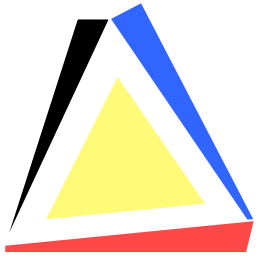


Agent-based Proof Search with Indexed Formulas

Malte Hübner, Serge Autexier, and Christoph Benzmüller

with thanks to Volker Sorge

Saarland University, Saarbrücken, Germany

Calculemus 2002, Marseille, July 3rd

What is the talk about?

- CORE: A new basic inference system for ΩMEGA that is based on a (dynamic) rewriting approach
  (developed by Serge Autexier)
- Adapting the suggestion mechanism Ω-ANTS to support proof search in CORE

This is work in progress

CORE: A new basis for ΩMEGA

- Rewriting and focus placement as basic constructs
- Set of rewrite rules dynamically created for each focus
- Overall goal: rewrite problem formalization to $\top$ or $\bot$
- Strong support by simplification; e.g., $\bot \Rightarrow a \vee b$ simplifies to $\top$
- Our motivation: better suited for practical reasoning than conventional calculi
  - treats (propositional) logical aspects implicit; e.g, unwrapping of hypothesis
  - no constraints on the order of quantifier elimination
- System recently applied to interactive proof of $\sqrt{2}$

How does it work ?

Example: $\neg(\neg A \vee \neg B) \wedge (A \Rightarrow (B \Rightarrow C)) \Rightarrow C \vee D$

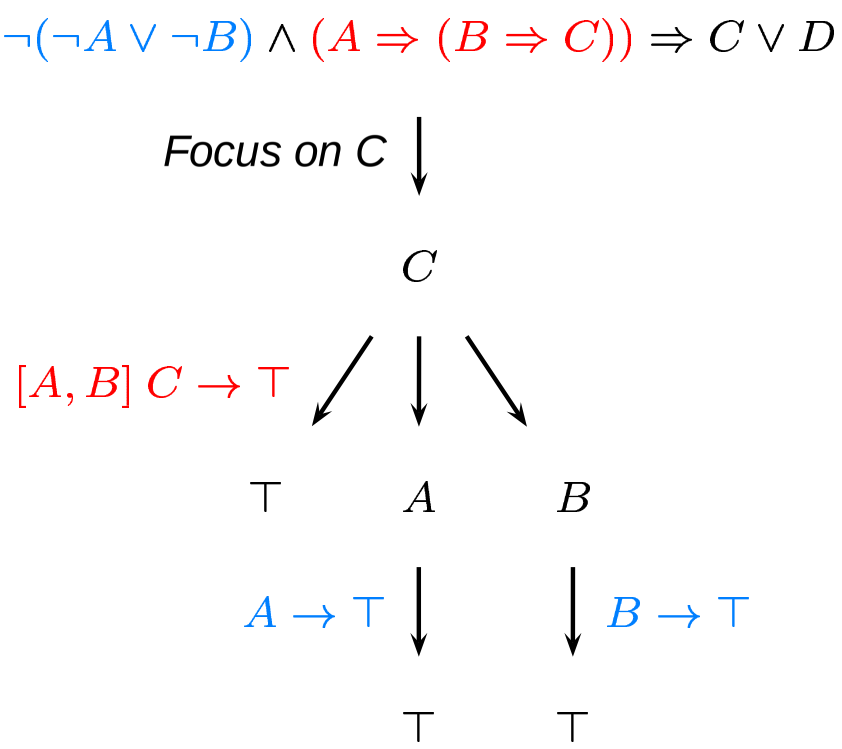
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{C} \vdash \mathbf{C}}}{C \vdash \neg A, \neg B, D, C} \quad \dfrac{\dfrac{\dfrac{}{\mathbf{B} \vdash \mathbf{B}}}{B \vdash \neg A, B, C, D}}{\vdash \neg A, \neg B, C, D, B}}{B \Rightarrow C \vdash \neg A, \neg B, C, D} \quad \dfrac{\dfrac{}{\mathbf{A} \vdash \mathbf{A}}}{A \vdash A, B, \neg B, D, C}}{\vdash A, \neg A, \neg B, C, D}}{A \Rightarrow (B \Rightarrow C) \vdash C, D, (\neg A \vee \neg B)}}{\neg(\neg A \vee \neg B), (A \Rightarrow (B \Rightarrow C)) \vdash C, D}}{\neg(\neg A \vee \neg B) \wedge (A \Rightarrow (B \Rightarrow C)) \vdash C, D}}{\neg(\neg A \vee \neg B) \wedge (A \Rightarrow (B \Rightarrow C)) \vdash C \vee D}}{\vdash \neg(\neg A \vee \neg B) \wedge (A \Rightarrow (B \Rightarrow C)) \Rightarrow C \vee D}$$

Proof of $A \wedge B \wedge (\neg A \vee (B \Rightarrow C)) \Rightarrow C \vee D$ is different!

