Assertion Application in Theorem Proving and Proof Planning

Quoc Bao Vo[1], Christoph Benzmüller[1] and Serge Autexier[2]
[1]FR Informatik, Universität des Saarlandes, 66041 Saarbrücken, Germany
{bao|chris}@ags.uni-sb.de
[2]DFKI GmbH, 66123 Saarbrücken, Germany
autexier@dfki.de

Our work addresses assertion retrieval and application in theorem proving systems or proof planning systems for classical first-order logic. We propose a distributed mediator *M* between a mathematical knowledge base KB and a theorem proving system TP which is independent of the particular proof and knowledge representation formats of TP and KB and which applies generalized resolution in order to analyze the logical consequences of arbitrary assertions for a proof context at hand. We discuss the connection to proof planning and motivate an application in a project aiming at a tutorial dialogue system for mathematics. This paper is a short version of [9].

1 Proof planning at the assertion level

Due to Huang [6], the notion of assertion comprises mathematical knowledge from a mathematical knowledge base KB such as axioms, definitions, and theorems. Huang argues that an assertion-based representation, i.e. *assertion level*, is just the right level for machine generated proofs to be transformed into before being presented to (human) users. In this paper we argue further that the assertion level can also serve as one of those levels of granularity where knowledge-based proof planning [7] should be based upon.

We are convinced that by planning directly on the assertion level it will be possible to overcome at least some of the identified limitations and problems of proof planning as discussed in [3, 5] — in particular, those, that are caused by an unfortunate intertwining of proof planning and calculus level theorem proving. The perspective we therefore motivate is to consider the assertion level as a well chosen borderline between proof planning and machine oriented methods. Determining the logical consequences of assertions in a proof context is the task of machine oriented methods (in our case generalized resolution). The tasks on top of this level belong to the scope of domain specific proof planning.

In summary, instead of reconstructing natural deduction (ND) proofs to obtain assertion level proofs as suggested by Huang, we propose to directly plan for proofs at the assertion level. This should improve the quality of the resulting proof plans and also facilitate better user interaction.

The development of our ideas revolves around the mathematical assistant system OMEGA [8] and the current initiative in this project to rebuild the system on top of the proof representation framework in [1, 2]. We furthermore employ OMEGA's agent-based search mechanism OANTS [4] for a distributed modeling of our framework and we motivate an application of the approach in a project aiming at a tutorial dialogue system for mathematics.

2 Assertion Application via Generalized Resolution

Depending on the proof context there may be several ways in which an assertion can be used. For instance, the assertion

$$\forall S_1, S_2 : Set.(S_1 \subseteq S_2 \Leftrightarrow \forall x : Element.(x \in S_1 \Rightarrow x \in S_2))$$

allows us to derive: (1) $a \in V$ from $a \in U$ and $U \subseteq V$, (2) $U \not\subseteq V$ from $a \in U$ and $a \notin V$, (3) $\forall x : Element.(x \in U \Rightarrow x \in V)$ from $U \subseteq V$, (4) etc.

Traditional theorem provers or proof planners that operate on calculus level can only achieve such conclusions after a number of proof steps to eliminate the quantifiers and other connectives such as implication and conjunction. We propose an algorithm *Assertion Application* based on a generalized form of resolution in order to achieve such conclusions in one step for arbitrary assertions and proof contexts at hand. The algorithm, which is described in [9], is based on:

(i) The representation of formulas as signed formula trees (SFT); see also [1, 2]. This way we achieve access to the literals of the formulas without breaking them apart.
(ii) The lifting of the resolution idea to directly operate on complementary and unifiable literals in SFTs instead of working on (unintuitive) normal forms.
(iii) The exhaustive application of (ii) to the SFTs generated for the assertion and the formulas of the proof context.

3 Modeling Assertion Agents

In our implementation of mediator *M* we model assertion application in form of distributed search processes employing the OANTS approach [4]. This agent based formalism is the driving force behind distributed proof search in OMEGA.

The general application scenario we have in mind is a theorem prover TP that is connected to an independent mathematical knowledge base KB. TP is focusing a proof task τ consisting of the formula TP has to prove, *viz.* the theorem H, and a set of assumptions it can use to prove H. TP can determine the relevant parts, i.e. the related theories Th, from KB and hand them over to our assertion module *M*. The task of *M* is to compute with respect to proof task τ all possible logical consequences of the available assertions $\mathcal{B}_i$ taken from Th.

