

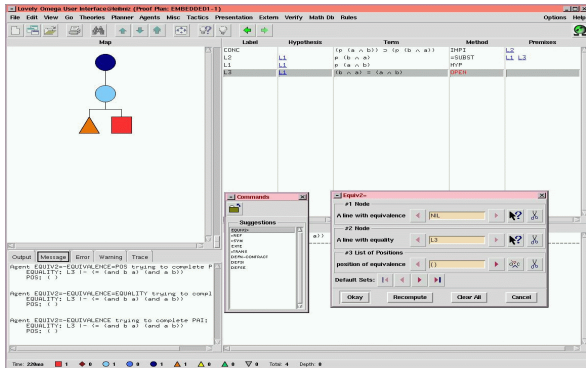


Fine-Grained Proof Planning with Critical Agents



Christoph Benz Müller & Volker Sorge, OMEGA-Group, Universität des Saarlandes, Saarbrücken

Suggesting Commands



- > suggestions for commands and arguments
- > works steadily in the background
- > flexible argument computations and (re-)computation wrt. user input
- > easy specification of agents
- > runtime compilation

Architecture

Planning with Agents

- > checking the applicability of methods with agents
 - > retrieve possible conclusions and premises
 - > matching
 - > wrt. application conditions
- > (concurrently) verify application conditions
- > enriched constraint language to specify concurrency
- > implement heuristics and control rules with selection criteria

- > one society of command agents
- > societies of argument agents
- > partial argument instantiations accumulated and broadcasted via blackboards

Advantages

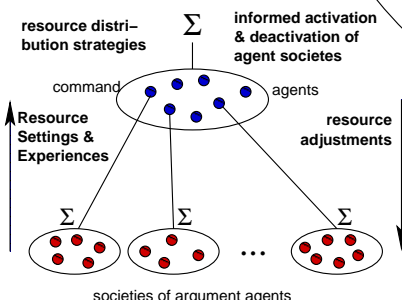
- use of concurrency
- heuristically sorted suggestions



Guidance

Resources

- > memory usage
- > computation time
- > domain specific knowledge



forced adjustment & self-adjustment

Planning

- Strips-like planning
- forw./back. chaining
- control by:
 - goal agenda
 - control rules

- planning on partial proof data structure

- proof methods specify
 - > premise and conclusion lines
 - > application conditions & constraints
 - > proof schema

Foci: two layered mechanism for structuring proofs

Method : Diag	
Conclusions	$(\ominus TH)$
Premises	$P1 (\oplus Ps)$
Application Conditions	$[match(formula(IR), \Phi) \& \{P \leftarrow P1\}] \mid$ $[assert(\Phi, formula(IR), Ps)$ $\& \{P \leftarrow IR\}$ $\& \{J \leftarrow ass_just(P1, Ps)\}]$
Ordering Constraint	$before(Ps, TH)$
Proof Schema	$P1 \vdash \Phi$ $IR \vdash \forall x_{\alpha} \rightarrow \beta E(x) \rightarrow \exists y_{\alpha} N(y) \wedge x = F(y) \quad (J)$ $TH \vdash \perp \quad (Diag\text{-}by\ P)$

