
Assertion-Level Proof Representation with Under-Specification

Serge Autexier, Christoph Benzmüller, Helmut Horacek, Bao Quoc Vo
Armin Fiedler

`afiedler@cs.uni-sb.de`

`http://www.ags.uni-sb.de/~afiedler`

Universität des Saarlandes



Motivation

- **Aim:** natural-language tutorial dialogue with a mathematical assistant system

Motivation

- **Aim:** natural-language tutorial dialogue with a mathematical assistant system
- **Problem:**
 - natural-language dialogue about proofs
 - mapping between natural-language proofs and formal proofs

Motivation

- **Aim:** natural-language tutorial dialogue with a mathematical assistant system
- **Problem:**
 - natural-language dialogue about proofs
 - mapping between natural-language proofs and formal proofs
- **Solution:** intermediate representation of proofs that allows for under-specification

Intelligent Tutoring Systems



- support student in solving a problem in a specific domain

Intelligent Tutoring Systems



- support student in solving a problem in a specific domain
- Problems:
 - Domain Modeling
 - **static**: precompiled solutions

Intelligent Tutoring Systems



- support student in solving a problem in a specific domain
- Problems:
 - Domain Modeling
 - **static**: precompiled solutions
 - User Interaction
 - dialogue **menu-** or **keyword-**based

Intelligent Tutoring Systems



- support student in solving a problem in a specific domain
 - Problems:
 - Domain Modeling
 - **static**: precompiled solutions
 - User Interaction
 - dialogue **menu-** or **keyword-**based
- ⇒ **dynamic** solutions with Ω MEGA

Intelligent Tutoring Systems



- support student in solving a problem in a specific domain
 - Problems:
 - Domain Modeling
 - **static**: precompiled solutions
 - User Interaction
 - dialogue **menu-** or **keyword-**based
- ⇒ **dynamic** solutions with Ω MEGA
natural-language communication

Contributions

- interactive human-oriented proving
 - What is a *human-oriented* proof?

Contributions

- interactive human-oriented proving
 - What is a *human-oriented* proof?
 - ⇒ empirical investigation collecting a corpus

