Project MI 3:

DIALOG:
Natural Language-based Interaction with a Mathematics Assistance System

2.1 General Information

2.1.1 Topic

Natural Language-based Interaction with a Mathematics Assistance System

2.1.2 Scientific Discipline and Field of Work

- Computational Linguistics: dialog modeling, natural language understanding
- Computer Science: automated theorem proving, mathematical knowledge management

2.1.3 Leaders

Name, date of birth	(1) Pinkal, Manfred (24.8.1949) (2) Siekmann, Jörg (5.8.1941) (2) Benzmüller, Christoph (08.09.68) (4) Kruijff-Korbayová, Ivana (22.11.67)
Office address	(1,4) FR Computerlinguistik / (2,3) FR Informatik Universität des Saarlandes D-66041 Saarbrücken
Phone	(1) 0681-302-4343 / (2) 0681-302-5275 (3) 0681-302-4501 / (4) 0681-302-4574
Fax	(1,3) 0681-302-4351 / (2,4) 0681-302-5076
Email	{pinkal, korbay} @ coli.uni-sb.de {siekmann, chris} @ ags.uni-sb.de

Are the positions of the project leaders for a restricted period of time?

Pinkal	No
Siekmann	No
Benzmüller	Yes, (currently) restricted until 31.12.2012.
Kruijff-Korbayová	Yes, (currently) restricted until 31.12.2005.

2.1.4 Transfer of the Project

The project is not being transferred to the Transfer Center.

2.1.5 Planned Experiments

Studies with human subjects	[X] Yes	[ ] No
Studies with human embryonal stem cells	[ ] Yes	[X] No
Clinical studies on somatic cell or gene therapy	[ ] Yes	[X] No
Experiments with animals	[ ] Yes	[X] No
Studies involving genetic technology	[ ] Yes	[X] No

2.1.6 Previous and proposed funding of the project so far in the context of the collaborative research center (*external funding*)

Fiscal year	Personnel costs	Administrative expenses	Investments	Total
Through 2001				
2002	119.141			114.027
2003	120.796			115.796
2004	122.800			117.600
Subtotal	347.423			347.423
2005	147.600			
2006	147.600			
2007	147.600			

(Amounts in 1000s of EUR)

2.2 Summary

The final goal of the DIALOG project is a natural tutorial dialog between a student and an assistance system for mathematics.

Based on the research of the previous project phase, we will concentrate on novel and deep problems regarding the interpretation and evaluation of proof step utterances.

The most important problems we shall concentrate on in the final funding period are:

- the linguistic analysis of informal input to the tutoring system combining deep and shallow processing methods,

- the evaluation of user utterances describing proof steps with respect to soundness, granularity, and relevance, and
- the disambiguation of ambiguous utterances using a combination of discourse and domain knowledge and underspecification techniques.

We expect the results of our research to be highly relevant for both dialog processing and mathematical knowledge management. Furthermore, the results obtained in this project using mathematics as a case study may be generalized easily to dialogs on other domains of technical nature.

2.3 State of the Art

In its first project phase, the DIALOG project met a new multi-disciplinary scientific challenge situated between (i) advanced natural language processing, (ii) flexible tutorial dialog, and (iii) dynamic, abstract level mathematical domain reasoning. In each of these fields, there are results to base our research on. However, we entered a mainly unstructured area when we started to combine the different aspects to a dialog-based math-tutoring system. Not only research results to build on but also empirical data, both of which could guide our research, were missing. We, therefore, approached the project by using a methodology with a strong initial emphasis on empirical investigations and a top-down modeling of the over-all architecture followed by refinements of the architecture, down to implementation.

We do not know about any competing projects on flexible math-tutoring dialog within the reported period. There has however been considerable growth and differentiation in all the three research areas involved, yielding results of immediate relevance to our project. In the field of dialog modeling, general solutions on the conceptual and also on the software level are becoming increasingly available. The emphasis of research is rather on the development of inter-operable individual modules, that are specialized in single, specific tasks, than on the development of big monolithic systems. This move is analogous to recent developments in the areas of mathematical assistance systems and tutorial systems. The changed situation is reflected in the over-all concept for the second phase of our project, which will be focused on a selection of relevant research questions rather than on a complete realization of a standalone architecture. To arrive at a complete system, we will be able to take advantage of several complementary developments in other research projects.

In the following, we will briefly sketch some specific research results of the last few years.

More detailed information about previous research we are building on is given in the goals and work program description, where necessary.

