



Using a Blackboard Architecture for Assertion Application in Proof Planning

Christoph Benz Müller, Andreas Meier, and Volker Sorge

Universität des Saarlandes, Saarbrücken, Germany

Proof Planning in the OMEGA System:

- considers theorem proving as planning process
- employs methods as planning operators
- applies mathematical facts stored in data base:
axioms, theorems, lemmas (so-called **assertions**)

During proof planning:

Methods access data base to look up and apply Assertions:

$$\frac{\textit{Premis}}{\textit{Goal}} \textit{Assertion}$$

Classifying residue class structures wrt.

- algebraic structure they form (semi-group, monoid etc.)
- isomorphic structures

Proof obligations, e.g., *Closed*($\mathbb{Z}_5, \lambda x. \lambda y. (x \bar{*} y) \bar{+} \bar{3}_5$)

- \mathbb{Z}_5 : set of integer congruence classes modulo 5
- $\lambda x. \lambda y. (x \bar{*} y) \bar{+} \bar{3}_5$: binary operation on \mathbb{Z}_5

One approach: **apply known theorems from data base**

Example



$$\textit{ClosedConst} : \forall n:\mathbb{Z}. \forall c:\mathbb{Z}_n. \textit{Closed}(\mathbb{Z}_n, \lambda x. \lambda y. c)$$

$$\textit{ClosedFV} : \forall n:\mathbb{Z}. \textit{Closed}(\mathbb{Z}_n, \lambda x. \lambda y. x)$$

$$\textit{ClosedSV} : \forall n:\mathbb{Z}. \textit{Closed}(\mathbb{Z}_n, \lambda x. \lambda y. y)$$

$$\begin{aligned} \textit{Closed}\bar{+} : \forall n:\mathbb{Z}. \forall op_1. \forall op_2. (\textit{Closed}(\mathbb{Z}_n, op_1) \wedge \textit{Closed}(\mathbb{Z}_n, op_2)) \Rightarrow \\ \textit{Closed}(\mathbb{Z}_n, \lambda x. \lambda y. (x \textit{op}_1 y) \bar{+} (x \textit{op}_2 y)) \end{aligned}$$

$$\begin{aligned} \textit{Closed}\bar{-} : \forall n:\mathbb{Z}. \forall op_1. \forall op_2. (\textit{Closed}(\mathbb{Z}_n, op_1) \wedge \textit{Closed}(\mathbb{Z}_n, op_2)) \Rightarrow \\ \textit{Closed}(\mathbb{Z}_n, \lambda x. \lambda y. (x \textit{op}_1 y) \bar{-} (x \textit{op}_2 y)) \end{aligned}$$

$$\begin{aligned} \textit{Closed}\bar{*} : \forall n:\mathbb{Z}. \forall op_1. \forall op_2. (\textit{Closed}(\mathbb{Z}_n, op_1) \wedge \textit{Closed}(\mathbb{Z}_n, op_2)) \Rightarrow \\ \textit{Closed}(\mathbb{Z}_n, \lambda x. \lambda y. (x \textit{op}_1 y) \bar{*} (x \textit{op}_2 y)) \end{aligned}$$