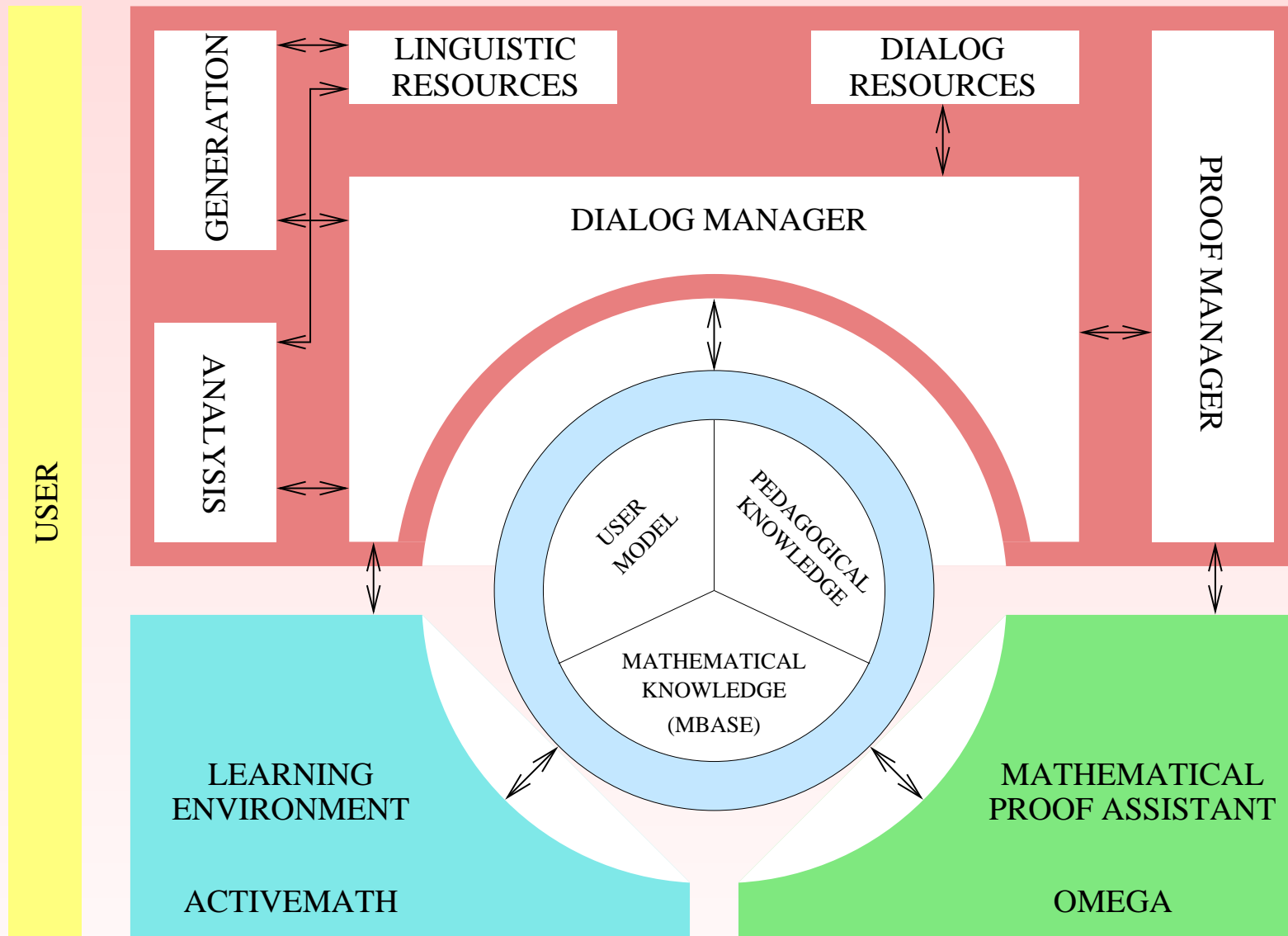
Contributions

- interactive human-oriented proving
 - What is a *human-oriented* proof?
 - ⇒ empirical investigation collecting a corpus
- user-interfaces for theorem provers
 - What are the *requirements* at the proof representation?

Contributions

- interactive human-oriented proving
 - What is a *human-oriented* proof?
 - ⇒ empirical investigation collecting a corpus
- user-interfaces for theorem provers
 - What are the *requirements* at the proof representation?
 - ⇒ separation of proof representation in the user interface and in the theorem prover

Architecture



Domain: Naive Set Theory

- **simple** domain
 - suitable for empirical studies
 - in-depth domain modeling feasible
 - manageable by Ω MEGA
 - appropriate for multi-modal interaction

Domain: Naive Set Theory

- **simple** domain
 - suitable for empirical studies
 - in-depth domain modeling feasible
 - manageable by Ω MEGA
 - appropriate for multi-modal interaction
- *too* simple?
 - no embedded theories

Wizard-of-Oz Experiments



- subjects **interact** with system
- human **wizard** simulates the system

Wizard-of-Oz Experiments

- subjects **interact** with system
- human **wizard** simulates the system
- advantages:
 - test architectures and algorithms before (complete) implementation

Wizard-of-Oz Experiments

- subjects **interact** with system
- human **wizard** simulates the system
- advantages:
 - test architectures and algorithms before (complete) implementation
 - evaluate implemented modules before completed implementation of the whole system

Wizard-of-Oz Experiments

- subjects **interact** with system
- human **wizard** simulates the system
- advantages:
 - test architectures and algorithms before (complete) implementation
 - evaluate implemented modules before completed implementation of the whole system
 - abstract from problematic parts

Wizard-of-Oz Experiment



- subjects had to prove three theorems:

- warm-up:

$$K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D)).$$

- If $A \subseteq K(B)$, then $B \subseteq K(A)$.

- $A \cap B \in P((A \cup C) \cap (B \cup C))$.

K stands for the complement of a set

P stands for the powerset of a set

An Example Dialogue



T1: Bitte zeigen Sie : $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$

[Please show: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$]

S1: nach deMorgan-Regel-2 ist $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cup K(C \cup D))$

[by deMorgan-Rule-2 $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cup K(C \cup D))$ holds]

T2: Das ist richtig!

[This is correct!]

S2: $K(A \cup B)$ ist laut deMorgan-1 $K(A) \cap K(B)$

[$K(A \cup B)$ is $K(A) \cap K(B)$ according to deMorgan-1]

T3: Das stimmt auch.

[That is also right.]

S3: und $K(C \cup D)$ ist ebenfalls laut deMorgan-1 $K(C) \cap K(D)$

[and $K(C \cup D)$ is also $K(C) \cap K(D)$ according to deMorgan-1]

T4: Auch das stimmt.

[That also is right.]

An Example Dialogue (cont'd)



S4: also folgt letztendlich: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$.

[hence follows finally: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$.]

T5: Das stimmt genau. Ich wiederhole noch einmal: Auf die linke Seite der Gleichung kann ich zuerst die zweite und danach die erste deMorgan-Regel anwenden, so daß sich folgende Argumentationskette ergibt:

$K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$.

[This is absolutely correct. I repeat once more: On the left side of the equation I can apply first the second and then the first deMorgan rule, such that we achieve the following chain of argument:

$K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$.]

Proof Step Categorisation



- **Information Completeness:** Has all necessary information been provided?