The only recent work we are aware of that addresses mathematical text processing is (Zinn, 2003), where an extended Discourse Representation Theory (DRT) approach is applied to underspecification resolution. Like the earlier work (Zinn, 1999; Baur, 1999), (Zinn, 2003) analyzes complete, carefully structured textbook proofs, and relies on given text-structure, typesetting and additional information that identifies mathematical symbols, formulae, and proof steps. Both Baur and Zinn provide useful insights, but of only limited impact in our setting, because of important differences between proofs in textbooks and in a tutorial dialog.

In the area of robust approaches to (large scale) text processing, recent developments witness a shift of interest from surface-syntax oriented analysis (e.g., part-of-speech tagging and surface-syntax bracketing) toward deeper levels of analysis, such as assigning lexical semantics and deep semantic structures.

With respect to our goal of ambiguity and underspecification resolution, (Bos, 2003) provides an algorithm for efficient presupposition and anaphora resolution in DRT, which uses state-of-the-art traditional automated theorem provers for checking consistency and informativeness conditions. We will go beyond this by also checking plausibility and relevance, and by employing a mathematical domain reasoning system with access to large domain knowledge.

The mathematical knowledge management research initiative[1] fosters the transition from pen-and-paper practice in mathematics to computer-supported mathematial practice. The lack of human-oriented representations of mathematical knowledge is recognized as one of the bottlenecks, and our research contributes to overcome it. We also identify additional aspects arising in the tutorial context.

2.4 Own Preliminary Work

Important fundamental contributions to the project have been made in the first project phase. We briefly list the main results below. They are described in more detail in the project report.

- Experiment design and an empirical test environment realized in a tool for Wizard of Oz experiments addressing flexible tutorial dialog in mathematical problem solving.
- First experiment in the naive set theory domain, with written dialog input and output. (The

---

[1] `http://mizar.uwb.edu.pl/MKM2004/`

second experiment is scheduled for the final year of the first project phase; the aim here is to collect additional empirical data in combined written and spoken mode accompanied with simple mouse pointing.)

- Preliminary corpus investigation and subsequent formal annotation at several levels of interpretation: deep semantic structure, dialog moves, and tutorial task aspects.
- Coarse grained architecture for a flexible dialog system for tutoring mathematical problem solving and specification of refined modules for input analysis, proof management and tutorial dialog moves (especially hinting).
- Realization of input analysis using a deep dependency-based grammar, focusing on uniform interpretation of informal interleaved natural language and mathematical formulae.
- Realization of the proof manager: proof representation languages for the proof manager, interfacing to the underlying domain reasoner (the Omega theorem prover); agent-based assertion reasoning that can enumerate proof step suggestions.
- Development of a demonstrator (in progress at the time of writing, to be completed before the review meeting).

Analysis of the corpus collected in our first project experiment revealed an ambitious range and heterogeneity of our multi-faceted research project. It also provided crucial insights concerning the requirements on input analysis and domain reasoning in flexible mathematical problem solving tutorial dialog. The research questions arising for the second project phase are addressed in Section 2.5.

In the following we list other available research results from the last years in which our groups are involved:

In the OMEGA project techniques for abstract-level theorem proving, in particular, knowledge-based proof planning, have been developed and applied. The OMEGA mathematical proof assistance environment integrates and orchestrates various external specialist reasoners and provides tools for the maintenance and management of mathematical knowledge.

In the Chorus project, techniques have been developed for representation and efficient direct processing of ambiguity and underspecification. We will investigate the application of these techniques in our system set-up in the following project phase.

The SALSA[2] project employs the FrameNet[3] paradigm to develop large scale lexical semantic resources on the basis of the TiGer[4] corpus. We want to make use of them for robust deep and shallow linguistic analysis of input utterances.

We have gained experience with Information-State-Update (ISU) based dialog management in systems supporting flexible dialog and user-initiative in the Siridus[5] project. In Siridus, we focused on modular solutions and reconfigurability for different domains. In the new EU project TALK[6] (Talk & Look: Tools for Ambient Linguistic Knowledge), we continue pursuing the issues of generic solutions for modeling flexible and adaptive dialog along with advanced Information State modeling, and we extend the ISU-based approach to multimodal dialog processing.

Finally, an innovative third generation eLearning system supporting both classroom tutoring and self-learning is being developed in the new EU project LeActiveMath[7]. It adapts to the learner and the learning context and comprises personalization, tutorial dialogs, open student modeling and interactivity that is tool-supported for active and exploratory learning. The LeActiveMath project is coordinated by Erica Melis. Her research group, which is part of the AG Siekmann, develops the ActiveMath Learning Environment.[8]

2.5 Work Plan (Goals, Methods, Schedule)

2.5.1 Goals

In the initial project phase, we made a first start to open up a new field of natural-language based mathematical tutoring dialogs. Our foremost aim was to obtain a general view of the interplay between advanced natural language processing in a flexible tutoring dialog, and dynamic, abstract level mathematical domain reasoning, while moving from the collection of empirical data through modeling of the different components and their interfaces to a demonstrator implementation.

