An Agent based Approach to (Mathematical) Reasoning

Christoph Benzmüller

joint work with

Mateja Jamnik, Manfred Kerber, and Volker Sorge

Fachbereich Informatik, Universität des Saarlandes
School of Computer Science, The University of Birmingham

with thanks to: Alan Bundy, Michael Fisher, Andreas Franke, Malte Hübner, Andrew Ireland, Jürgen Zimmer

ARW/AISB'01, The University of York, March 23th 2001

Motivation *Cognitive Perspective*

To solve complex problems in mathematics or engineering

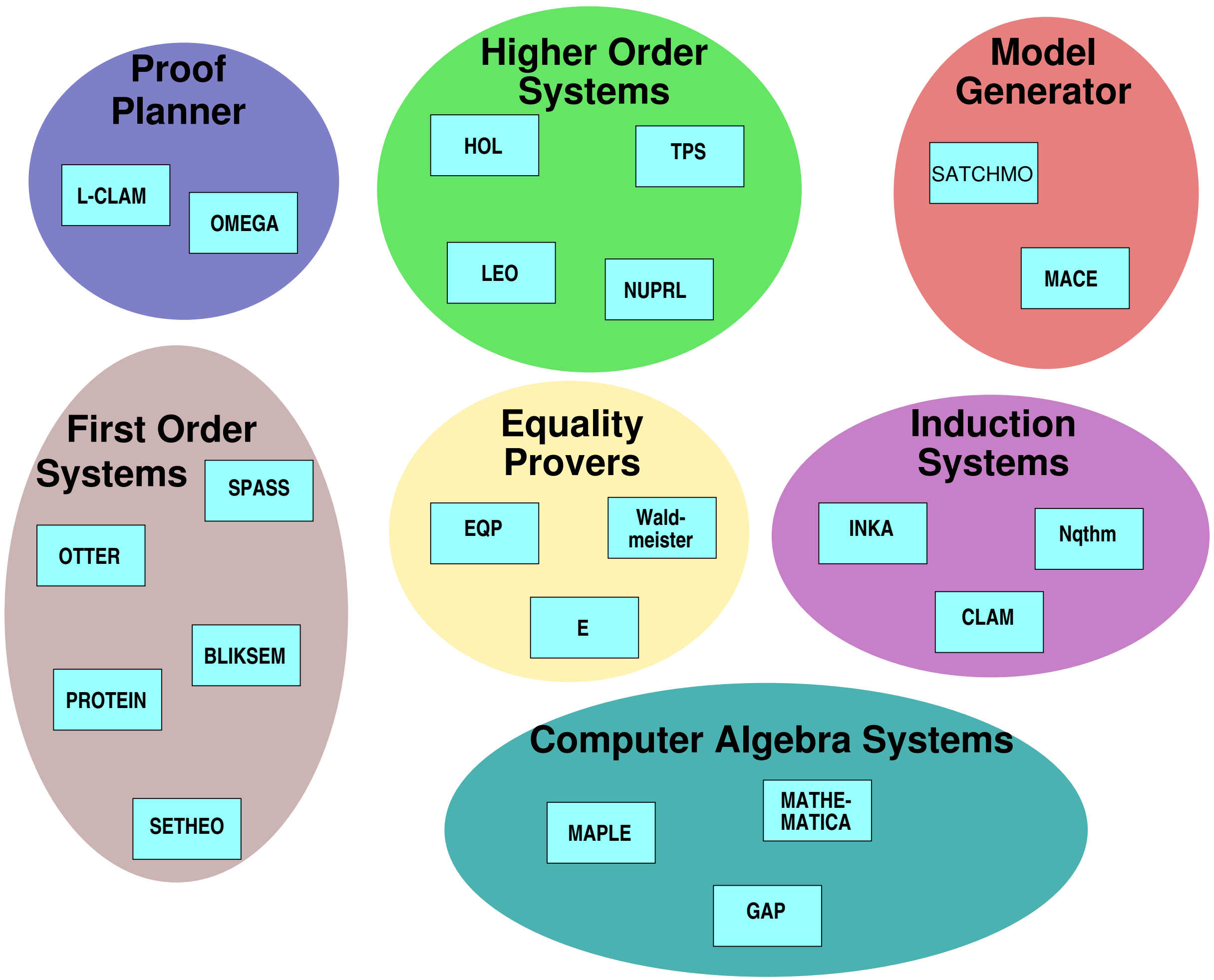
- different specialists may have to bring in their expertise and cooperate
- a communication language is required

A single mathematician

- possesses a large repertoire of specialised reasoning and problem solving techniques
- uses experience and intuition to flexibly combine them in an appropriate way

What is the best architecture for mathematical reasoning systems? **an agent based architecture?**