We propose to create for each assertion $\mathcal{B}_i$ one associated instance $\mathcal{AG}_{\mathcal{B}_i}$ of a generic *assertion agent* $\mathcal{AG}$. The generic assertion agent $\mathcal{AG}$ is based on our algorithm *AssertionApplication*. Note that this algorithm only depends on the SFT of the focused assertion and a further set of SFTs for the proof context, and both are specified as parameters of *AssertionApplication*. Each assertion agent instance $\mathcal{AG}_{\mathcal{B}_i}$ computes and suggest the logical consequences of $\mathcal{B}_i$ for proof task τ to our module *M* which passes them further to TP.

Depending on the size of KB there could be too many applicable assertions passed to *M* and also too many ways an assertion can be applied. One possibility is to restrict the search space by imposing prerequisites for assertion retrieval. For instance, a proof planner may employ its domain specific meta knowledge to formulate and pass respective context sensitive syntactical filter criteria to the mediator for an efficient preselection by syntactic means.

Proof planning, however, has developed more sophisticated ways to guide and constrain the possible instantiations and applications of assertions. The investigation on how these techniques can optimally be employed on top of our assertion application module *M* are further research.

4 An application: The DIALOG project

Our approach to assertion application is motivated by an application in the DIALOG project as part of the Collaborative Research Center on *Resource adaptive cognitive processes* at Saarland University. The goal of this research project is to develop an experimental prototype system gradually embodying the empirical findings from experiments of mathematical tutoring dialogues. The experimental system will engage in a dialogue in natural language (and perhaps other modes of communication) and help a student to understand and produce mathematical proofs. It is important that such a system is supported by a human oriented mathematical proof development environment and the OMEGA system with its advanced proof presentation and proof planning facilities is a suitable answer to this requirement.

Thus, it is important that the mathematical assistant system be capable of constructing proof (either interactively or automatically) at a human oriented level of granularity for the exercises the student need to solve. It is easy to motivate the design of our assertion application module for this scenario since we do not want the user to puzzle around with the peculiarities of, for instance, logical derivations in sequent or natural deduction calculus. The system capability of applying assertions from a dynamically varying set of assertions is crucial for the project.

Summary: Mediating mathematical knowledge between (web-based) mathematical knowledge bases, mathematical reasoning systems and human users is a mathematical knowledge management task of increasing importance. We sketched a flexible and distributed solution for the subtasks of suggesting the logical consequences of assertions at an intuitive reasoning level.

References

[1] S. Autexier. A proof-planning framework with explicit abstractions based on indexed formulas. *ENTCS*, 58(2), 2001.

[2] S. Autexier. Hierarchical Contextual Reasoning. PhD Thesis, Saarland University, forthcoming.

[3] C. Benzmüller, A. Meier, E. Melis, M. Pollet, and V. Sorge. Proof planning: A fresh start? In *Proc. of the IJCAR 2001 Workshop: Future Directions in Automated Reasoning*, Siena, Italy, 2001.

[4] C. Benzmüller and V. Sorge. OANTS – an open approach at combining interactive and automated theorem proving. In M. Kerber and M. Kohlhase, editors, *Symbolic Computation and Automated Reasoning*, A.K.Peters, 2000.

[5] A. Bundy. A critique of proof planning. In A. C. Kakas and F. Sadri, editors, *Computational Logic. Logic Programming and Beyond*, LNCS 2408, Springer, 2002.

[6] X. Huang. Reconstructing proofs at the assertion level. newblock In A. Bundy, editor, *Proc. CADE-12*, LNAI 814, Springer, 1994.

[7] E. Melis and J. Siekmann. Knowledge-based proof planning. *Articial Intelligence Journal*, 115(1):65–105, 1999.

[8] J. Siekmann et al. Proof development with OMEGA. In A. Voronkov, editor, *Proc. CADE-18*, LNAI 2392, Springer, 2002.

[9] Q. B. Vo, C. Benzmüller, and S. Autexier. An approach to assertion application via generalized resolution. Seki Report SR-03-01, Saarland University, February 2003. `http://www.ags.uni-sb.de/~omega/pub/postscript/SR-03-01.ps.gz`