Tutorial Dialog on Mathematical Proofs

Christoph Benzmüller

The DIALOG Group

chris@ags.uni-sb.de

http://www.ags.uni-sb.de/~chris

Universität des Saarlandes, Saarbrücken, Germany

Motivation

Computational Linguistics

- natural language in mathematics
- need for deep semantical processing (in combination with shallow processing)
- develop missing corpus

Deduction Systems / Maths Assistant Systems

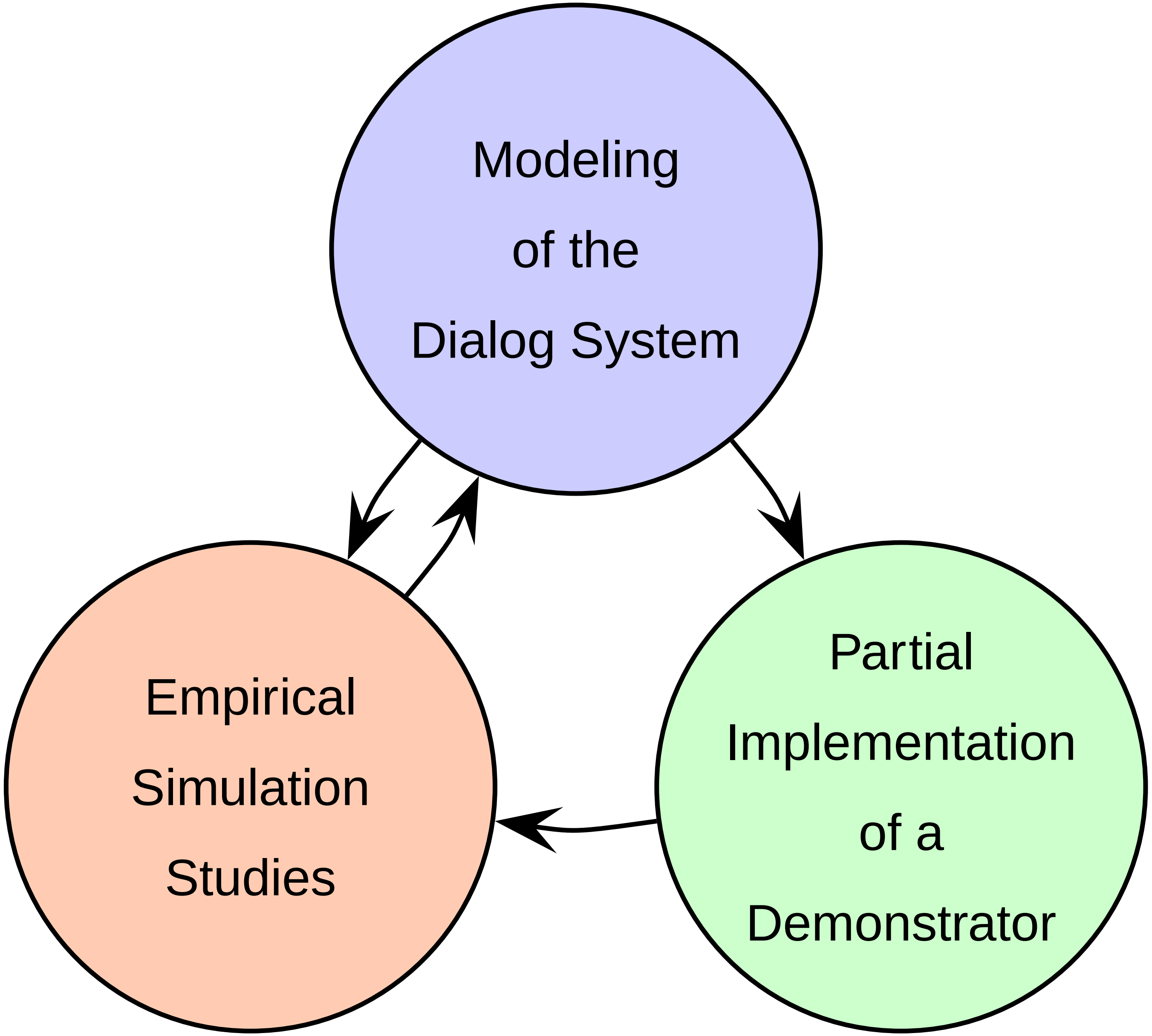
- prospectous application of deduction systems
- interesting system integration aspects
- novel requirement for theorem proving:
  quality of proofs (proof plans)

