Towards Fine-Grained Proof Planning with Critical Agents

Christoph Benzmüller and Volker Sorge
Universität des Saarlandes {chris|sorge}@ags.uni-sb.de

Agent Based Command Suggestion Mechanism In order to support users within large proofs, most interactive theorem provers have some limited capability to suggest parameters of choosen commands or even commands themselves. However, those mechanisms generally have the shortcomings that they use inflexible sequential computation strategies (i.e., although most parameters influence the suggestions of each other they are suggested in some order such that parameters choosen earlier influence those required later, but not vice versa) and do not sufficiently use available system resources by computing suggestions inbetween single interaction steps.

We have developed a mechanism that suggests commands, applicable in the current proof state together with a suitable argument instantiations [BeSo98]. It is based on two layers of societies of autonomous, concurrent agents which steadily work in the background of a system and dynamically update their computational behavior to the state of the proof and/or specific user queries to the suggestion mechanism (cf. Fig. 1). By exchanging relevant results via blackboards the agents cooperatively accumulate useful command suggestions which can be heuristically sorted and presented to the user.

The architecture has been prototypically implemented in the ΩMEGA mathematical assistant system [Ome97] for interactive tactical theorem proving in natural deduction calculus. In [BeSo99] we have enriched the architecture with resource concepts that on the one hand enable the lower layer agents to adjust their computations with respect to their performance in the past and to the logical structure of the proof. On the other hand the behaviour of whole agent societies can be adapted according to the resources available to the overall system.

Although, the mechanism has been developed in the context of interactive theorem proving within ND-calculus it is neither restricted to a particular calculus nor to a solely interactive system. In fact, it can directly be used to compute possible applicable rules for some automated proving procedure or even be automated itself by introducing some limited automation and backtracking capabilities in the mechanism. In the remainder of this abstract we both hint at the latter and describe potential applications of the mechanims within automated proof planning.

Tackling Problems Automatically When automating the suggestion mechanism in a way that always the heuristically preferred suggestion is automatically executed and that backtracking over the single suggestions can be performed, some small proof problems can already be solved fully automatically. In fact, some problems, such as $(p_{o\to o}\ (a_o \land b_o)) \Rightarrow$

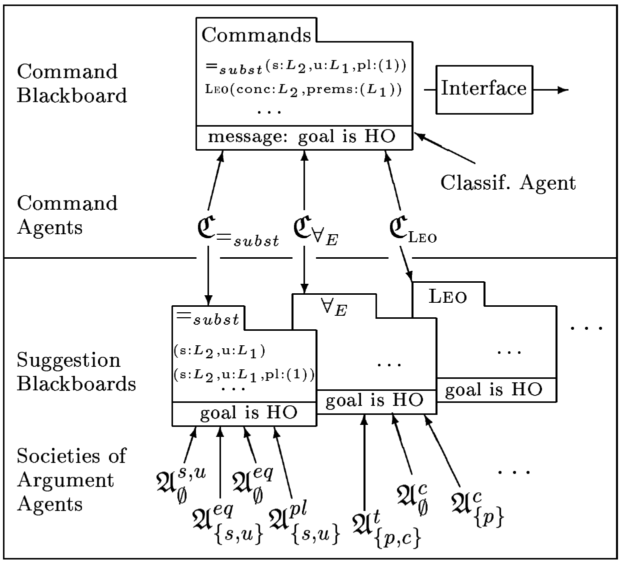

Figure 1: The two layered suggestion mechanism.

$(p\ (b \wedge a))^1$ can be solved even without the need for backtracking. Since the agent mechanism enables to work on several subproblems in parallel the approach can be viewed as a weak form of concurrent, tactic based theorem proving. Surely, tactic based theorem proving based on an underlying natural deduction approach stretches huge search spaces when considering non-trivial examples, such that the problem solving capabilities of such a low level approach will certainly be limited. But note that we can flexibly alter and exchange the rules and tactics underlying our approach in order to influence the size of the search space. It especially appears to be interesting to apply our proposed approach to exactly the rules provided by the Normal Form Natural Calculus [SiBy98] which aims at a fruitful restriction of the search space by allowing only for the construction of ND-proofs that have a particular normal form. The additonal side conditions of the rules in this approach, which restrict and guide their applicabilty, can be easily specified within our approach.

Suggesting Methods in Proof Planning The approach is not restricted to specify applicability conditions for rules and tactics only, but can analogously be employed for proof methods as used in proof planning approaches (e.g., see [Bun88]). So far the specification of the argument agents for rules and tactics describes structural properties of single arguments as well as structural dependencies between the different arguments. Similarly we can specify agents which check the particular pre-conditions of a proof method, i.e., check for proof lines matching with those required for a method to be applicable or verify additional application conditions of the method. Thus the hope is that we can at least to some extend exploit concurrency even within a traditional proof planner when computing the applicable methods with respect to the given proof goal by cooperating argument agents in each proof step.

To guide the proof planners selection of methods heuristical knowledge can be realised with the suggestion mechanism in different ways: (i) Resource allocations: By adjusting the resource allocations for the agents belonging to the proof methods one influences the suggested

[1] The example is small but quite challenging as it involves the application of extensionality principles.

steps in the proof planning approach in the same way as adjustments influence the suggested commands in the current approach. Within the resource adaptive extension [BeSo99] the proof methods can even adapt their own resource allocations with respect to the current goal and their experience in the past. (ii) Sorting criteria: Sorting criteria are employed to select particular suggestions from the suggestion blackboards. E.g., among all suggestions on the command blackboard we prefer those that might yield the largest proof steps. Any modification to these sorting criterias will obviously influence the proof planners selection of methods in each planning step. (iii) Informed activation & deactivation: In our approach agents are aware of the mathematical theory the command they work for is associated with. E.g., agents that work for a command which applies a tactic that deals with computations on natural numbers will automatically retire if the proof is in a theory where the concept of numbers is not even defined. By classifying proof methods with respect to different mathematical theories on can restrict the computations of the proof planner when considering planning goals belonging to a particular theory.

It seems to be interesting to both implement and experiment with this architecture and furthermore check how far the approach can be coupled with traditional proof planning approaches.

References

[BeSo98] Christoph Benzmüller and Volker Sorge. A Blackboard Architecture for Guiding Interactive Proofs. In F. Giunchiglia, editor, *Proceedings of AIMSA'98*, LNAI 1480, pages 102–114. Springer, Berlin, Germany, 1998.

[BeSo99] Christoph Benzmüller and Volker Sorge. Critical Agents Supporting Interactive Theorem Proving. Seki Report SR-99-02, Computer Science Department, Universität des Saarlandes, 1999.

[Bun88] Alan Bundy. The Use of Explicit Plans to Guide Inductive Proofs. In E. Lusk and R. Overbeek, editors, *Proceedings of CADE–9*, *LNCS* 310, Argonne, USA, 1988. Springer, Berlin, Germany.

[Ome97] The ΩMEGA Group. ΩMega: Towards a Mathematical Assistant. In W. McCune, editor, *Proceedings of CADE–14*, *LNAI* 1249. Springer, Berlin, Germany, 1997.

[SiBy98] Wilfried Sieg and John Byrnes. Normal natural deduction proofs (in classical logic). *Studia Logica*, 60(1):67–106, January1998.