Ω-ANTS– An open approach at combining Interactive and Automated Theorem Proving

Christoph Benzmüller[1] and Volker Sorge[2]

[1]School of Computer Science, The University of Birmingham

[2]Fachbereich Informatik, Universität des Saarlandes,

Calculemus Symposium, St. Andrews, 6./7. August 2000

Overview

Ω-ANTS

Reactive User Suggestion Mechanism

Automated Theorem Prover (normal form ND)

Flexible Integration of External Systems

Formal Description & Analysis of the System

Ω-ANTS

as basis for Agent-based Proof Planning

Original Idea of Ω-ANTS

Shortcoming of traditional Suggestion Mechanisms (HOL, TPS, VSE, . . .)

- resource-wasting in-between user interactions

Instead

- computations in the background in-between interactions
- also support expensive and potentially non-terminating computations
- dynamically update list of user suggestions (commands)
- more time . . . better suggestions

Solution proposed in

- [AIMSA'98] Ω-ANTS Architecture, Focus Mechanism
- [EPIA'99] Ω-ANTS Resource Concept and Interaction Facilities

Partial Argument Instantiations (PAI)

- Rules, Tactics, Methods, External Reasoners, etc.

$$\frac{\forall x_{\bullet} A}{[t/x]A}\ \forall_E(t) \qquad \frac{\forall x_1,\ldots,x_{n\bullet} A}{[t_1/x_1,\ldots,t_n/x_n]A}\ \forall_E^*(t_1,\ldots,t_n) \qquad \frac{A^1 \ldots A^n}{C}\ Otter$$

- ... are invoked by associated Commands

$$\frac{A \quad B}{A \wedge B}\ \wedge_I \ \longrightarrow\ \frac{\text{LConj} \quad \text{RConj}}{\text{Conj}}\ \text{AndI}$$

- PAI's: ... a way to communicate command + argument suggestions

AndI(Conj: L5) $\rightarrow$ backward application of AndI to L5

AndI(RConj: L2, LConj: L1) $\rightarrow$ forward application of AndI to L1 and L2

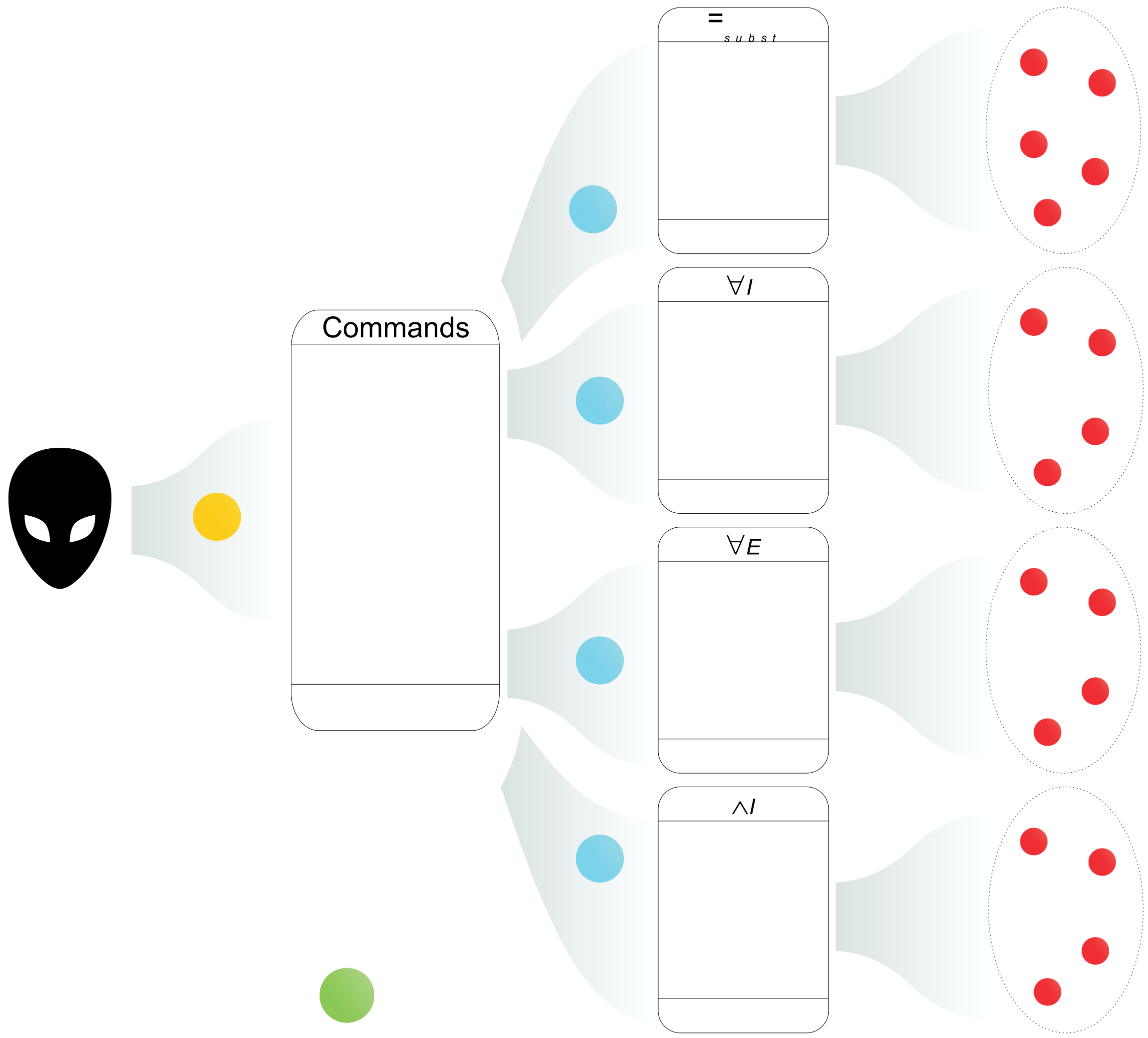
PAI's as functions (substitutions)