Malte Hübner, Serge Autexier, and Christoph Benzmüller

How does it work ?

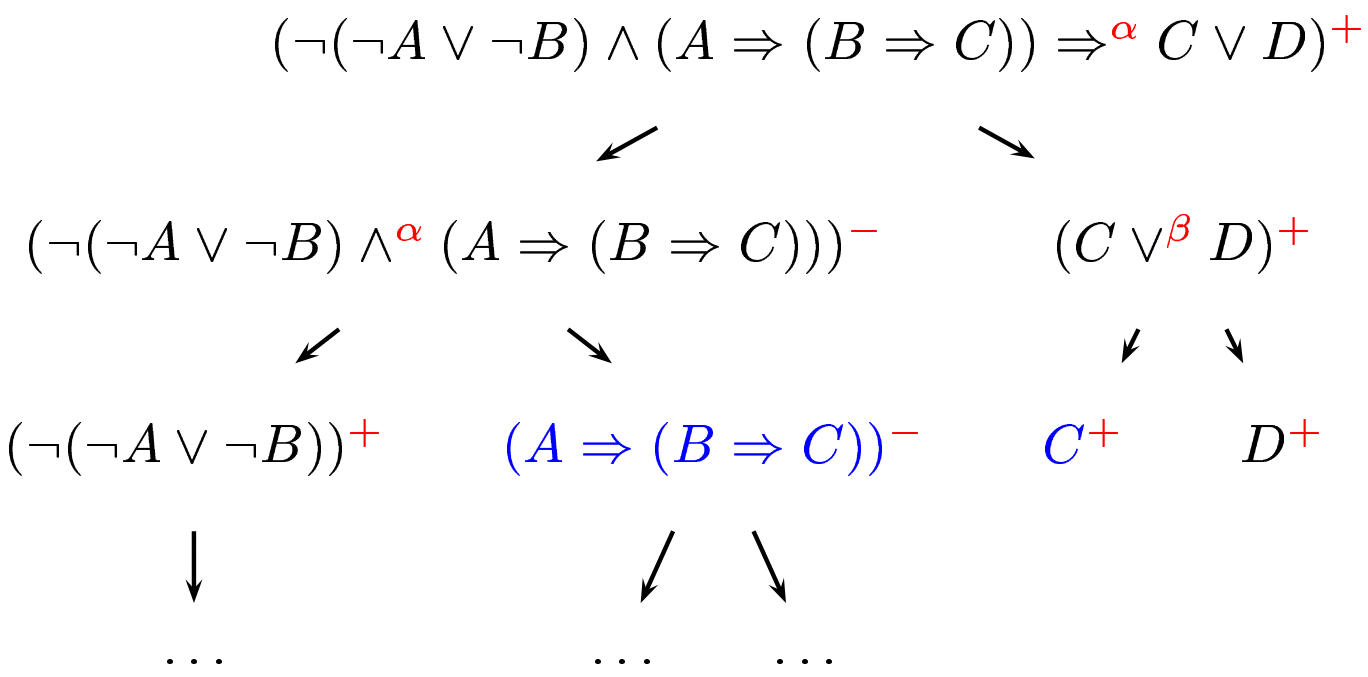
Internal representation: Indexed formula tree



Simplification: $\neg(\neg A \vee \neg B) \wedge (A \Rightarrow (B \Rightarrow C)) \Rightarrow \top \vee D$ simplifies to $\top$

Note: Identical proof for $A \wedge B \wedge (\neg A \vee (B \Rightarrow C)) \Rightarrow C \vee D$.