Intelligent Tutoring Systems

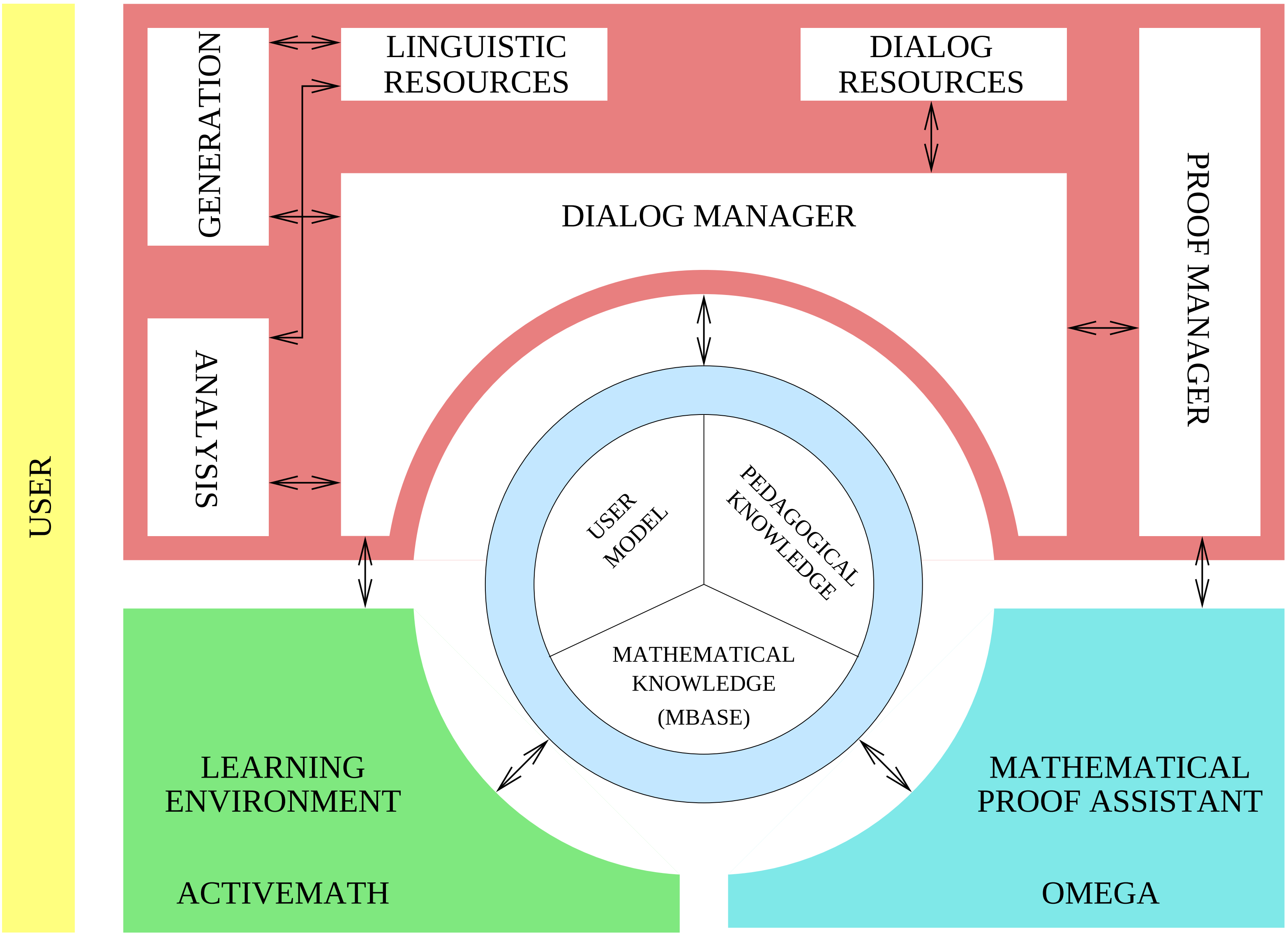
Active learning through solving problems in a specific domain