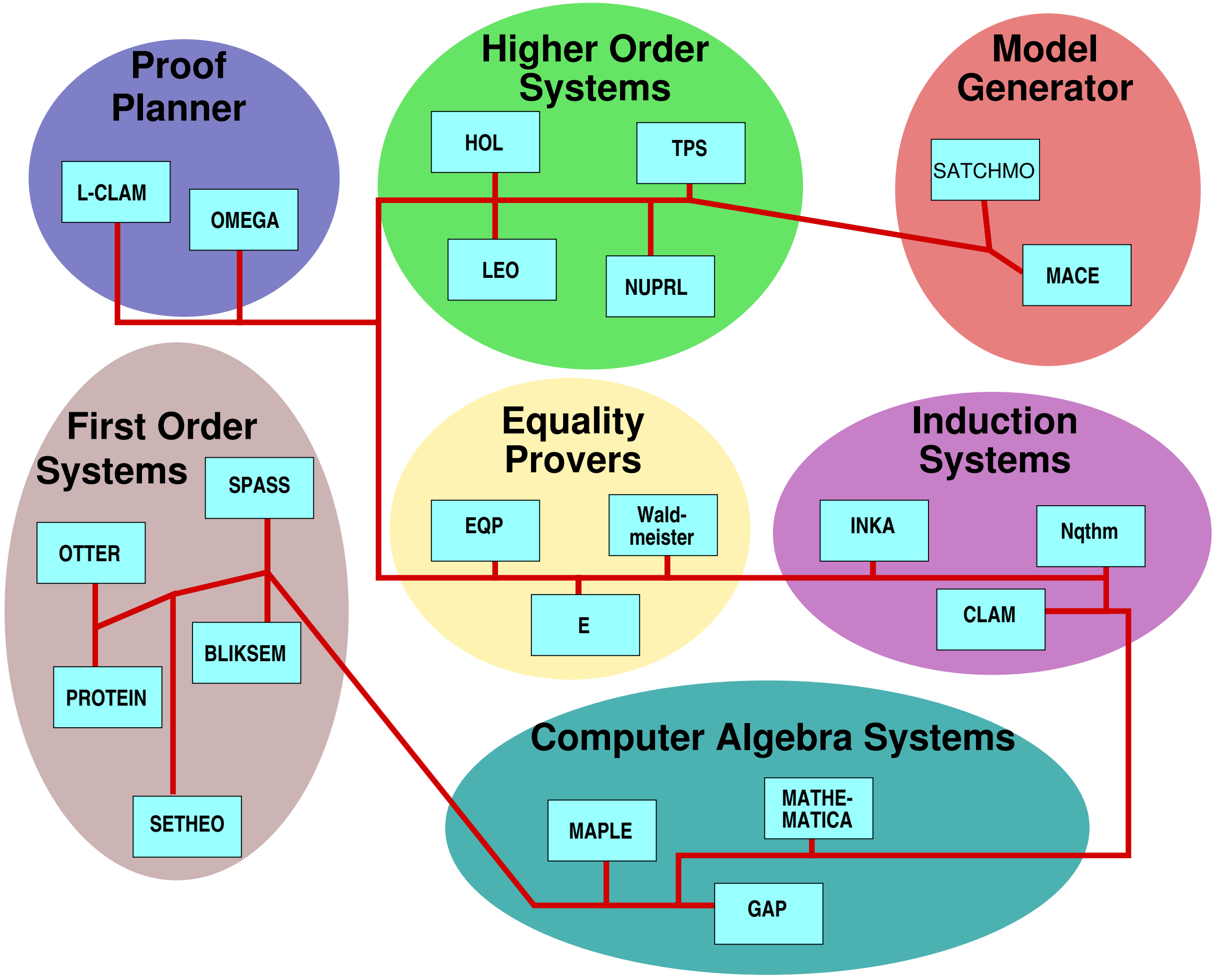
Motivation *Existing Systems*



- heterogeneous
- different niches

Motivation

Existing Systems



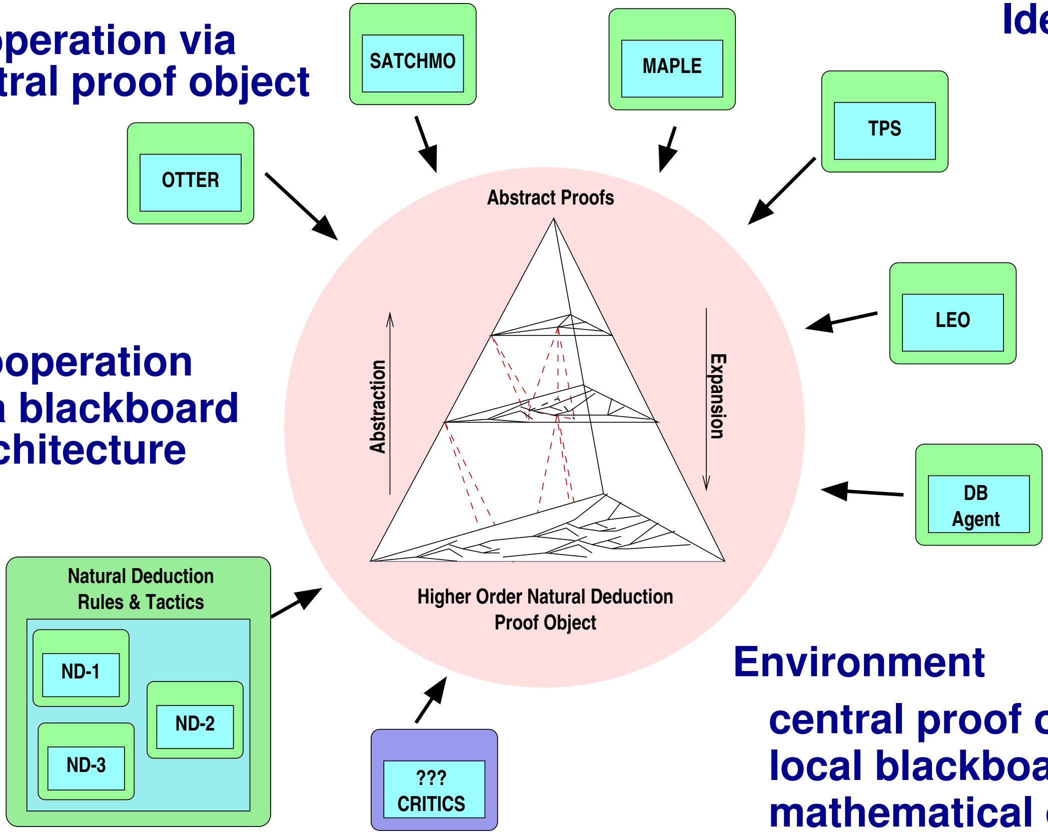
- heterogeneous
- different niches
- system networks:
  MATHWEB, PROSPER
- communication problem
- inflexible applications

How to realise a flexible interplay?