Proof Step Categorisation



- **Information Completeness:** Has all necessary information been provided?
- **Accuracy:** Is the provided information accurate?

Proof Step Categorisation

- **Information Completeness:** Has all necessary information been provided?
 - **Accuracy:** Is the provided information accurate?
- ⇒ **correct** iff complete and accurate

Proof Step Categorisation

- **Information Completeness:** Has all necessary information been provided?
 - **Accuracy:** Is the provided information accurate?
- ⇒ **correct** iff complete and accurate
- **Step Size:** Has the provided proof an acceptable granularity?

Proof Step Categorisation

- **Information Completeness:** Has all necessary information been provided?
 - **Accuracy:** Is the provided information accurate?
- ⇒ **correct** iff complete and accurate
- **Step Size:** Has the provided proof an acceptable granularity?
 - **Relevance:** Is the provided step relevant for the proof under consideration?

Information Completeness



- Has all necessary information been provided?

T1: Bitte zeigen Sie: $A \cap B \in P((A \cup C) \cap (B \cup C))!$

[Please show: $A \cap B \in P((A \cup C) \cap (B \cup C))!$]

S1: $= P((A \cap B) \cup C)$

- Is the provided information accurate?

T1: Bitte zeigen Sie: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))!$
[Please show: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))!$]

S1: $K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D)$

S2: $K(A \cup B) \cup K(C \cup D) = K(A) \cap K(B) \cup K(C) \cap K(D)$

Step Size



- Has the provided proof an acceptable granularity?

S6: da $a \subseteq U \setminus B$, gilt dass a keine elemente enthält, die in b vorkommen
[since $a \subseteq U \setminus B$, it holds that a contains no elements that occur in b]

S7: daraus folgt, dass $K(a) \supseteq b$
[from that follows that $K(a) \supseteq b$]

Relevance

- Is the provided step relevant for the proof under consideration?

T1: Bitte zeigen Sie: Wenn $A \subseteq K(B)$, dann $B \subseteq K(A)$!
[Please show: If $A \subseteq K(B)$, then $B \subseteq K(A)$!]

S1: $b \notin k(b)$

Under-Specification



- information often missing in user's utterances

Under-Specification



- information often missing in user's utterances
 - if **forward** or **backward** step

Under-Specification



- information often missing in user's utterances
 - if **forward** or **backward** step
 - **references** to used assertions

Under-Specification



- information often missing in user's utterances
 - if **forward** or **backward** step
 - **references** to used assertions
 - **instantiations** of quantified variables

Under-Specification



- information often missing in user's utterances
 - if **forward** or **backward** step
 - **references** to used assertions
 - **instantiations** of quantified variables
 - **positions** of considered subformulae

Example



T1: Bitte zeigen Sie: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))!$

[Please show: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))!$]

S1: $K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D)$

T2: Das ist richtig! *[That is correct]*

S2: $K(A \cup B) = K(A) \cap K(B)$

T3: Das ist auch richtig! *[That is also correct!]*

S3: $K(C \cup D) = K(C) \cap K(D)$

T4: Genau! *[Exactly!]*

S4: wenn man das nun zusammensetzt ergibt sich der beweis
[if one puts that together now, the proof is achieved]

Categories of Proof Steps



- analysis of the corpus
 - derivation of new facts with potentially under-specified references to used facts
 - introduction of new subgoals referring to replaced goals
 - decomposition of goal formula introducing new hypotheses
 - assignment of values to instantiable variables
 - introduction of abbreviations
 - signal of end of proof

Proof Representation Language



Step	$S ::=$.
		Trivial
		Fact $N:F$ from $R^*;S$
		Subgoals $(N:F)^+$ for R by R^* in S^+ End
		Assume H^* prove $N:F$ (from R) in S End
		Assign ($SUBST$ $ABBRV$); S
		Or(S_1 ... S_n)
		Cases $F^+ : ($ Case $N:F : S$ End) $^+$ End
Hypotheses	$H ::=$	$N : F$ $CONST : TYPE?$ $VAR : TYPE?$
Substitutions	$SUBST ::=$	Let $VAR := TERM$
Abbreviations	$ABBRV ::=$	Let $CONST := TERM$
Constants	$CONST ::=$	const N
Variables	$VAR ::=$	var N
Types	$TYPE ::=$...