Determining applicable assertions can be difficult

Idea: Separate search for applicable assertions from main proving process

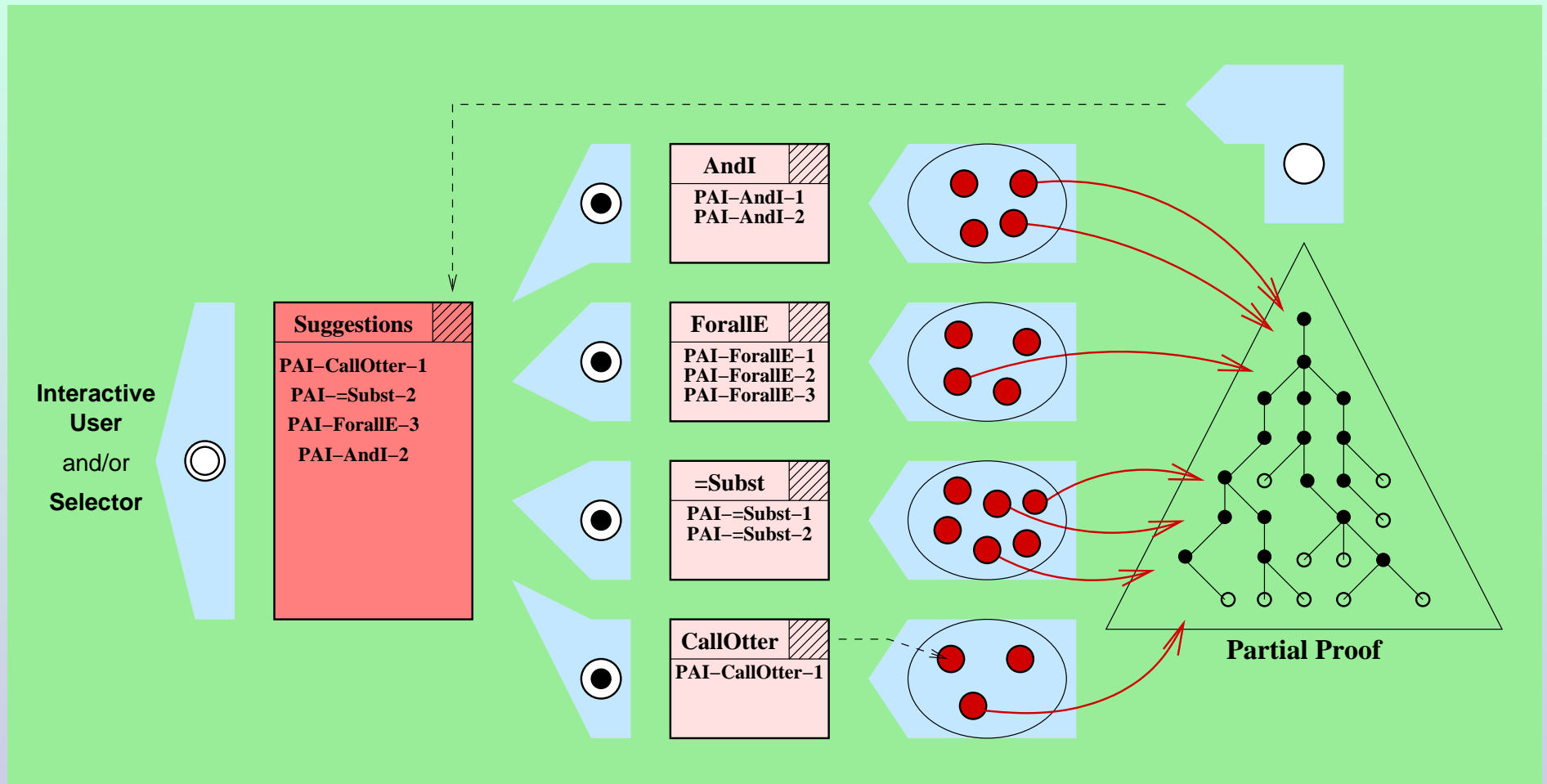
Addressed aspects

- **Concurrency:** parallelizing applicability check to gain efficiency and any-time behavior
- **Flexibility:** parameterize applicability check to employ, for instance, different matching procedures
- **Robustness:** become independent from data base details such as theorem/theory names

- Form clusters of related theorems
- Use two filters (simple & complex)
- Dynamically extend mechanism for new assertions

- Employ the hierarchical blackboard architecture
 Ω -ANTS
- In-built concurrency
- Enables cooperation of knowledge sources
(so-called agents)

Realization – The Idea



Realization – The Mechanism



Applicability check performed in three stages:

Applicability check performed in three stages:

- check for suitable goal (first filter)

Applicability check performed in three stages:

- check for suitable goal (first filter)
- try to match theorems
(with several algorithms in parallel)

Applicability check performed in three stages:

- check for suitable goal (first filter)
- try to match theorems
(with several algorithms in parallel)
- search for possible premises of an applicable theorem

Applicability check performed in three stages:

- Check for suitable goal (first filter)
- Try to match theorems
(with several algorithms in parallel)
- Search for possible premises of an applicable theorem