Motivation

Flexible Integration

Cooperation via central proof object



Cooperation via blackboard architecture

ND-Backbone

Idea
- **decentralised control**

Agents
- **reactive**
- **proactive**
- **heterogenous**
- **simple & complex**
- **cooperative & competetive**
- **distributed via MathWeb**
- **run-time definable**
- **resource adapted**

Environment
- **central proof object**
- **local blackboards**
- **mathematical database**

Example

HO- and FO-ATP

Higher Order ATP with LEO

C_1: favourite-numbers($\lambda x.$ odd(x) $\wedge$ square(x))

unifies (semantically) with

C_2: $\neg$favourite-numbers($\lambda x.$ square(x) $\wedge$ (square(x) $\Rightarrow$ odd(x)))

iff the following set of first order clauses can be contradicted

First Order ATP with OTTER

square(N)

$\neg$odd(N) $\vee$ $\neg$square(N)

odd(N) $\vee$ $\neg$square(N)

Example

HO- and FO-ATP

fn $=$ **favourite-numbers**

o $=$ **odd**

s $=$ **square**

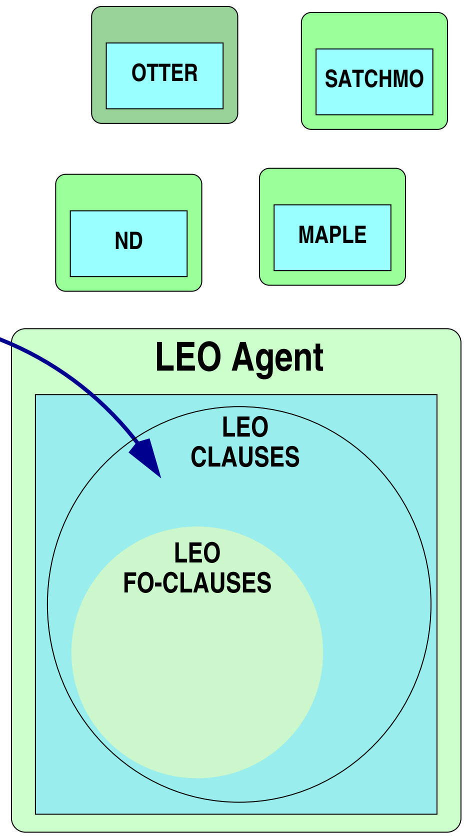
$C_1 : \mathbf{fn}(\lambda x.\, \mathbf{o}(x) \wedge \mathbf{s}(x))$

$C_2 : \neg\mathbf{fn}(\lambda x.\, \mathbf{s}(x) \wedge (\mathbf{s}(x) \Rightarrow \mathbf{o}(x)))$

$$\mathbf{fn}(\lambda x.\, o(x) \wedge \mathbf{s}(x))$$

$$\vdots\ ?$$

$$\mathbf{fn}(\lambda x.\, \mathbf{s}(x) \wedge (\mathbf{s}(x) \Rightarrow o(x)))$$



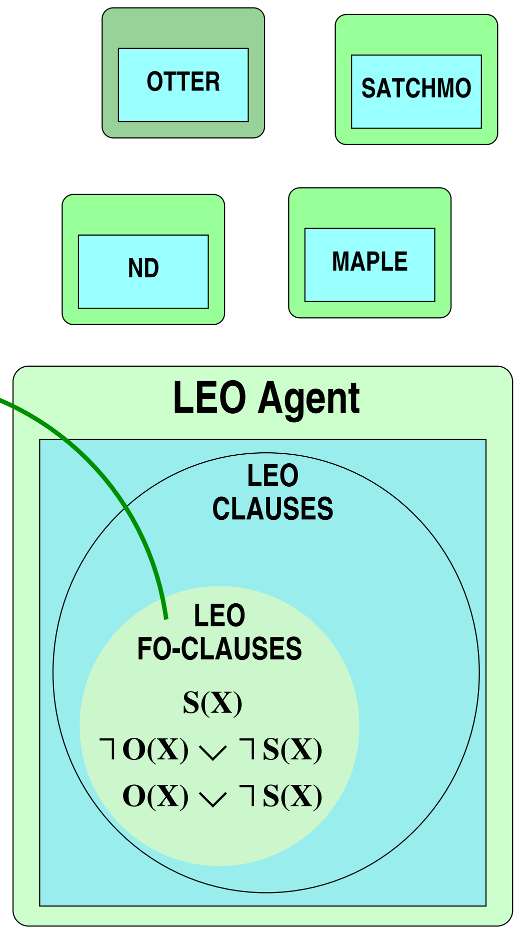
Example

HO- and FO-ATP

$$\frac{\begin{array}{c}\textsf{fn}(\lambda x.\, \textsf{o}(x) \wedge \textsf{s}(x)) \\ \neg\textsf{fn}(\lambda x.\, \textsf{s}(x) \wedge (\textsf{s}(x) \Rightarrow \textsf{o}(x)))\end{array}}{\textsf{s}(N) \wedge (\neg\textsf{o}(N) \vee \neg\textsf{s}(N)) \wedge (\textsf{o}(N) \vee \neg\textsf{s}(N))}\ \mathit{LEO}$$

$$\vdots\ ?$$

$$\frac{\bot}{\textsf{fn}(\lambda x.\, \textsf{s}(x) \wedge (\textsf{s}(x) \Rightarrow \textsf{o}(x)))}\ \mathit{indirect}$$



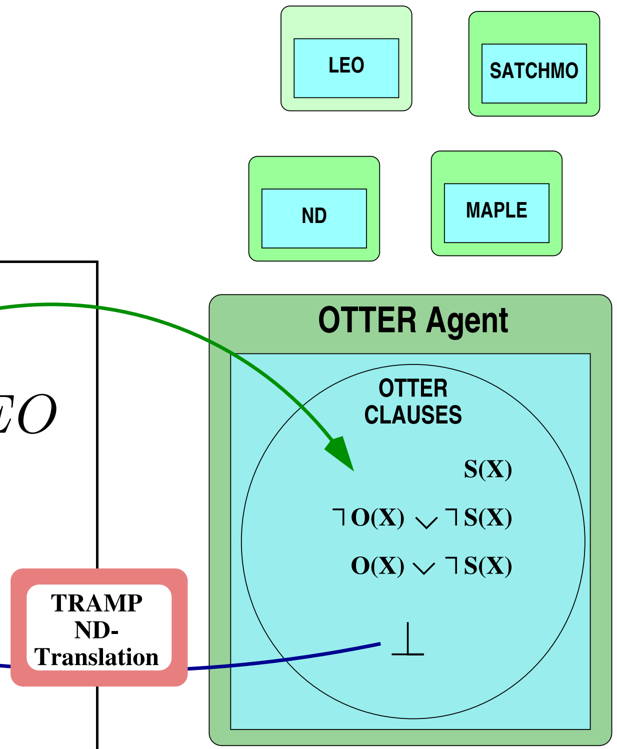
Example

HO- and FO-ATP

$$\frac{\mathbf{fn}(\lambda x.\, \mathbf{o}(x) \land \mathbf{s}(x)) \qquad \neg\mathbf{fn}(\lambda x.\, \mathbf{s}(x) \land (\mathbf{s}(x) \Rightarrow \mathbf{o}(x)))}{\mathbf{s}(N) \land (\neg\mathbf{o}(N) \lor \neg\mathbf{s}(N)) \land (\mathbf{o}(N) \lor \neg\mathbf{s}(N))}\ LEO$$

