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

[2]DFKI GmbH, 66123 Saarbrücken, Germany

{bao|chris}@ags.uni-sb.de, autexier@dfki.de

German Research Center for Artificial Intelligence GmbH

The Assertion Level

Motivation

What is an adequate level of abstraction from the logic layer for proof planning and interactive theorem proving?

Natural deduction (ND) and sequent style calculi are not optimally suited!

Working hypotheses: Assertion level reasoning [Huang, CADE12] is adequate.

CORE framework [Autexier, 2003] provides a fruitful basis; full assertion level reasoning based on CORE, however, is still missing.

Assertion Level

Assertions: knowledge-level representations of mathematics such as axioms, definitions, lemmas, theorems, global and local assumptions, ...

Mathematical textbook proofs: abstract away most calculus level derivations when dealing with assertions; decomposition is avoided (treated implicitly).

Traditional theorem provers: normalization of input usually breaks assertion level structure to pieces.

ND and sequent style calculi: assertion application requires explicit decomposition.

Example Assertion

Definition of subset:

$$\forall_{S_1,S_2}.S_1 \subseteq S_2 \Leftrightarrow \forall_x.x \in S_1 \Rightarrow x \in S_2$$

The following assertion level proof steps are immediately derivable:

- $a \in V$ from $a \in U$ and $U \subseteq V$
- $U \not\subseteq V$ from $a \in U$ and $a \notin V$
- $\forall_x.x \in U \Rightarrow x \in V$ from $U \subseteq V$

Natural language: "since a is a member of U and U is a subset of V, according to the definition of subset, a is a member of V."

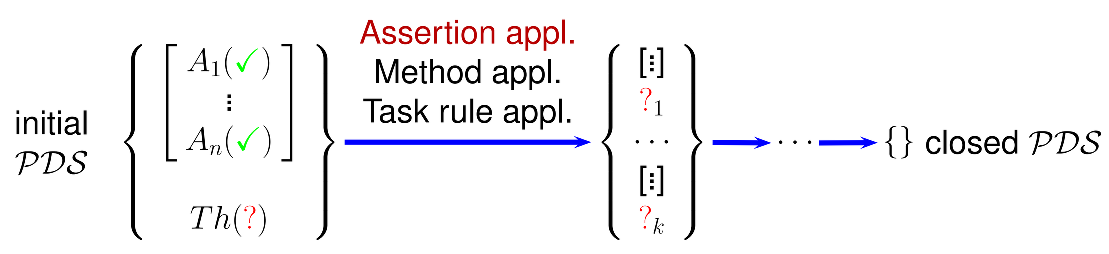
The Task Layer: Reasoning with Assertions

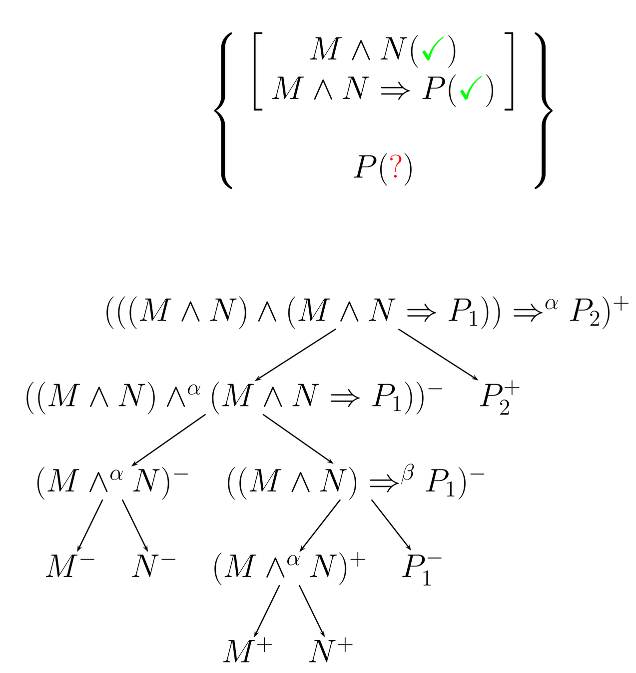
Interactive Theorem Proving
Proof Planning
Agent-based Reasoning

Task Level
(assertion level representation of proof goals)
[Hübner et al., 2003]

Logic Engine CORE
[Autexier, 2003]

- supports flexible assertion level reasoning
- hides logic layer from the user
- avoids decomposition





Goal: support for the following argumentation level

Theorem: $\sqrt{2}$ is irrational.