Formation of clusters done automatically

⇒ **Special predicate to acquire theorems**

Example (continued)



$$\mathcal{G}_{\{\},\{T,P\}}^{\{G\}} = \{G: G \text{ contains the } \textit{Closed} \text{ predicate}\}$$

$$\mathcal{F}_{\{G\},\{P\}}^{\{T\}} = \{T: \text{Conclusion matches } G \text{ with first order matching}\}$$
$$\left\{ \textit{Acquisition}: \text{Conclusion contains } \textit{Closed} \text{ as outermost} \right\}$$

$$\mathcal{F}_{\{G\},\{P\}}^{\{T\}} = \{T: \text{Conclusion matches } G \text{ with special algorithm}\}$$
$$\left\{ \textit{Acquisition}: \text{Conclusion contains } \textit{Closed} \text{ as outermost} \right\}$$

$$\mathcal{G}_{\{G,T\},\{\}}^{\{P\}} = \{P: \text{The nodes matching the premises of } T\}$$

Example (continued)



Goal: $Closed(\mathbb{Z}_5, \lambda x. \lambda y. (x \bar{*} y) \bar{+} \bar{3}_5)$

Closed

Goal contains *Closed* predicate?

Example (continued)



Goal: $Closed(\mathbb{Z}_5, \lambda x. \lambda y. (x \bar{*} y) \bar{+} \bar{3}_5)$

Closed
(Goal:Closed(...))

$ClosedConst : Closed(\mathbb{Z}_n, \lambda x. \lambda y. c)$

$Closed\bar{+} : \dots \lambda x. \lambda y. (\dots \bar{+} \dots)$

$ClosedFV : \dots \lambda x. \lambda y. x$

$Closed\bar{-} : \dots \lambda x. \lambda y. (\dots \bar{-} \dots)$

$ClosedSV : \dots \lambda x. \lambda y. y$

$Closed\bar{*} : \dots \lambda x. \lambda y. (\dots \bar{*} \dots)$

with FO matching?

with special algorithm?

Example (continued)



Goal: $Closed(\mathbb{Z}_5, \lambda x. \lambda y. (x \bar{*} y) \bar{+} \bar{3}_5)$

Closed
(Goal: Closed(...))
(Goal: Closed(...), Thm: $Closed \bar{+}$)

Nodes matching the premises of Thm?

Example (continued)



Goal: $Closed(\mathbb{Z}_5, \lambda x. \lambda y. \bar{3}_5)$

Closed

Goal contains *Closed* predicate?

Example (continued)



Goal: $Closed(\mathbb{Z}_5, \lambda x. \lambda y. \bar{3}_5)$

Closed
(Goal:Closed(...))

$ClosedConst : Closed(\mathbb{Z}_n, \lambda x. \lambda y. c)$

$Closed\bar{+} : \dots \lambda x. \lambda y. (\dots \bar{+} \dots)$

$ClosedFV : \dots \lambda x. \lambda y. x$

$Closed\bar{-} : \dots \lambda x. \lambda y. (\dots \bar{-} \dots)$

$ClosedSV : \dots \lambda x. \lambda y. y$

$Closed\bar{*} : \dots \lambda x. \lambda y. (\dots \bar{*} \dots)$

with FO matching?

with special algorithm?

Example (continued)



Goal: $Closed(\mathbb{Z}_5, \lambda x. \lambda y. \bar{3}_5)$

Closed
(Goal:Closed(...)) (Goal: Closed(...), Thm: <i>ClosedConst</i>)

Nodes matching the premises of Thm?

1. Interactive theorem proving

- approach also supports interactive theorem proving

- Ω -ANTS:

ranking and suggestion of applicable theorems to user

2. Retrieval from other data bases

- approach not restricted to particular data base

- also possible (not implemented yet):

retrieval from distributed data base via www

Alternative use of Ω -ANTS:

- one (or several) agent for each single theorem
- unique agent to search for matching supports

Use of alternative techniques

- for knowledge base retrieval:
hashing or term indexing techniques

Use of alternative techniques

- for knowledge base retrieval:
hashing or term indexing techniques
- for assertion matching:
higher-order pattern matching
higher-order (pre-)unification
theorem proving

Evaluation:

- compare our approach with some of the alternative techniques
- conduct a large case study

Evaluation:

- compare our approach with some of the alternative techniques
- conduct a large case study

Integration:

- use our approach in combination with / as part of a more advanced knowledge base