Indexed formula tree



α-related subformulas belong to the same context r rewriting

Properties of CORE

- New system well suited for interaction with humans (but take care with focus)
- Search Space:
  - Challenge: **choice of focus**
  - Rules of the form $[\Phi]\ u \to v_1 \ldots v_n$ are generated **dynamically**
  - Focus choice and rule selection are interdependent
  - Backtracking
- Advantages for proof search:
  - **big steps** rather than reasoning about logical details
  - **more information is available** in each proof state

Proof Search

To support interactive and automated proof search we want a mechanism that:

- makes suggestions on how to place focus
- computes instances of applicable rewrite rules
- computes instances of additional tactics and methods
- filters and heuristically orders the instances of applicable rules

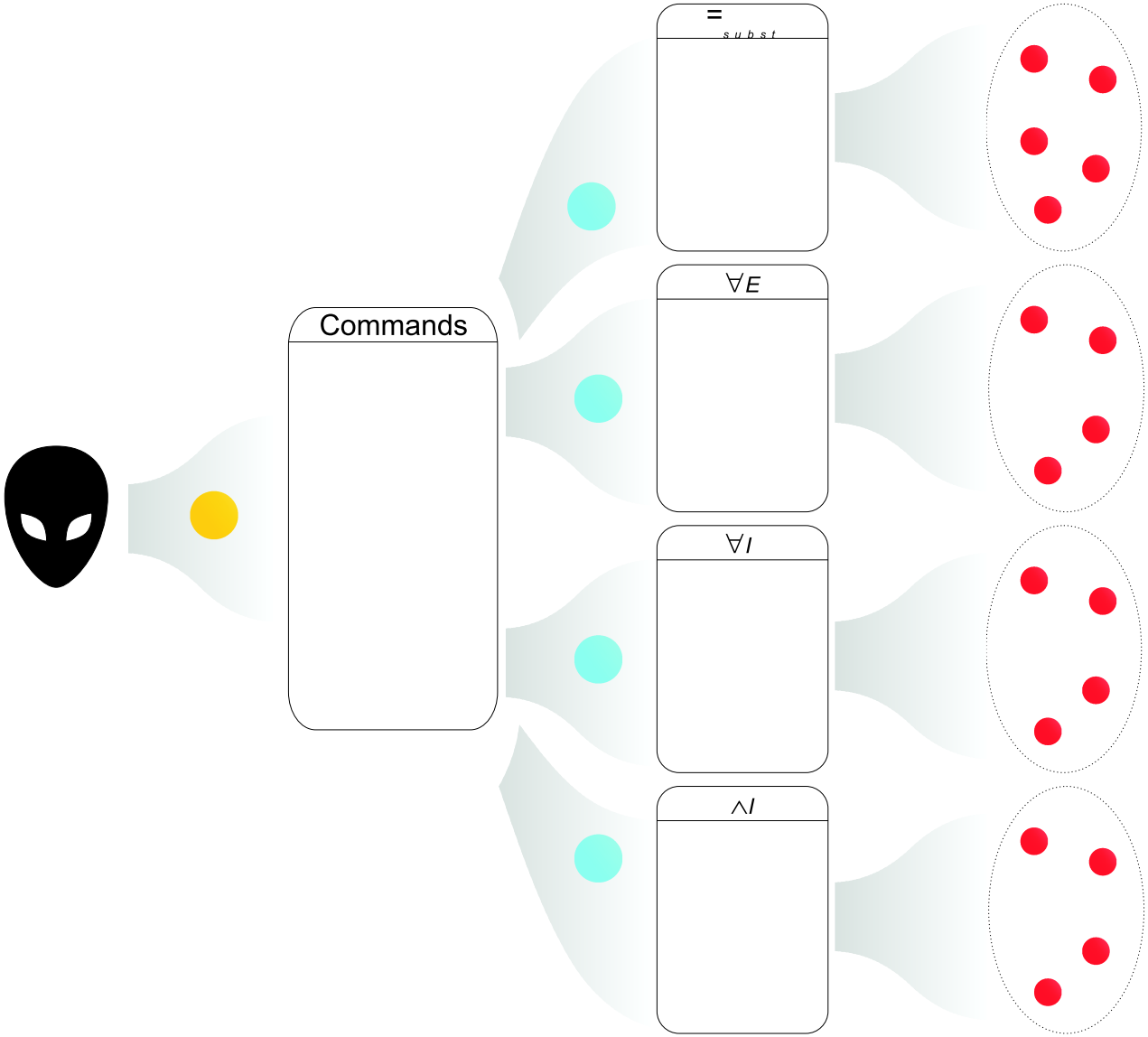
$\Rightarrow$Adapt and employ sugggestion mechanism Ω-ANTS