Translated OTTER Proof

$$\frac{\bot}{\mathbf{fn}(\lambda x.\, \mathbf{s}(x) \land (\mathbf{s}(x) \Rightarrow \mathbf{o}(x)))}\ indirect$$

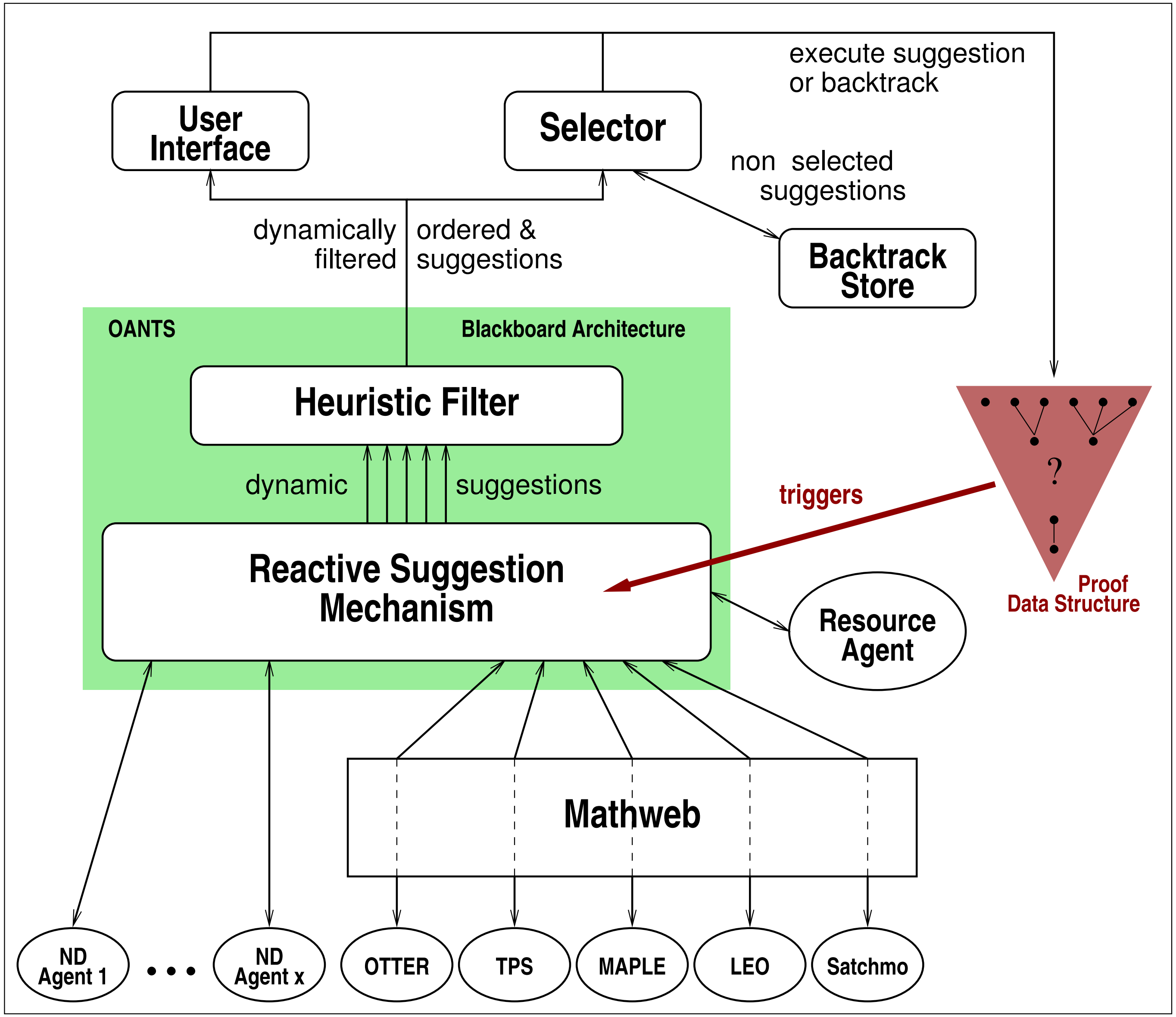


ΩANTS — *Agent based Theorem Prover*

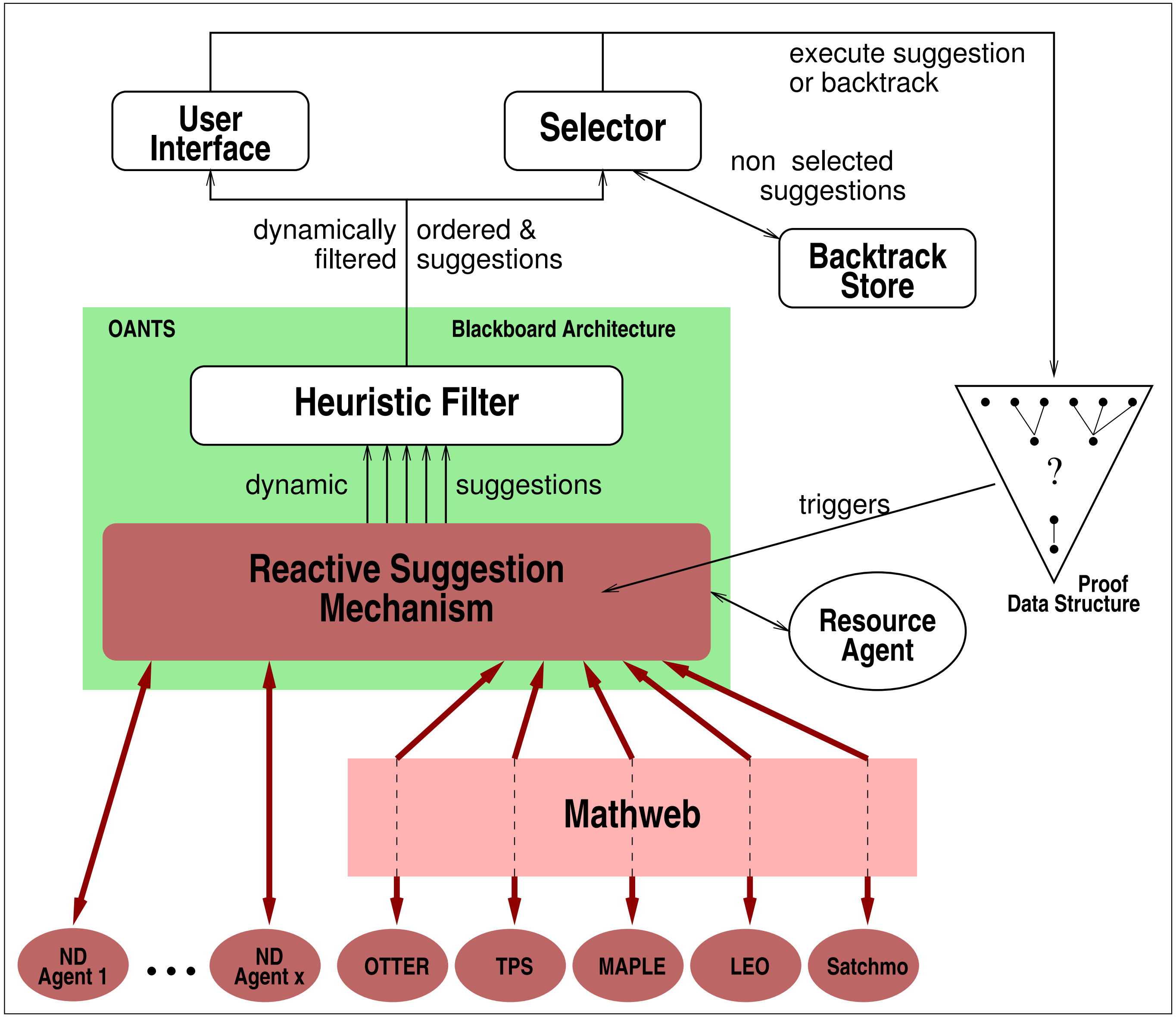
Main Components:

- ΩMEGA-System (Saarbrücken); ΩMEGA's proof data structure PDS
- ΩANTS blackboard architecture
- MATHWEB-System developed in Saarbrücken
- Various external systems integrated to ΩMEGA
- Translation modules like
  - TRAMP (FO resolution $\Longrightarrow$ ND)
  - SAPPER (CAS $\Longrightarrow$ ND)
- Calculus NIC (Carnegie Mellon University) for efficient natural deduction proof search