Our work in the second project phase will focus on three important research questions that have grown out of this research:

---

[2] `www.coli.uni-sb.de/lexicon`
[3] `www.icsi.berkeley.edu/~framenet/`
[4] `www.ims.uni-stuttgart.de/projekte/TIGER/`
[5] `www.ling.gu.se/projekt/siridus/`
[6] `www.talk-project.org`
[7] `www.leactivemath.de`
[8] `www.activemath.org`

- Processing *informal* input that consists of *interleaved* natural language text and mathematical expressions. We will start with the framework for deep semantically-oriented analysis of this special kind of multi-modal utterances that has been developed in the first project phase. In order to reconcile detailed and reliable target representations with the requirement to deal with sloppy, incomplete, partially ungrammatical or out-of-grammar input, we shall *combine* the *deep analysis* approach from the first project phase *with methods for flat and partial interpretation*. We will extend the analysis methodology to simultaneous written and spoken input, combined also with mouse pointing.
- We will investigate criteria and methods for *proof step evaluation*. The linguistic analysis yields a formal representation of a proof step as proposed by the user. This proof step has to be evaluated as to its soundness and appropriateness with respect to the proof problem at hand, in order to complete the interpretation and to select the correct response. While evaluation of *soundness* is straightforward (it amounts to the verification of a proof step by a theorem prover, which delivers just ‘yes’ or ‘no’ results independent of the tutorial context), evaluating the appropriateness of a proposed proof step, viz., the right *granularity* and *relevance*, is much more demanding. The system must be able to discriminate between target-oriented proof-step proposals and all the rest of syntactically correct and logically sound, but irrelevant, tautological, or misleading steps. To the best of our knowledge, this topic has not been addressed in the literature, and it is very unclear how the notion of relevance can be captured in a general and operational way. Ideas of how to obtain partial and approximative relevance estimates are outlined in WP 2.
- *Ambiguity* pervades all levels of processing: We shall continue our work on the representation of ambiguous input, and then tackle the new task of how to resolve these ambiguities in the given context. There are various linguistic sources of ambiguity, as well as utterances which are linguistically unambiguous but nevertheless do not provide enough information for an unambiguous mapping to a domain-specific interpretation. In the first project phase, a language (LU) was developed that is able to handle ambiguity of the latter kind, and different methods have been applied to treat ambiguous linguistic input. The real challenge is however *resolution* based on domain knowledge. It may be easy to sort out those interpretations which would lead to illegal operations or unsound proof steps. In other cases, we need some formal relevance estimates. We will collaborate with the CHORUS project on the representation and *processing of ambiguity in an underspecification framework*. In particular, we want to explore how the CHORUS techniques to reduce or resolve ambiguity without complete enumeration of readings can be applied to underspecification in the mathematical dialog domain.

All three research questions outlined above focus on user input interpretation, and our solutions will contribute to a comprehensive and integrated model of the interpretation of student utterances in tutorial dialogs. This interpretation and final representation of the users' input provide the essential information for the choice of an appropriate tutor system response.

Empirical studies and demonstrator implementation will also be part of our work in the proposed project phase. Unlike in the first phase, however, both will be done in a more focused and goal directed manner. The demonstrator architecture of phase 2 will build on the demonstrator of the first project phase. However, the individual modules will be worked out at very different levels of detail and coverage. We will concentrate on the full realization of the interpretation process. Other components of the first-phase demonstrator (e.g., generation, dialog model and tutoring strategy) will be realized in the final demonstrator only as far as they are necessary to provide a natural scenario and context for user utterance interpretation, and for the demonstration and evaluation of the interpretation model. Experiments will be carried out to obtain appropriate measures for the different dimensions of proof-step evaluation.

We should mention that several strands of research pursued in the first project phase will be continued outside the dialog project, e.g., generic dialog structure and aspects of multimodality in the EU project TALK and tutorial aspects of mathematical dialog in the EU-project LeActiveMath.

2.5.2 Methods and Work Packages

The proposed research is broken down into four work packages.