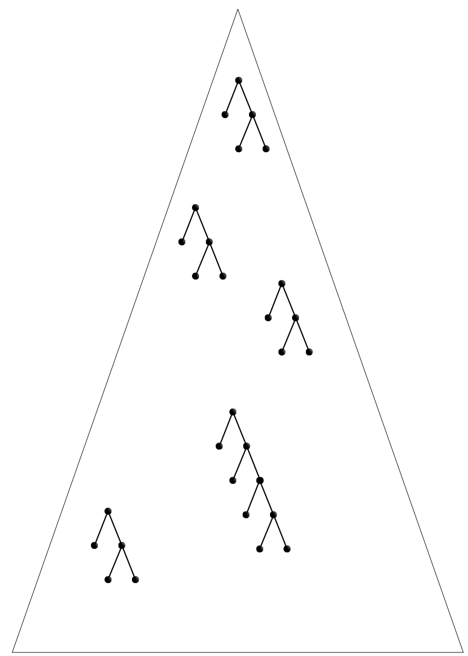
$$PAI^{AndI} : \underbrace{\{\text{LConj, RConj, Conj}\}}_{\text{AndI-Parameternames}} \longrightarrow \underbrace{\{\text{L1}, \ldots, \text{Ln}, \ldots\}}_{\text{Prooflines}} \cup \{\epsilon\}$$

E.g.:

AndI(RConj: L2, LConj: L1) represents a respective function with

$$PAI^{AndI}(\text{LConj}) \equiv \text{L1}$$
$$PAI^{AndI}(\text{RConj}) \equiv \text{L2}$$
$$PAI^{AndI}(\text{Conj}) \equiv \epsilon$$


Commands
=
subst
∀I
∀E
∧I



Components

Blackboard's

They accumulate PAI's. The Suggestion Blackboard contains the (heuristically) best rated applicable PAI's (for all rules) wrt. given partial proof.

Command Agents & Suggestion Agent

They heuristically select and report PAI's from one layer to next one in the hierarchical system.

Argument Agents

They pickup PAI's, employ the represented information, search through the partial proof according to their specification, and report a new (extended) PAI.

Argument Agents

- Remember: $\frac{A \quad B}{A \wedge B} \wedge_I \longrightarrow \frac{\text{LConj} \quad \text{RConj}}{\text{Conj}}$ AndI

- Specification of Argument Agents for AndI

1	$\mathfrak{C}^{\{\text{Conj}\}}_{\{\},\{\text{LConj},\text{RConj}\}}$	=	"Is-Conjunction Conj "
2	$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{LConj}\},\{\text{RConj}\}}$	=	"(Is-Conjunction Conj) & (Is-Left-Conjunct LConj Conj)"
3	$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{RConj}\},\{\text{LConj}\}}$	=	"(Is-Conjunction Conj) & (Is-Right-Conjunct RConj Conj)"
4	$\mathfrak{S}^{\{\text{RConj}\}}_{\{\text{Conj}\},\{\}}$	=	"(Is-Right-Conjunct RConj Conj)"
5	$\mathfrak{S}^{\{\text{LConj}\}}_{\{\text{Conj}\},\{\}}$	=	"(Is-Left-Conjunct LConj Conj)"
6	$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{LConj},\text{RConj}\},\{\}}$	=	"(Is-Conjunction Conj) & . . . "

Argument Agents (more formal) . . .