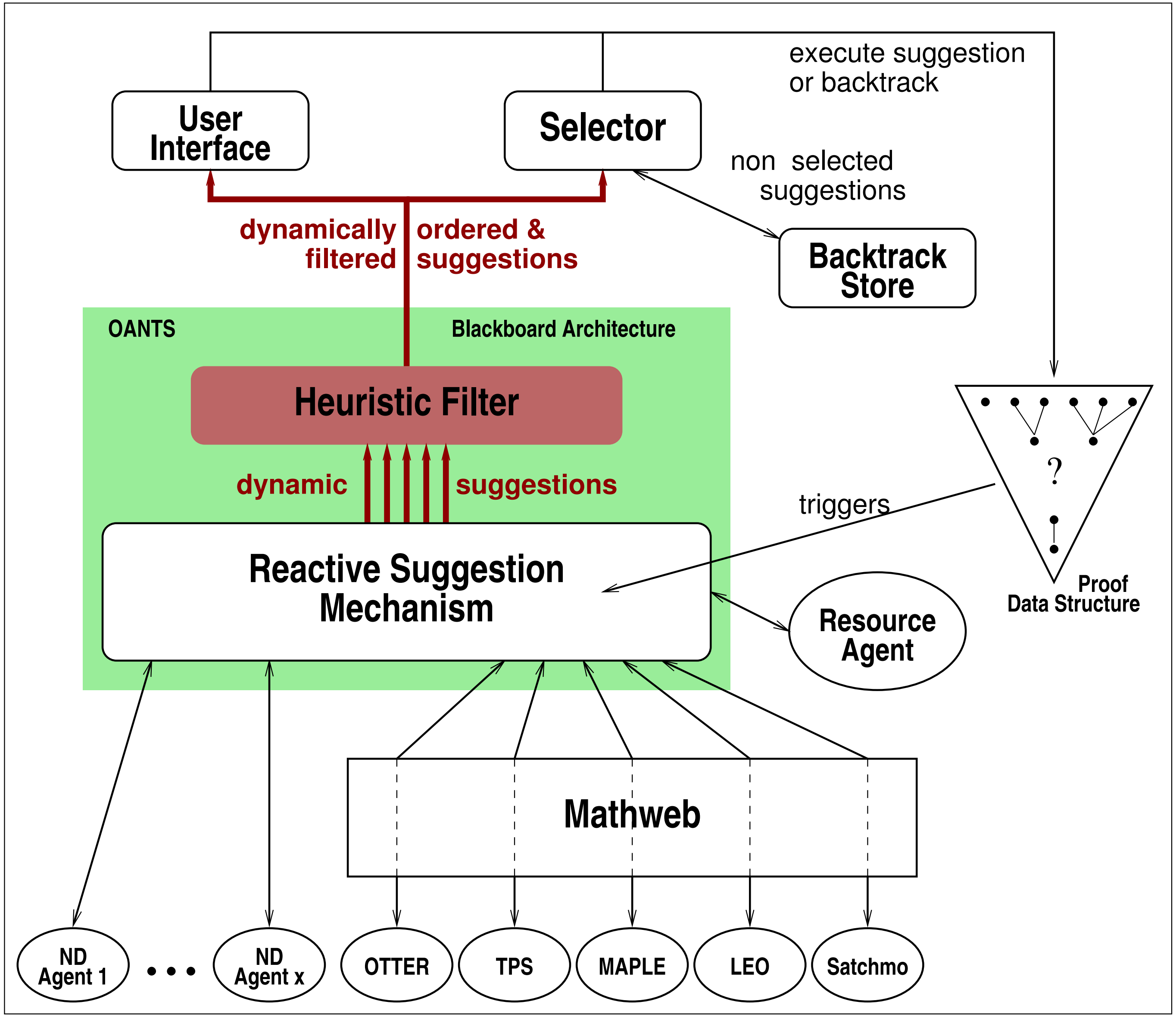
ΩANTS Agent based Theorem Prover



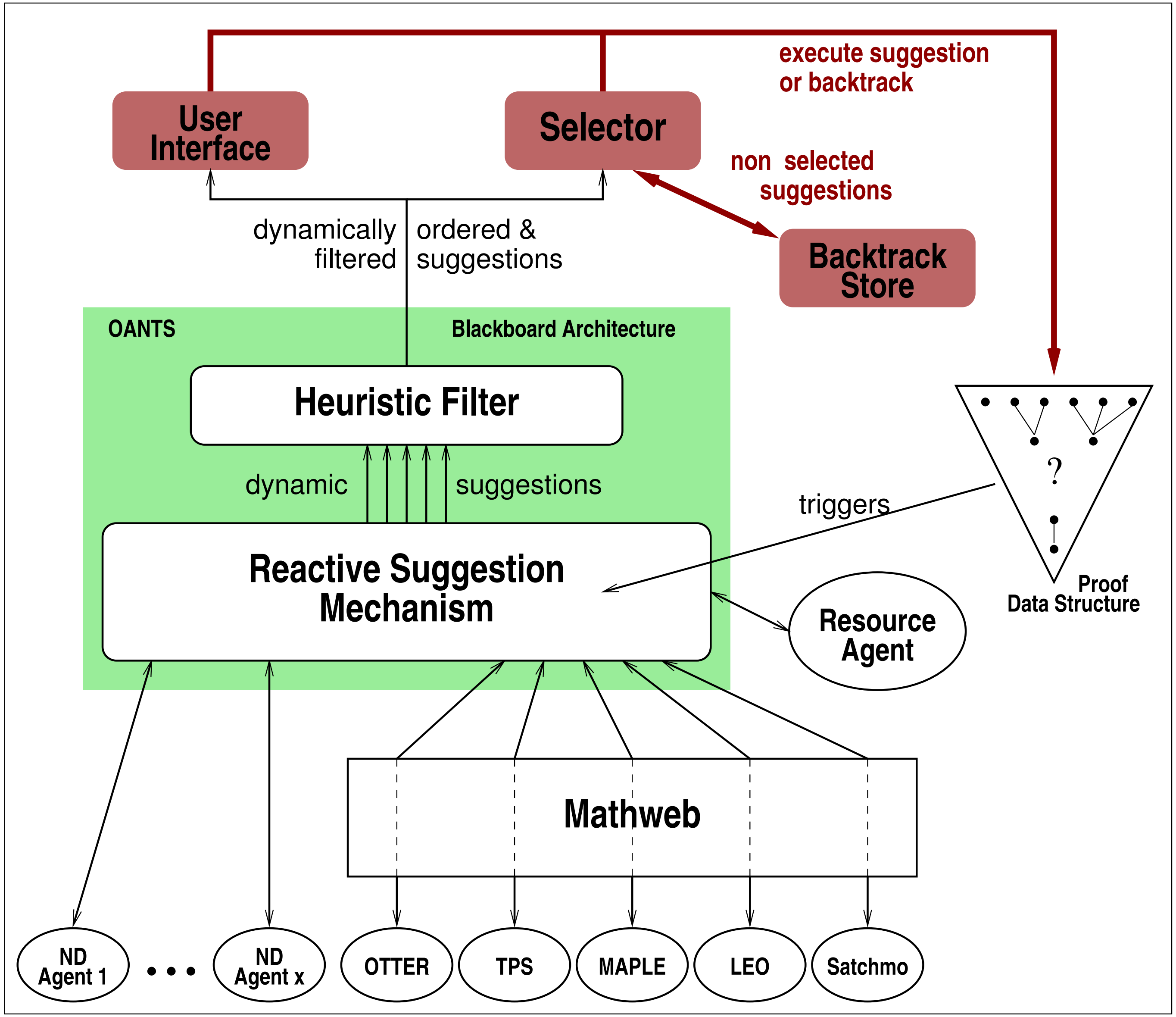
ΩANTS Agent based Theorem Prover



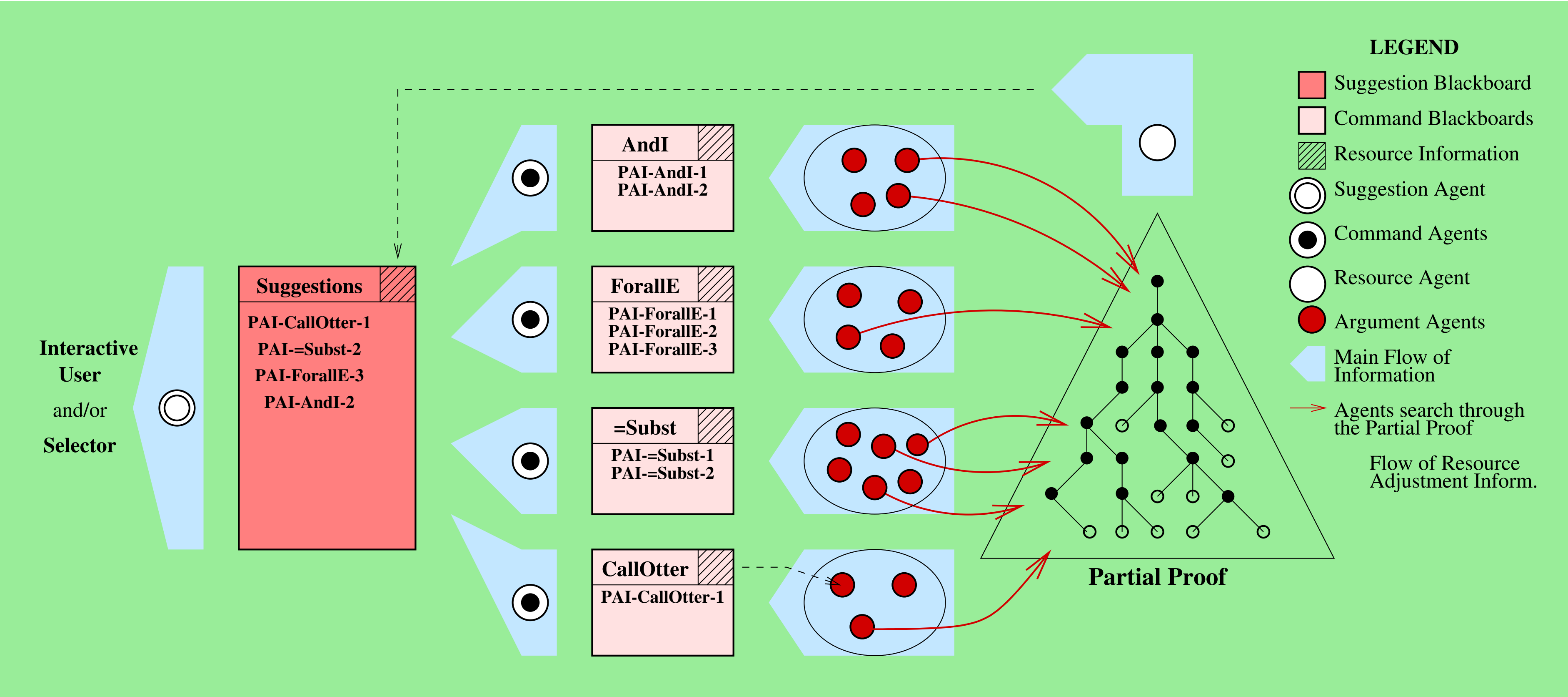
ΩANTS Theorem Prover



ΩANTS Theorem Prover



ΩANTS — Blackboard Architecture



Task: Which commands (rules, tactics, external systems) are promising in the current proof state?

$$\forall x.\,\mathsf{fn}(x)$$
$$\mathsf{fn}(N) \Rightarrow \mathsf{o}(N) \wedge \mathsf{s}(N)$$
$$\mathsf{o}(N)$$
$$\mathsf{s}(N)$$
$$\vdots\ ?$$
$$\mathsf{o}(N) \wedge \mathsf{s}(N)$$

$$\frac{\texttt{Left}\!:\!A \quad \texttt{Right}\!:\!B}{\texttt{Conj}\!:\!A \wedge B}\ \wedge E$$

$$\frac{\texttt{Ant}\!:\!A \quad \texttt{Imp}\!:\!B \Rightarrow C}{\texttt{Succ}\!:\!C}\ \texttt{mp-mod}\ (A \rightarrow B)$$