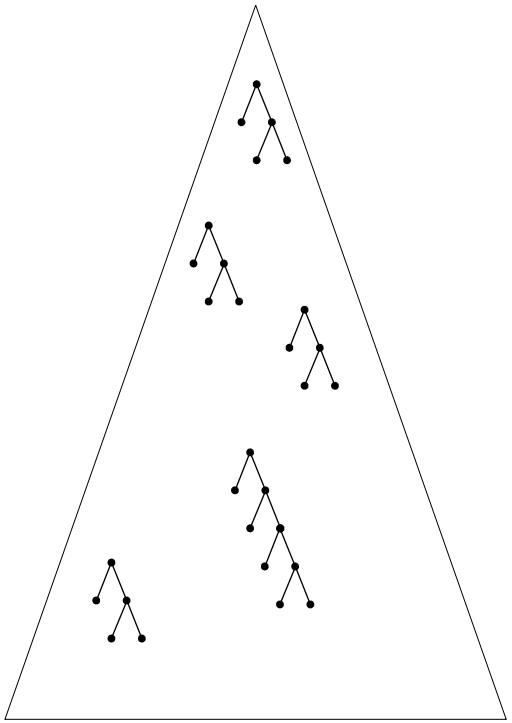
Ω-ANTS

- Key idea:

$$\frac{p_1 \ldots p_m}{c} \; name(t_1, \ldots, t_n)$$

  - Distributed search for instantiations of $p_1, \ldots, p_m, c$ in partial proof; distributed computation of instantiations for $t_1, \ldots, t_m$
  - Communication (about dependencies between parameters) via blackboards
- Certain patterns of instantiations are considered to be applicable
- Heuristically order applicable rules





Malte Hübner, Serge Autexier, and Christoph Benzmüller

Ressourcenadaptive Kognitive Prozesse

Approach for Rewrite Rules

- Rewriting with dynamically generated rewrite rules instead of ND
- Needed: suggestion of applicable rewrite rule instances for current focus
- How to search for applicable instances in formula tree?
  - Model all (dynamic) rewrite rules as agents in Ω-ANTS:
    - dynamic creation of agents for each rule
    + distribution of search for instances
  - Model one generic rewrite agent that handles all rules sequentially:
    + no dynamic creation of agents
    - no distribution of complex instantiation criteria (HO-unification)
- Proposal is to split work
  - one agent that employs FO-unification
  - one or several working with HO-unification

Approach for Tactics/Methods

Static tactics/methods still specified and modeled as before:

$$\frac{p_1, p_2, p_3, p_4, p_5 \ldots, p_n}{c}\ name(t_1, \ldots, t_m)$$

Applicable *partial argument instantiations* have to be cast into rewrite rules for application in the system

- Use Ω-ANTS as it is
- Agents have to search for instantiations in indexed formula tree
- Transform each applicable instantiation pattern in rewrite rule:
  If c, p_1, p_3 is the set of parameters for which instantiations were found

$$c \rightarrow p_2 \wedge p_4 \wedge \ldots \wedge p_n$$

Approach for Tactics/Methods

$$\frac{p_1 : A \Rightarrow B \quad p_2 : B \Rightarrow C}{c : A \Rightarrow C}\ ModusBarbara$$

Example: $(P(a) \Rightarrow Q(a)) \wedge A(f(g)) \wedge R(y) \Rightarrow (P(a) \Rightarrow R(a))$

$$\frac{p_1 : P(a) \Rightarrow Q(a) \quad p_2 : \emptyset}{c : P(a) \Rightarrow R(a)}\ ModusBarbara$$

Focusing on $P(a) \Rightarrow R(a)$ should yield the following rewrite rules:

$$(P(a) \Rightarrow R(a)) \rightarrow (Q(a) \Rightarrow R(a))$$

$$[Q(a) \Rightarrow R(a)] \ \ P(a) \rightarrow R(a)$$

Approach for Focus Placement

Focus Agent:

- Suggests to consider certain foci and excludes other foci from the search
  - Criteria from theoretical investigations for heuristics/strategies
  - Criteria of human mathematicians
- Problem: focus placement can easily become to complicate for humans
  - user model needed for optimal support ?

Winding up

- Adaptation of Ω-ANTS to support interactive proof search in CORE
  - Difficulty: interplay between focus placement, dynamic generated rewrite rules, tactics/methods, and backtracking
- Current focus of work: Interactive theorem proving
  Possible evaluation:
  Get a novice user (e.g. maths student) to replay his blackboard proof of a mathematical problem such as *irrationnality of* $\sqrt{2}$ in the system as adequately and as comfortably as possible
- Future work: Automation of proof search based on Ω-ANTS and mixed-initiative proof search