Example Proof

S1: *[by deMorgan-Rule-2 $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cup K(C \cup D))$ holds]*

Fact $\therefore K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$

from (deMorgan-Rule-2,...);

S2: *[$K(A \cup B)$ is $K(A) \cap K(B)$ according to deMorgan-1]*

Fact $\therefore K(A \cup B) = K(A) \cap K(B)$ from (deMorgan-Rule-1,...);

S3: *[and $K(C \cup D)$ is also $K(C) \cap K(D)$ according to deMorgan-1]*

Fact $\therefore K(C \cup D) = K(C) \cap K(D)$ from (deMorgan-Rule-1,...);

S4: *[hence follows finally: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$.]*

Trivial

Proof Checking



$$\frac{P : \Gamma \Longrightarrow_{\text{Triv}} \Delta}{\Gamma \langle \text{Trivial} \rangle \Delta} \text{ Trivial} \quad \frac{\Gamma[S_i]\Delta}{\Gamma \langle \text{Or}(S_1 \parallel \dots \parallel S_n) \rangle \Delta} \text{ Or} \quad \frac{P(R^*) : \Gamma \Longrightarrow_{\text{Fact}} F, \Delta \quad \Gamma, N : F[S]\Delta}{\Gamma \langle \text{Fact } N : F \text{ from } R^*; S \rangle \Delta} \text{ Fact}$$

$$\frac{P(R, R^*) : \Gamma, (F_1 \wedge \dots \wedge F_k) \Longrightarrow_{\text{Subgoal}} \Delta \quad \Gamma[S_1]N_1 : F_1, \Delta \quad \dots \quad \Gamma[S_k]N_k : F_k, \Delta}{\Gamma \langle \text{Subgoals } N_1 : F_1, \dots, N_k : F_k \text{ for } R \text{ by } R^* \text{ in } S_1 \mid \dots \mid S_k \text{ End} \rangle \Delta} \text{ Subgoals}$$

$$\frac{P(R) : \Gamma, F \Longrightarrow_{\text{Focus}} \Delta \quad P'(R) : \Gamma \Longrightarrow_{\text{Hyp}} (H_1 \wedge \dots \wedge H_n), \Delta, \quad \Gamma, H_1 \wedge \dots \wedge H_n[S]N : F, \Delta}{\Gamma \langle \text{Assume } H_1, \dots, H_n \text{ prove } N : F \text{ (from } R \text{) in } S \text{ End} \rangle \Delta} \text{ Assume}$$

$$\frac{\text{var } x : \tau \in \Gamma \quad \Gamma \Longrightarrow_{\text{Type}} t : \tau \quad P : \Gamma \Longrightarrow_{\text{Subst}} x = t, \Delta \quad \Gamma, \dots : x = t[S]\Delta}{\Gamma \langle \text{Assign var } x := t; S \rangle \Delta} \text{ Assign-Subst}$$

$$\frac{\Gamma \Longrightarrow_{\text{Type}} t : \tau \quad c \notin \Gamma \cup \Delta \quad \Gamma, \dots : \text{const } c : \tau, \dots : c = t[S]\Delta}{\Gamma \langle \text{Assign const } c := t; S \rangle \Delta} \text{ Assign-Abbrv}$$

$$\frac{P : \Gamma \Longrightarrow_{\text{Case}} F_1 \vee \dots \vee F_n, \Delta \quad \Gamma, N_1 : F_1[S_1]\Delta \quad \dots \quad \Gamma, N_n : F_n[S_n]\Delta}{\Gamma \langle \text{Cases } F_1, \dots, F_n : \text{Case } N_1 : F_1 : S_1 \text{ End} \dots \text{Case } N_n : F_n : S_n \text{ End End} \rangle \Delta} \text{ Case}$$

Related Work

- Abel, Chang and Pfenning (2001)
 - proof representation language for interactive theorem proving
 - first-order constructive logic

Related Work

- Abel, Chang and Pfenning (2001)
 - proof representation language for interactive theorem proving
 - first-order constructive logic
 - no user-adaptation
 - no under-specification

Related Work

- Abel, Chang and Pfenning (2001)
 - proof representation language for interactive theorem proving
 - first-order constructive logic
 - no user-adaptation
 - no under-specification
- Kamareddine, Maarek and Wells (2003)

Related Work

- Abel, Chang and Pfenning (2001)
 - proof representation language for interactive theorem proving
 - first-order constructive logic
 - no user-adaptation
 - no under-specification
- Kamareddine, Maarek and Wells (2003)
 - Listen to their talk tomorrow, 16:00

Related Work

- Abel, Chang and Pfenning (2001)
 - proof representation language for interactive theorem proving
 - first-order constructive logic
 - no user-adaptation
 - no under-specification
- Kamareddine, Maarek and Wells (2003)
 - Listen to their talk tomorrow, 16:00
- ?

Conclusion

- corpus of natural language data on proofs
 - human-oriented proofs
 - aspects to categorise proof steps

Conclusion

- corpus of natural language data on proofs
 - human-oriented proofs
 - aspects to categorise proof steps
- separate proof representation that allows for underspecification
 - useful for user interfaces of theorem provers