∧E-Agent-1
search for: Conj
required:
excluded: Left,Right

∧E-Agent-2
search for: Conj
required: Lef
excluded: Right

mp-mod-Agent-1
search for: Succ,Imp
required:
excluded: Ant

$$\frac{\texttt{Left}\!:\!A \quad \texttt{Right}\!:\!B}{\texttt{Conj}\!:\!\mathsf{o}(N) \wedge \mathsf{s}(N)}\ \wedge E$$

$$\frac{\texttt{Ant}\!:\!A \quad \texttt{Imp}\!:\!\mathsf{fn}(N) \Rightarrow \mathsf{o}(N) \wedge \mathsf{s}(N)}{\texttt{Succ}\!:\!\mathsf{o}(N) \wedge \mathsf{s}(N)}\ \texttt{mp-mod}\ (A \rightarrow B)$$

$$\forall x.\, \mathbf{fn}(x)$$
$$\mathbf{fn}(N) \Rightarrow \mathbf{o}(N) \wedge \mathbf{s}(N)$$
$$\mathbf{o}(N)$$
$$\mathbf{s}(N)$$
$$\vdots\ ?$$
$$\mathbf{o}(N) \wedge \mathbf{s}(N)$$

$$\frac{\texttt{Left}\!:\!A \quad \texttt{Right}\!:\!B}{\texttt{Conj}\!:\!\mathbf{o}(N) \wedge \mathbf{s}(N)} \wedge E$$

$$\frac{\texttt{Ant}\!:\!A \quad \texttt{Imp}\!:\!\mathbf{fn}(N) \Rightarrow \mathbf{o}(N) \wedge \mathbf{s}(N)}{\texttt{Succ}\!:\!\mathbf{o}(N) \wedge \mathbf{s}(N)} \text{ mp-mod } (A \rightarrow B)$$

∧E-Agent-2
search for: Left
required: Conj
excluded:

∧E-Agent-3
search for: Right
required: Conj
excluded:

mp-mod-Agent-2
search for: Ant
required: Succ
excluded:

$$\frac{\texttt{Left}\!:\!\mathbf{o}(N) \quad \texttt{Right}\!:\!\mathbf{s}(N)}{\texttt{Conj}\!:\!\mathbf{o}(N) \wedge \mathbf{s}(N)} \wedge E$$

$$\frac{\texttt{Ant}\!:\!\forall x.\, \mathbf{fn}(x) \quad \texttt{Imp}\!:\!\mathbf{fn}(N) \Rightarrow \mathbf{o}(N) \wedge \mathbf{s}(N)}{\texttt{Succ}\!:\!\mathbf{o}(N) \wedge \mathbf{s}(N)} \text{ mp-mod } (A \rightarrow B)$$