$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{LConj}\},\{\text{RConj}\}}$:= "(Is-Conjunction Conj) & (Is-Left-Conjunct LConj Conj)"

can be represented as the predicate

$$\lambda\, Conj_{\text{Open}} \,\blacksquare\, (Conj \equiv \text{A} \wedge \text{B}) \;\&\; (\text{A} \equiv \text{LConj})$$

or equivalently ↘ ↙

$$\lambda\, Conj_{\text{Open}} \,\blacksquare\, (Conj \equiv \text{LConj} \wedge \text{B})$$

Note: The aspect that these agents pickup PAI's from the blackboard and return potentially extended PAI's is not represented here.

Argument Agents (more formal)

1 $$\mathfrak{C}^{\{\text{Conj}\}}_{\{\},\{\text{LConj},\text{RConj}\}} := \lambda\, Conj_{\text{Open}} \,.\, (Conj \equiv \text{A} \wedge \text{B})$$

2 $$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{LConj}\},\{\text{RConj}\}} := \lambda\, Conj_{\text{Open}} \,.\, (Conj \equiv \text{A} \wedge \text{B}) \,\&\, (\text{LConj} \equiv \text{A})$$

3 $$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{RConj}\},\{\text{LConj}\}} := \lambda\, Conj_{\text{Open}} \,.\, (Conj \equiv \text{A} \wedge \text{B}) \,\&\, (\text{RConj} \equiv \text{B})$$

4 $$\mathfrak{S}^{\{\text{RConj}\}}_{\{\text{Conj}\},\{\}} := \lambda\, RConj_{\text{Premise}} \,.\, (\text{Conj} \equiv \text{A} \wedge \text{B}) \,\&\, (RConj \equiv \text{B})$$

5 $$\mathfrak{S}^{\{\text{LConj}\}}_{\{\text{Conj}\},\{\}} := \lambda\, LConj_{\text{Premise}} \,.\, (\text{Conj} \equiv \text{A} \wedge \text{B}) \,\&\, (LConj \equiv \text{A})$$

6 $$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{LConj},\text{RConj}\},\{\}} := \lambda\, Conj_{\text{Open}} \,.\, (Conj \equiv \text{A} \wedge \text{B}) \,\&\, (LConj \equiv \text{A}) \,\&\, (\text{RConj} \equiv \text{B})$$

Argument Agents (even more formal)

$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{LConj}\},\{\text{RConj}\}}$:= "(Is-Conjunction Conj) & (Is-Left-Conjunct LConj Conj)"

can be represented as a function that picks up PAI's for AndI and returns potentially extended PAI's thereby employing an encapsulated search predicate as described before:

$\lambda\ PAI\,.$

$\quad \lambda\ Conj_{\textit{Open}}\,.$

$\quad\quad$ if $PAI(\text{Conj}) \equiv \epsilon \;\&\; PAI(\text{LConj}) \not\equiv \epsilon \;\&\; PAI(\text{RConj}) \equiv \epsilon$

$\quad\quad$ then if $Conj \equiv PAI(\text{LConj})\ \wedge \text{B}$

$\quad\quad\quad$ then $PAI|_{\{\text{LCONJ, RCONJ}\}} \cup \{\text{Conj} \mapsto Conj\}$ $\quad \to$ new ext. PAI

$\quad\quad\quad$ else PAI $\quad \to$ no new PAI

$\quad\quad\quad$ fi

$\quad\quad$ else PAI $\quad \to$ no new PAI

$\quad\quad$ fi

Declarative Specification in ΩMEGA

$\mathfrak{C}^{\{\text{Conj}\}}_{\{\text{LConj}\},\{\text{RConj}\}} :=$

```
(agent~defagent AndI c-predicate
   (for Conj) (uses LConj)
   (definition
     (and (logic~conjunction-p Conj)
          (logic~left-conjunct-p LConj Conj)))
   (information :plω) (level 1))
```