- Domain Modeling
  - static vs. dynamic generation of solutions
- User Input
  - menu-based vs. unrestricted user input
  - meaning-insensitive vs. meaning-conscious analysis
  - combine shallow and deep processing
- System Output
  - canned text, templates or full-fledged generation

Method: Progressive Refinement



Architecture



Domain: Naive Set Theory

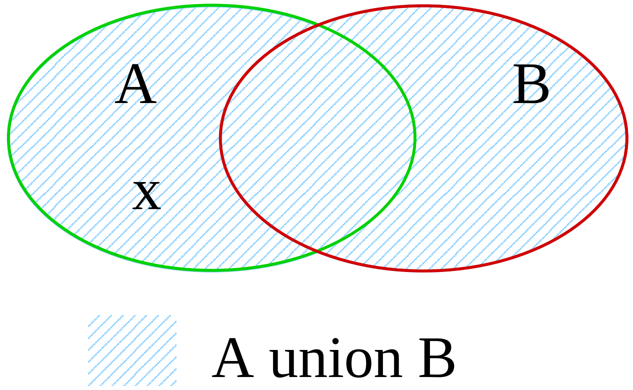
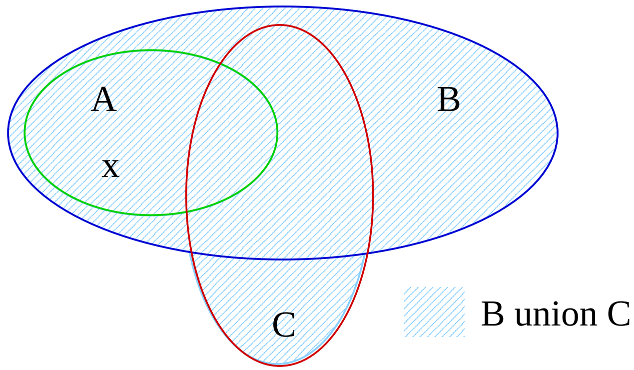
Concepts

- Subset: $U \subseteq V$ iff for all $x \in U$ holds $x \in V$
- Superset: $U \supseteq V$ iff for all $x \in V$ holds $x \in U$
- ...

Relations

- Antithesis: $\in$ is in antithesis to $\notin$
- Duality: $\subseteq$ is dual to $\supseteq$
- Hypotaxis: $\in$ is hypotactical to $\subseteq$
- ...

Domain: Naive Set Theory

Declarative View	Procedural View	Diagrammatic View
$\forall x, A, B. x \in A \Rightarrow (x \in A \cup B)$	$\dfrac{x \in A}{x \in A \cup B}$	
$\forall A, B, C. A \subseteq B \Rightarrow (A \subseteq B \cup C)$	$\dfrac{A \subseteq B}{A \subseteq B \cup C}$	
$\forall A, B. (A \subseteq B) \Rightarrow (A \in P(B))$	$\dfrac{A \subseteq B}{A \in P(B)}$	?

Mathematics Tutorial Dialog

- student answer categorization
  - preliminary taxonomy of student answers
- taxonomy of hints
  - elaborate hierarchy of hint categories
- socratic tutorial strategy
  - elicit problem solving through hinting
- marking of given information