Proof: (by contradiction)
Assume $\sqrt{2}$ is rational, that is, there exist natural numbers m, n with no common divisor such that $\sqrt{2} = m/n$. Then $n\sqrt{2} = m$, and thus $2n^2 = m^2$. Hence m^2 is even and, since odd numbers square to odds, m is even; say $m = 2k$. Then $2n^2 = (2k)^2 = 4k^2$, that is, $n^2 = 2k^2$. Thus, n^2 is even too, and so is n. That means that both n and m are even, contradicting the fact that they do not have a common divisor.

Required: Module $AssAppl$ that computes and suggests all possible assertion level proof steps for a given task.

$$AssAppl : \text{Tasks} \times \text{Additional-External-Assertions} \rightarrow 2^{\text{Tasks}}$$

Input: A given task (and probably some focus to particular assertions in this task)

Optional additional input: Assertions from external databases that are not imported yet into the proof context (support for dynamic search for applicable lemmas in knowledge-bases)

Output: A list of new tasks that are deducible from the given task by making use of the available assertions.

Generalized Resolution (with signed formula trees)

- Employ signed formulas and uniform notation
- Algorithm for $AssAppl$ employs
  1. resolution on complementary pairs of leaves of tree(s)
  2. manipulation of tree structures
- Do NOT require clausal form (vs. machine oriented methods, resolution)
- Do NOT require decomposition of formulas (vs. ND and sequent calculi)
- Do NOT restrict to refutation context (vs. many machine oriented methods).

Example (F stands for $\forall_x(x \in S_1^? \Rightarrow x \in S_2^?)$.):

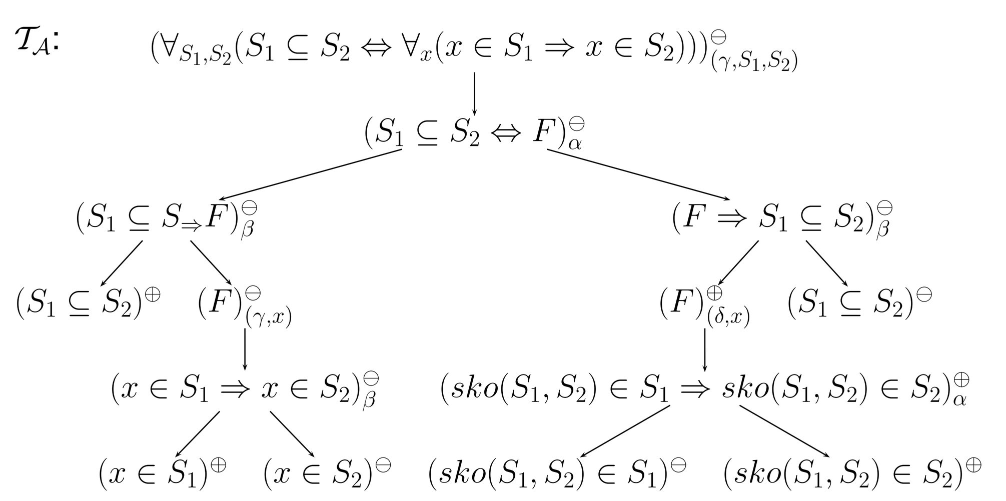
Current Task

$$\left\{\begin{array}{c}\left[\begin{array}{c}\mathcal{T}_A(\checkmark)\\ \mathcal{T}_B(\checkmark)\end{array}\right]\\ \mathcal{T}_G(?)\end{array}\right\}$$