⟶ Run-time definability and modifiability of all agents

Integration of External Systems

$$\frac{P_1 \quad \ldots \quad P_n}{C}\ \textit{Otter}\ \textit{Mace} \qquad \frac{A \quad B \Rightarrow C}{C}\ \begin{array}{l}\textit{mp-mod-CAS}(A \stackrel{\text{simpl}}{\rightarrow} B)\\ \textit{mp-mod-Otter}(A \Rightarrow B)\end{array}$$

$$\frac{\texttt{Prem}_1 \quad \ldots \quad \texttt{Prem}_n}{\texttt{Conc}}\ \texttt{Otter}\ \texttt{Mace} \qquad \frac{\texttt{Left} \quad \texttt{Impl}}{\texttt{Conc}}\ \begin{array}{l}\texttt{mp-mod-CAS}(\texttt{Simpl-Prob})\\ \texttt{mp-mod-Otter}(\texttt{Impl-Prob})\end{array}$$

Integration of External Systems

$$\frac{A \quad B \Rightarrow C}{C}\ \textit{mp-mod-Otter}(A \Rightarrow B)$$

$$\frac{\texttt{Left} \quad \texttt{Impl}}{\texttt{Conc}}\ \texttt{mp-mod-Otter}(\texttt{Impl-Prob})$$

$\mathfrak{C}^{\{\text{Left}\}}_{\{\text{Impl}\},\{\}} :=$

$\lambda\ PAI$.

$\lambda\ Left_{\textit{Premise}}$.

if $PAI(\text{Left}) \equiv \epsilon$ & $PAI(\text{Impl}) \not\equiv \epsilon$

then if provable-by-OTTER($Left \Rightarrow$ left-conjunct-of($PAI(\text{Impl})$))

then $PAI|_{\{\text{Impl, Conc}\}} \cup \{\text{Left} \rightarrow Left\}$ $\rightarrow$ new extended PAI