Wizard-of-Oz Experiment: Goals

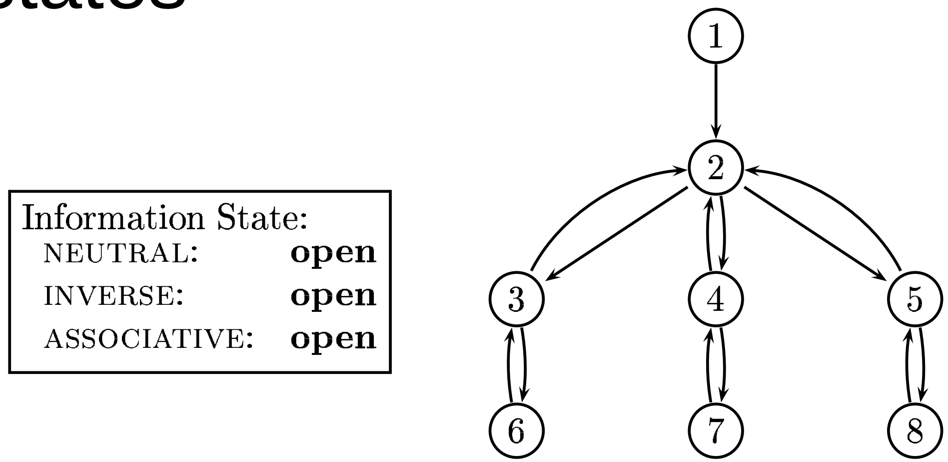
- collect data
  - tutoring process
  - student answers
  - dialog behavior
  - use of natural language
- test hinting algorithm
- test underlying concepts

Wizard-of-Oz Experiment: Set Up

- 24 Subjects:
  - university students
  - varying background
  - varying math knowledge
- Wizard:
  - mathematician with tutoring experience
  - assisted by developers of hinting algorithm
- Experimenter

DiaWoZ

- combination of finite-state automaton with information states



- dialog modeling on desired level of granularity
- flexible substitution of wizard functions by implemented modules

WOz Experiment: Phases

- Pre-Phase: background questionnaire, lesson material, test proof on paper
- Tutoring session: evaluate a tutoring system with NL dialog capabilities
  - familiarization proof:
    $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$
  - two “serious” proofs in varying order:
    $A \cap B \in P((A \cup C) \cap (B \cup C))$
    $Wenn\ A \subseteq K(B),\ dann\ B \subseteq K(A)$
- Post-Phase: test proof on paper, evaluation questionnaire

WOz Experiment: Wizard’s Task

- categorize student’s answer:
  - information completeness
  - accuracy (correctness and step size)
  - relevance
- decide next dialog move(s): hint selection restricted by algorithm
- verbalize dialog move(s): unconstrained

Tasks of NL Input Analysis

- formula parsing and type checking
  - $A \cap B \in P(A \cap B) \in P(A \cap B) \cup P(C)$
- parsing and interpreting interleaved NL and ML
  - A *muss in* B *sein*
  - B *enthaelt kein* $x \in A$
- recognizing patterns expressing proof steps
  - *[wenn* $A \subseteq K(B)$*,] [dann* $A \neq B$*,] [weil* $B \neq K(B)$*]*
  - *[falls* $A \subseteq B$ *und* $A \subseteq C$*] [dann gilt* $A \subseteq B \cap C$*]*

Tasks of NL Input Analysis

- reference resolution
  - co-reference
    Da, wenn $A \subseteq K(B)$ *sein soll,* A *Element von* $K(B)$ *sein muss. Und wenn* $B \subseteq K(A)$ *sein soll, muss es auch Element von* $K(A)$ *sein.*
  - discourse deixis:
    den oberen Ausdruck,
    aus der regel in der zweiten zeile
  - metonymy: *Dies fuer die innere Klammer.*

Tasks of NL Input Analysis

- resolving imprecise or informal naming of domain concepts and relations:

  A *enthaelt* B,

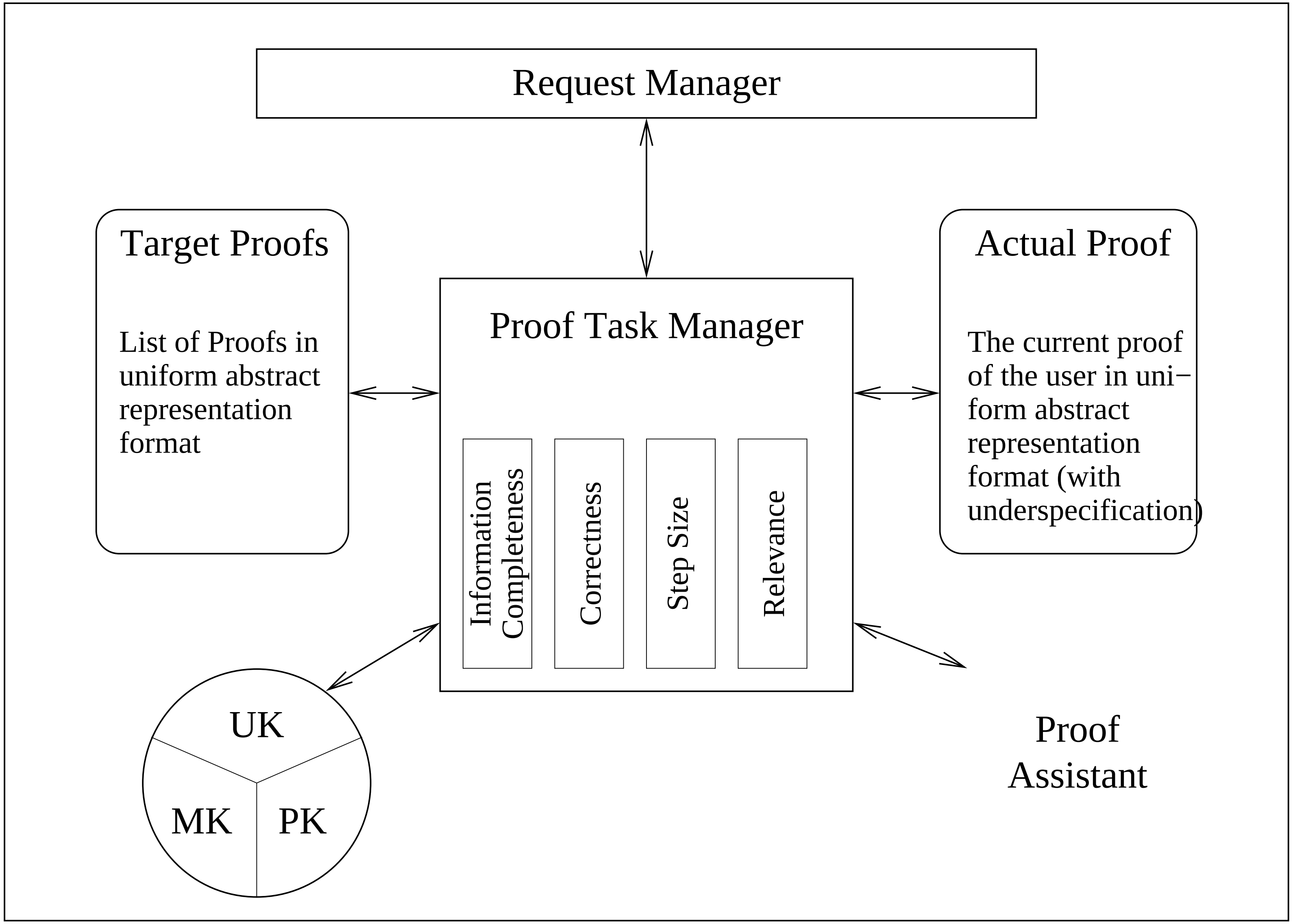
  . . . dann sind A *und* B *(vollkommen) verschieden, haben keine gemeinsamen Elemente*