$\mathcal{T}_B$: $(A \subseteq B)^\ominus$

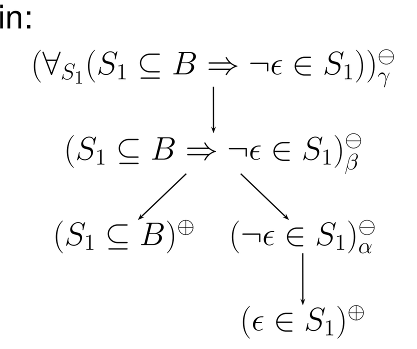
$\mathcal{T}_G$: $(\epsilon \in B)^\oplus$

$\mathcal{T}_A$:



Applying $\mathcal{T}_G$ to $\mathcal{T}_A$ by unifying it to the leaf node $(x \in S_2)^\ominus$, we obtain:

$\mathcal{T}_C$:



We apply $\mathcal{T}_B$ to $\mathcal{T}_C$ and obtain the result tree $\mathcal{T}_D$ and thus task R:

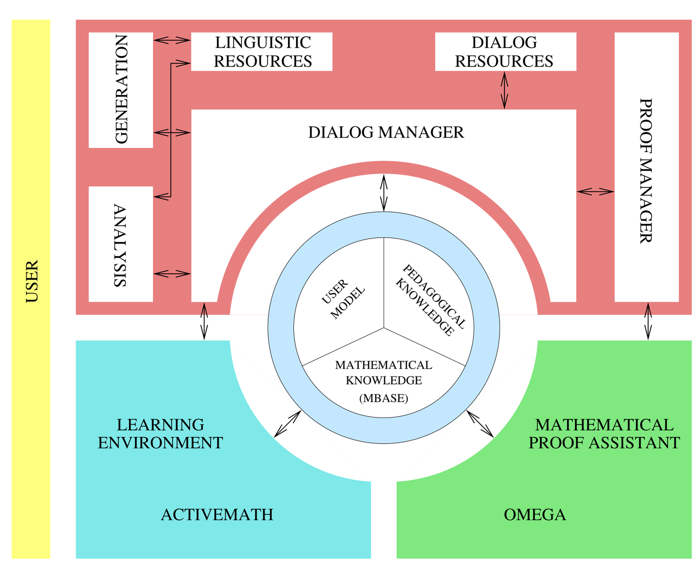
$\mathcal{T}_D$: $(\neg\epsilon \in A)^\ominus_\alpha$ — $(\epsilon \in A)^\oplus$

R: $\left\{\begin{array}{c}\left[\begin{array}{c}\mathcal{T}_A(\checkmark)\\ \mathcal{T}_B(\checkmark)\end{array}\right]\\ \epsilon \in A(?)\end{array}\right\}$

Assertion Level Proof (in ND like presentation)

1.	1;	$\vdash$	*symmetric*(A)	Hyp	
2.	2;	$\vdash$	*symmetric*(B)	Hyp	
3.	1,2;	$\vdash$	$\forall_{x,y}\langle x,y\rangle \in A \Rightarrow \langle y,x\rangle \in A$	Sym-Def	1
4.	1,2;	$\vdash$	$\forall_{x,y}\langle x,y\rangle \in B \Rightarrow \langle y,x\rangle \in B$	Sym-Def	2
5.	5;	$\vdash$	$\langle c_1,c_2\rangle \in A \wedge \langle c_1,c_2\rangle \in B$	Hyp	
6.	1,2,5;	$\vdash$	$\langle c_2,c_1\rangle \in A$	[3]	5
7.	1,2,5;	$\vdash$	$\langle c_2,c_1\rangle \in B$	[4]	5
8.	1,2,5;	$\vdash$	$\langle c_2,c_1\rangle \in A \wedge \langle c_2,c_1\rangle \in B$	And-I	6 7
9.	1,2;	$\vdash$	$\forall_{x,y}\langle x,y\rangle \in A \cap B \Rightarrow \langle y,x\rangle \in A \cap B$	∩-Def	5 8
10.	1,2;	$\vdash$	*symmetric*$(A \cap B)$	Sym-Def	9

An Application: The DIALOG Project



- Tutorial natural language dialog with a mathematical assistant system.
- First empirical findings: adequate support for assertion level reasoning plays a crucial role for the project.