to them in each reasoning phase.



We argue that the effect of resource management leads to a less brittle search technique which we call focused search.

Breadth-first search is robust in the sense that it is impossible to miss a solution. However, it is normally prohibitively expensive. Heuristic search may be considered as the other extreme case, it is possible to go with modest resources very deep in a search tree. However, the search is brittle in that a single wrong decision may make it go astray and miss a solution, independently of how big the allocated resources are. Focused search can be considered as a compromise — it requires more resources than heuristic search, but not as much as breadth-first search. As a result, a solution can still be found even if the focus of the search is misplaced. Clearly, more resources are necessary in the case of a bad than of a good focus.

We currently realise the so-called focused proof search as an adaptation of the multi-agent planning architecture, MPA Wilkins and Myers (1998), in the proof planning domain. Important infrastructure for this enterprise is provided by the ΩMEGAproof development environment. The main component of MPA is a multi-agent proof planning cell, which consists of 1) several planning agents, 2) a plan server, 3) a domain server, and finally 4) a planning cell manager.

1. The quite heterorgeneous reasoning systems (FO-Reasoners, HO-Reasoners, CAS, etc.) already integrated to ΩMEGA are available as planning agents. And an interactive user may become a concurrent planning agent as well.
2. The plan server stores promising partial proof plans returned by the planning agents in their previous runs within a unified data format. This enables backtracking on two distinct levels: we can backtrack within the actual proof plan by taking back single proof steps or subproofs contributed by some of the planning agents and we can completely shift to some alternative proof attempt that has been abandoned previously.
3. A domain server provides the necessary knowledge for the planning cell manager as well as for the single planning agents. In our context it consists of a structured database of mathematical theories. Moreover, it should contain domain specific knowledge relevant to certain planning agents.
4. The planning cell manager re-organizes and controls the reasoning process in each iteration phase based on its (and/or the users) crucial evaluation and assesment considerations. Its prototype is based on the agent-architecture described in Benzmüller and Sorge (1999) allowing for a close and flexible integration of an interactive user into automated reasoning processes.

3 Conclusion

Our work does not directly follow the long-term goal of building a ‘complete mind’. However we think that we will encounter many of the problems in our limited domain which will have to be solved in building a complete mind. In particular a distinction between different levels, reactive and deliberative modes, meta-level reasoning and so on, seems to be very important in the wider context of mathematical reasoning, maybe even feelings play a role. So we think that our work could be of general interest and that we could benefit from the general work in the area.

References

C. Benzmüller and V. Sorge. Critical Agents Supporting Interactive Theorem Proving. *Proceedings of EPIA-99*, Volume 1695 of *LNAI*, 1999. Springer.

C. Benzmüller, M. Jamnik, and M. Kerber a nd V. Sorge. Towards concurrent resource managed deduction. Tech-Report CSRP-99-17, University of Birmingham, School of Computer Science, 1999.

A. Bundy. The Use of Explicit Plans to Guide Inductive Proofs. *Proceedings of the CADE-9*, volume 310 of *LNCS*, 1988. Springer Verlag, Berlin, Germany.

J. Hadamard. *The Psychology of Invention in the Mathematical Field.* Dover Publications, New York, USA; edition 1949, 1944.

M. Minsky. *The Society oCLf Mind.* Simon & Schuster, New York, USA, 1985.

OMEGA-Homepage (`http://www.ags.uni-sb.de/ omega/`)

D. E. Wilkins and K. L. Myers. A Multiagent Planning Architecture. *Proceedingsof AIPS'98*, 1998. AAAI Press, Menlo Park, CA, USA.

S. Zilberstein. Models of Bounded Rationality. In *AAAI Fall Symposium on Rational Agency*, Cambridge, Massachusetts, November 1995.