- interpreting informal descriptions of proof-step actions

  aufloesen, *herausbekommen*, *ausrechnen*, *zerlegen*, *umstellen*

- interpreting ill-formed input

Tasks for the Proof Manager



Tasks for the Proof Manager

Example: didactic, vp16, dryrun

T1: Bitte zeigen Sie : K ((A ∪ B) ∩ (C ∪ D)) = (K (A) ∩ K (B)) ∪ (K (C) ∩ K (D))
S1: nach deMorgan-Regel-2 ist K ((A ∪ B) ∩ (C ∪ D)) = (K (A ∪ B) ∪ K (C ∪ D))
T2: Das ist richtig !
S2: K (A ∪ B) ist laut DeMorgan-1 K (A) ∩ K (B)
T3: Das stimmt auch .
S3: und K (C ∪ D) ist ebenfalls laut DeMorgan-1 K (C) ∩ K (D)
T4: Auch das stimmt .
S4: also folgt letztendlich : K ((A ∪ B) ∩ (C ∪ D)) = (K (A) ∩ K (B)) ∪ (K (C) ∩ K (D)) .
T5: Das stimmt genau . Ich wiederhole noch einmal : Auf die linke Seite der Gleichung kann ich zuerst die zweite und danach die erste de-Morgan-Regel anwenden , so daß sich folgende Argumentationskette ergibt : K ((A ∪ B) ∩ (C ∪ D)) = K (A ∪ B) ∪ K (C ∪ D) = (K (A) ∩ K (B)) ∪ (K (C) ∩ K (D)) .

Tasks for the Proof Manager

T1: Bitte zeigen Sie : K ((A ∪ B) ∩ (C ∪ D)) = (K (A) ∩ K (B)) ∪ (K (C) ∩ K (D))

$Task[DM1 : \forall X,Y.\overline{X \cup Y} = \overline{X} \cap \overline{Y},\ DM2 : \forall X,Y.\overline{X \cap Y} = \overline{X} \cup \overline{Y} \triangleright G : \overline{(A \cup B) \cap (C \cup D)} = (\overline{A} \cap \overline{B}) \cup (\overline{C} \cap \overline{D})]$

S1: nach deMorgan-Regel-2 ist K ((A ∪ B) ∩ (C ∪ D)) = (K (A ∪ B) ∪ K (C ∪ D))

$D : \overline{(A \cup B) \cap (C \cup D)} = (\overline{A \cup B}) \cup (\overline{C \cup D}) \quad \overset{forw}{by} \quad Appl(Ass(DM2), Pos(?), Subst(?))$

$Task[DM1 : \ldots, DM2 : \ldots \triangleright D : \ldots]$

$Task[DM1 : \ldots, DM2 : \ldots, D : \ldots \triangleright G : \ldots]$

T2: Das ist richtig !

$Task[DM1 : \ldots, DM2 : \ldots, D : \ldots \triangleright G : \ldots]$

S2: K (A ∪ B) ist laut DeMorgan-1 K (A) ∩ K (B)

$M : \overline{A \cup B} = \overline{A} \cap \overline{B} \quad \overset{forw}{by} \quad Appl(Ass(DM1), Pos(?), Subst(?))$

$Task[DM1 : \ldots, DM2 : \ldots, D : \ldots \triangleright M : \ldots]$

$Task[DM1 : \ldots, DM2 : \ldots, D : \ldots, M : \ldots \triangleright G : \ldots]$

Information-Completeness Correctness Step-Size Relevance

Tasks for the Proof Manager

Information-Completeness $D:\ldots \overset{forw}{by}\ Appl(Ass(DM2), Pos(?), Subst(?))$

- (dynamic) criteria for admissible degree of underspecification
- is underspecified step supported in ΩMEGA?

Correctness $Task[DM1:\ldots, DM2:\ldots \triangleright D:\ldots]$

- yes/no check for task with any suitable theorem prover

Step-Size $Task[DM1:\ldots, DM2:\ldots \triangleright D:\ldots]$

- judge size of quality proof (plan)

Relevance $Task[DM1:\ldots, DM2:\ldots, D:\ldots \triangleright G:\ldots]$

- check whether new step occurs in quality proof (plan)

Outlook: Near Future

- further analysis of corpus
- implementation of demonstrator
  - interfaces between modules
  - elementary module functionalities
- second round of experiments

Outlook: Next Funding Period

- New methodology:
  - so far top-down
    (modelling $\longrightarrow$ empirical studies $\longrightarrow$ implementation)
  - now bottom-up by systematically extending the demonstrators functionalities, thereby focusing on particular research aspects required
- Broaden corpus: other tutorial strategies, domains
- Restrict corpus: interaction, freedom of input
- Empirical: testing the usefulness of full-fledged natural language dialog is more important than testing tutorial strategies