else PAI $\rightarrow$ no new PAI

fi

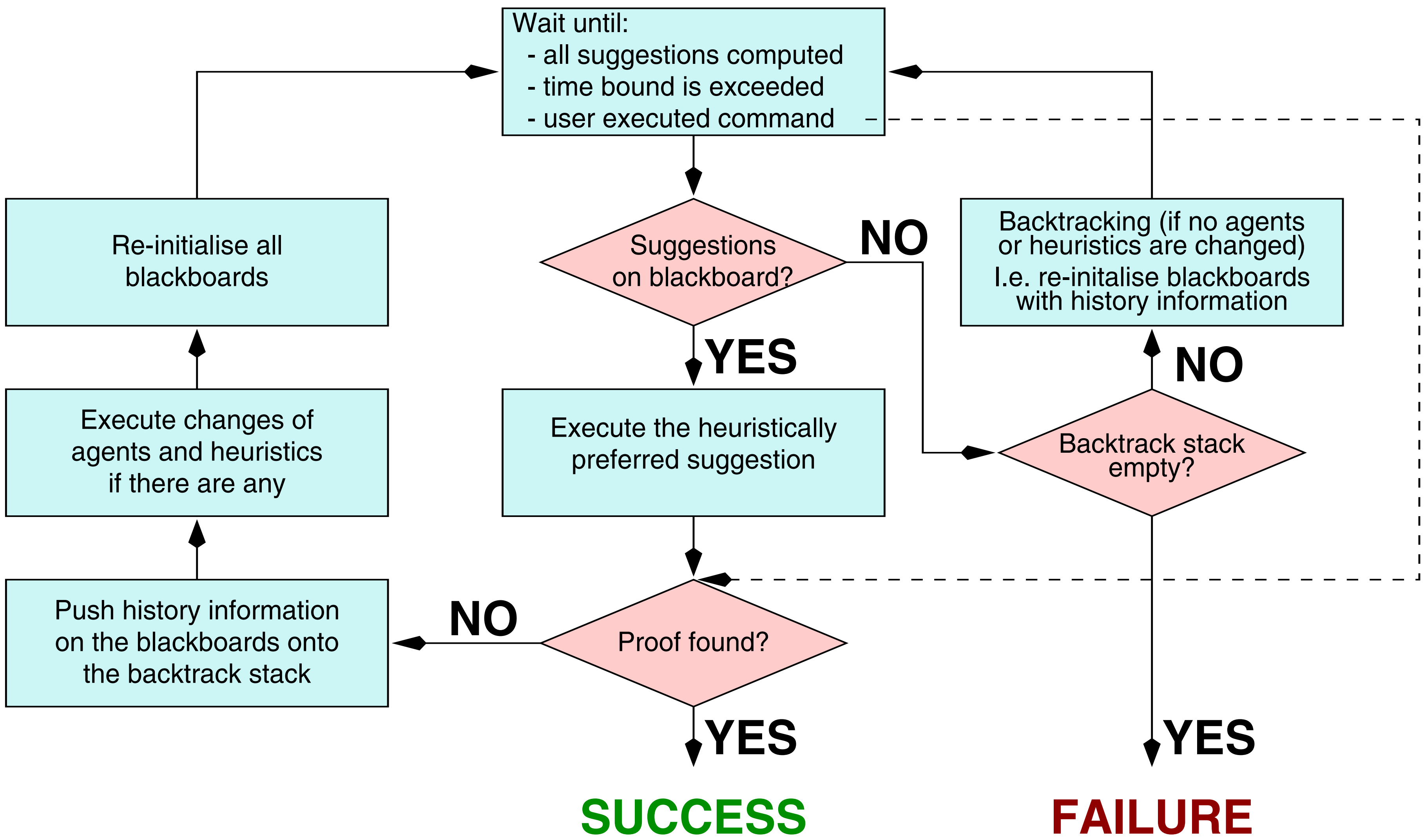
else PAI $\rightarrow$ no new PAI

fi

$\mathfrak{C}^{\{\text{Conc}\}}_{\{\},\{\text{Left},\text{Impl}\}} := \ldots$

$\mathfrak{C}^{\{\text{Impl}\}}_{\{\text{Conc}\},\{\}} := \ldots$

Automation


Wait until:
- all suggestions computed
- time bound is exceeded
- user executed command
Re-initialise all blackboards
Suggestions on blackboard?
NO
Backtracking (if no agents or heuristics are changed)
I.e. re-initalise blackboards with history information
YES
NO
Execute changes of agents and heuristics if there are any
Execute the heuristically preferred suggestion
Backtrack stack empty?
Push history information on the blackboards onto the backtrack stack
NO
Proof found?
YES
YES
SUCCESS
FAILURE

Completeness and Soundness ?

Given: A theoretically complete/sound calculus

Question: How can it be appropriately modelled in Ω-ANTS?

Completeness:

$\swarrow \quad \searrow$

Automation Completeness Interaction Completeness

Soundness: hfill Interest in potentially non-sound rules (proof methods)

$\rightarrow$ Applicability rather than Soundness

Automation Completeness

To investigate:

1. Are the individual parameter agent specifications adequate, wrt. to their intended tasks ?
2. Are there sufficiently many parameter agents specified to realise a fruitful interplay ?
3. Are the suggestion and command agents non-excluding, i.e. do they always report applicable entries ?
4. Does the sketched main search loop guarantee a fair search ?

Automation Completeness

Context: NIC calculus [Byrnes99], normal from ND with pure backward search

- Agents $\boxed{1}$ — $\boxed{6}$ are adequate
- Agents $\boxed{1}$, $\boxed{4}$, $\boxed{5}$ are sufficient to compute all (backward) PAI's

AndI()

$\downarrow \boxed{1}$

AndI(Conj:$A \wedge B$)

$\boxed{4} \swarrow \quad \searrow \boxed{5}$

AndI(Conj:$A \wedge B$,LConj:A)   AndI(Conj:$A \wedge B$,RConj:B)

$\boxed{5} \searrow \quad \swarrow \boxed{4}$

AndI(Conj:$A \wedge B$, LConj:A, RConj:B)

Interaction Completeness

Idea: Ensure that the user never has to fall back on another interaction mechanism, i.e. any applicable PAI can be computed.
(Assumption: The user does not accept the typically restricted rule application directions in an automated calculus.)

Examples:

- Now agents [1], [4], [5] are not sufficient anymore
- A user query PAI AndI(LConj:Ln) can be extended by agent [2] to AndI(LConj:Ln,Conj:Lm)
- A forward application of AndI is not supported by agents [1] — [6], i.e. no PAI AndI(LConj:Ln,RConj:Lm) can be computed

$\rightarrow$ no interaction completeness

Conclusion

- Ω-ANTS architecture supports interaction & automation
- Inheritance of Ω-ANTS main properties: resource adaptability, run-time extendibility, flexibility, robustness, . . .
- Integration of external systems even at a very fine grained layer
- No difference (from a practical view) between computation and search
- Formal semantics and completeness proofs? $\rightarrow$ Future
- Play with reasoning in main calculus (depth) $\leftrightarrow$ reasoning in external systems (breadth) $\rightarrow$ Future

Related Work:

HOL, PVS, TPS . . .

Proof Planning, ΩMEGA

OMRS