Resource Guided Concurrent Deduction

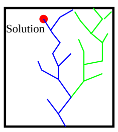
C. Benzmüller*, M. Jamnik*, M. Kerber*, and V. Sorge**

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

Focused Search

"... when he [the mathematician] does not succeed in guessing the whole answer, [he] tries to guess some part of the answer, some feature of the solution, some approach to the solution, or some feature of an approach to the solution. Then he seeks to expand his guess, and so he seeks to adapt his guess to the best information he can get at the moment." [Pólya]

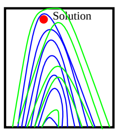
Heuristic Search



Proof Planning

A human oriented approach employing heuristic search at proof method level, i.e. on a more abstract and informed layer.

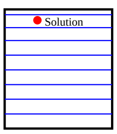
Focused Search



Resource Guided Concurrent Deduction

An attempt to combine the benefits of both approaches. Preference to the integrated reasoning components is given due to the available resources (knowledge, time, ...)

Breadth First Search



Traditional Automated Theorem Proving

A machine oriented approach typically aiming at exhaustive (complete) search through the search space at inference rule level.

Search	Reliability	Cost
Breadth First	robust	expensive
Heuristic	brittle	inexpensive
Focused	less brittle	moderately expensive

Architecture

"Those who have an excessive faith in their ideas are not well fitted to make discoveries." [Claude Bernard]

"In order to invent, one must think aside." [Souriau]

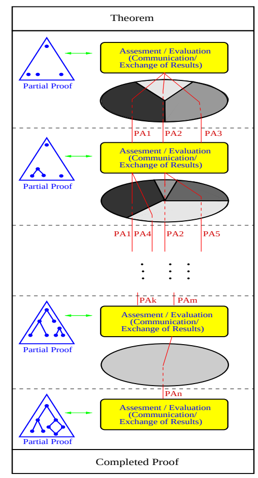
"Therefore, we see that the unconscious has the important property of being manifold; several and probably many things can and do occur in it simultaneously. This contrasts with conscious ego which is unique. We also see that this multiplicity of the unconscious enables it to carry out a work of synthesis." [Hadamard]

Idea

Run integrated reasoning components concurrrently but give prefenrence to the most promising ones with the help of an appropriate resource distribution. Periodically assesses the state of the proof search process, evaluate the progress, and choose a new promising direction for the further search, and redistribute the resources accordingly.

Resource Guided Reasoning Cycle

1. Assessment & evaluation of the proof progress
   - are the resulting partial proof plans new?
   - do they contain more simplified expressions?
   - do they contain simpler open goals?
   - do they contain open goals similar to lemmata in the database?
2. Selection of promising results
   - choose the most promising partial proof plan according to the above criteria and make it the new actual proof plan
   - save the best of the remaining results for backtracking
3. Redistribution of available resources
   - what is the logic language the focused problem belongs to?
   - what is the mathematical theory the focused problem belongs to?
   - does the database provide information which a reasoner already successfully used to solve similar problems in the past?



Mathematical Mind

promising search directions

thinking aside

proof techniques

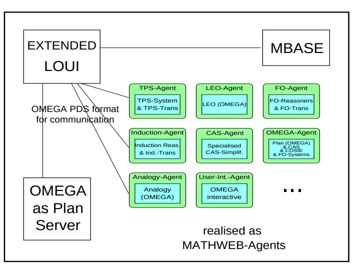
deliberative & reactive

Planning Agents

Approach

Adapt the Multi-Agent Planning (MPA) approach [Wilkins&Myers 1998] to the proof planning domain.

Employ the mathematical database MBASE as domain server, OMEGA as plan server and an extension of OMEGA/LOUI as planning cell manager. As planning cells use the algorithms and external reasoners already provided by the OMEGA/MATHWEB environment.

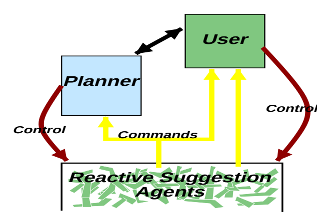


Some specialised Planning Agents

- **User-Int.-Agent** (OMEGA interactive): Interactive users may operate as a planning agents themselves
- **TPS-Agent** (TPS-System & TPS-Trans): The mating search based higher-order theorem prover TPS
- **LEO-Agent** (LEO (OMEGA)): The extensional higher-order resolution based theorem prover LEO
- **FO-Agent** (FO-Reasoners & FO-Trans): Various state of the art first-order theorem provers
- **Counter-Agent** (Model-Generator): An agent looking for counterexamples by employing a model-generator
- **Induction-Agent** (Induction Reas. & Ind.-Trans): An integrated theorem prover specialised in mathematical induction
- **CAS-Agent** (Specialised CAS-Simplif.): A problem simplifier based on an integrated computer algebra system
- **OMEGA-Agent** (Plan (OMEGA) & CAS & COSIE & FO-Systems): OMEGA's own traditional proof planner encapsulated as a planning agent
- **Analogy-Agent** (Analogy (OMEGA)): An analogy reasoner looking for similar, already proven lemmata in MBASE

Reactive Basis

The O-ANTS system was originally developed to support the user in interactive theorem proving. Recent experiments have shown that this agent-based, hierarchically organised system can also fruitfully support the resource guided integration of external reasoning components. Hence O-ANTS provides the architectural and implementational basis for the Multi-Agent Proof Planning approach.



Agent-based System Integration

$$\frac{P1 \; P2 \; \ldots \; Pn}{C} \text{ Call-TPS}$$

Inference Rule

```
(agent~defagent Call-TPS c-predicate
   (for C)
   (uses )
   (definition
      (tackle-by-TPS "P1 & P2 & ... & Pn --> C"))
   (complexity-level 100))
```

Declarative Agent Specification in O-ANTS

Advantages of the O-ANTS Architecture

- Supports interaction & automation
- Agents can be defined at run-time
- Agents can be modified at run-time
- Agents can be dynamically activated & deactivated at run-time
- Supports resource-adaptive guidance
- Very robust mechanism; faulty agent specifications do not harm the functioning of the overall mechanism

Implementation

The implementation employs the multiprocessing facilities of Allegro Common Lisp.

Currently we test the mechanism with up to 400 little software agents performing trivial computations.

Supported by EPSRC Grants GR/M99644 (Agent-Oriented Theorem Proving) & GR/M22031 (Proof Planning in Mathematics)