ΩANTS — *Various Further Aspects*

- **Declarative agent specification language**

  uniform way to define new agents
  run-time modifiability of agent societies

- **Attempt to a formal semantics**

  mapping agent declarations to simply typed λ-calculus
  some properties of agents and agent societies can be modelled

- **Self-evaluation of agents**

  knowledge about their own performance
  this knowledge is broadcasted via the blackboards
  explicit resource reasoning on informed layer

ΩANTS Resource Adapted Behaviour

Selection clock speed:

determines resource computation time ct for the agents

- ct high — automatic proof by single ATP
- ct medium — cooperative proof
- ct low — (unsuccessful) attack at ND level

First experiments (resource adaptivity in interactive sessions)

agents decide to get inactive/active wrt varying clock speed

Results

Example Classes

Ex1 **higher order ATP and first order ATP**

$\forall x, y, z. (x = y \cup z) \Leftrightarrow (y \subseteq x \wedge z \subseteq x \wedge \forall v. (y \subseteq v \wedge z \subseteq v) \Rightarrow (x \subseteq v))$

Ex2 **ND based TP, propositional ATP, and model generation**

$\forall x. \forall y. \forall z. ((x \cup y) \cap z) = (x \cap z) \cup (y \cap z)$ **10000 Examples**

$\forall x. \forall y. \forall z. ((x \cup y) \cup z) = (x \cap z) \cup (y \cap z)$ **988 valid / 9012 invalid**

Ex3 **computer algebra systems and higher order ATP**

$\{x | x > gcd(10,8) \wedge x < lcm(10,8)\} = \{x | x < 40\} \cap \{x | x > 2\}$

Ex4 **ND and tactical based TP, first-order ATP**

$\ldots$ *group-definition-1* $\ldots \Leftrightarrow \ldots$ *group-definition-2* $\ldots$

Results

Ex2: ND, PL-ATP, model generation

Conc.	$\vdash \forall x.\forall y.\forall z.((x \cup y) \cap z) = (x \cap z) \cup (y \cap z)$	(**Forall-I L1**)
	$\ldots$	
L3.	$\vdash ((X \cup Y) \cap Z) = (X \cap Z) \cup (Y \cap Z)$	(**Set-Ext L4**)
L4.	$\vdash \forall e.e \in ((X \cup Y) \cap Z) \leftrightarrow e \in (X \cap Z) \cup (Y \cap Z)$	(**Forall-I L5**)
L5.	$\vdash E \in ((X \cup Y) \cap Z) \leftrightarrow E \in (X \cap Z) \cup (Y \cap Z)$	(**Def L6**)
	$\ldots$	
L8.	$\vdash ((E \in X \vee E \in Y) \wedge E \in Z) \leftrightarrow ((E \in X \wedge E \in Z) \vee (E \in Y \wedge E \in Z))$	(**OTTER**)

Theorem

Results

Ex2: ND, PL-ATP, model generation

Conc.	$\vdash \forall x. \forall y. \forall z. ((x \cup y) \cup z) = (x \cap z) \cup (y \cap z)$	(**Forall-I L1**)
	$\dots$	
L3.	$\vdash ((X \cup Y) \cup Z) = (X \cap Z) \cup (Y \cap Z)$	(**Set-Ext L4**)
L4.	$\vdash \forall e. e \in ((X \cup Y) \cup Z) \leftrightarrow e \in (X \cap Z) \cup (Y \cap Z)$	(**Forall-I L5**)
L5.	$\vdash E \in ((X \cup Y) \cup Z) \leftrightarrow E \in (X \cap Z) \cup (Y \cap Z)$	(**Def L6**)
	$\dots$	
L8.	$\vdash ((E \in X \vee E \in Y) \vee E \in Z) \leftrightarrow ((E \in X \wedge E \in Z) \vee (E \in Y \wedge E \in Z))$	(**SATCHMO**)

Counter Model: $G \in Z \wedge G \notin X \wedge G \notin Y$

Results

Ex3: ND based TP, CAS and HO-ATP

$$\{x|x > gcd(10,8) \wedge x < lcm(10,8)\} = \{x|x < 40\} \cap \{x|x > 2\}$$

Conc.	$\vdash (\lambda x_\blacksquare x > gcd(10,8) \wedge x < lcm(10,8)) = (\lambda x_\blacksquare x < 40) \cap (\lambda x_\blacksquare x > 2)$	(**CAS L1**)
L1.	$\vdash (\lambda x_\blacksquare x > 2 \wedge x < 40) = (\lambda x_\blacksquare x < 40) \cap (\lambda x_\blacksquare x > 2)$	(**Def L3**)
L3.	$\vdash (\lambda x_\blacksquare x > 2 \wedge x < 40) = (\lambda x_\blacksquare x < 40 \wedge x > 2)$	(**LEO**)

Related Work

- **Overview of parallel & distributed theorem proving [Bonacina 2000]**
- TECHS **approach [Denzinger and Fuchs 1999]**
  heterogeneous first-order systems, filtered exchange of clauses
  no higher-order systems and no CAS
  no explicit proof object
  no user orientation
- **Open approach to concurrent theorem proving [Fisher 1997]**
- **Multi agent proof-planning [Fisher and Ireland 1998]**
- **Agent planning architectures, e.g. [Wilkins and Myers 1998]**
- . . . **agent based architectures** . . .

Conclusion

Agent based architecture

application to mathematical reasoning (HO)
with heterogeneous external systems
flexible integration of new agents
cooperation & competition
supports automation and interaction
abstract inferences & low level ND inferences
resource adapted & adaptive

Typical applications

domains where different specialist systems are required
higher-order examples with first-oder subtasks
exploration of new domains

Architecture not restricted to theorem proving

central proof object $\longrightarrow$ knowledge base
proof rules, tactics, external systems $\longrightarrow$ production rules

Problems and Future Work

Long-term goals

solving the communication problem

choice of interlingua between agents

adding or-parallelism

full integration with proof planning

critical (reflecting) agents

dynamic clustering of agents

experiment: iterated learning of tactics/methods

Short-term goals

employ counterexample information for early backtracking

counterexamples by Venn diagrams

various technical problems (copying of PDS)