WP1: Interpretation of Informal Mathematical Input

Investigation of the dialogs collected experimentally in the first phase revealed a range of language phenomena that occur frequently and present challenges for input interpretation. While the phenomena themselves are not new, the genre of informal mathematical dialog adds new twists to them due to mathematical content being verbalized (i) to varying degree, resulting in *a mixture of natural language and mathematical expressions*, (ii) *informally* and *imprecisely*. These characteristics permeate input analysis at the sentence-level, discourse-level as well as at the level of domain-specific interpretation. In the first phase of the project, we identified key challenges at each level, and started addressing them. To extend this work is the task of this work package as well as WP3 where we address specifically the problem of ambiguity and underspecification at all levels of interpretation.

Task 1.1 Robust Sentence-Level Analysis

In the first project phase, we developed a dependency-grammar based approach to deep analysis using the grammar formalism of Multi-Modal Combinatory Categorial Grammar (Baldridge, 2002; Baldridge & Kruijff, 2003). We focused on achieving uniform analysis of inputs with different degree of mathematical content verbalization. In a preprocessing step, mathematical expressions are identified, analyzed, categorized, and substituted with default lexicon entries. Then the input is syntactically parsed using openCCG[9], an open source MMCCG parser. The meaning representation constructed compositionally along with the parse is represented using *Hybrid Logic Dependency Semantics* (HLDS) (Kruijff, 2001; Baldridge & Kruijff, 2002).[10]

In the first phase, we concentrated on processing written, syntactically well-formed input within the coverage of the grammar. We are also aware that experimentally collected dialogs contain also ill-formed or incomplete utterances (because students make errors and/or write telegraphically) and out-of-grammar utterances. These all lead to deep analysis failures. In such cases, the system could initiate a correction subdialog. However, it is not desirable to go into (syntactic) details distracting the student from the main tutoring goal (and the human tutor in our experiment did not do this either). Therefore, we want to develop more robust methods of input analysis, handling also some ill-formed or out-of-grammar input. On the other hand, a tutorial system must not be too accommodating (in constructing a domain-specific interpretation from “too little” input). We plan to carry out additional experiments targeting this particular problem.

For reasons that motivate the use of deep analysis in the first place, the techniques for robust input processing cannot be purely statistical or simple keyphrase-spotting. Rather, we need to combine deep and shallow methods in a way that will allow us to arrive at (i) partial syntactic analyses, and (ii) partial domain-specific interpretation(s). We plan to achieve this by a combination of methods: On the one hand, we will read off partial analyses from the chart built by the parser during deep syntactic analysis, and construct underspecified semantic representations from them. On the other hand, we will parse the input with an external syntactic parser (e.g. (Dubey & Frank, 2003) or (Braun, 2003)), or a shallow chunk parser (e.g. (Skut & Brants, 1998)), and then apply deep syntactic analysis to the chunks.

---

[9] `http://openccg.sourceforge.net`

[10] We have been collaborating with the NEGRA project in which the same approach is employed at the level of syntactic representation.

Dialog in written mode is not the most natural and flexible form of interaction in general, and especially in a one-to-one tutorial setting, where a combination of speech, writing and drawing would be typical. Therefore, we also want to extend the input analysis methods to processing a simultaneous written and spoken input (accompanied with simple pointing/selection on the screen) using data collected in the second experiment, which will still be carried out in the first project phase. To avoid the pitfalls of contemporary automatic speech recognition, we will first work on transcriptions of the spoken input. We expect that: (i) linguistic as well as mathematical content will be distributed and possibly redundantly presented in the different modes; (ii) users will formulate ideas differently in spoken and written mode, so we will identify new structures to analyze; (iii) there will be different kinds of telegraphic or syntactically ill-formed input, and even more underspecification of the mathematical expressions (e.g., missing parentheses).

Task 1.2 Discourse Representation of Informal Mathematical Input

In the first project phase, we developed a rudimentary HLDS-based solution to discourse modeling by applying the proposal presented in (Kruijff, 2001; Kruijff & Kruijff-Korbayová, 2001). We plan to extend the account to properly treat the (non-)accessibility of discourse referents due to (i) scopes of various types of quantifiers, (ii) discourse relations, and (iii) information structure. This will enable us to cope with genre-specific discourse phenomena we identified in the corpus.

The antecedent of an anaphor can be (a part of) a math expression. (1) illustrates a demonstrative anaphoric expression whose antecedent is a formula; (2) illustrates pronominal coreference where the antecedent of “es” is B in $B\subseteq K(A)$;

(1) $A \subseteq K(B)$ **daraus** folgt $B \subseteq K(A)$

(2) Und wenn $B\subseteq K(A)$ sein soll, muss **es** auch Element von $K(A)$ sein.

(3) illustrates a metonymic anaphoric expression which refers to a structural sub-part of a formula, resulting in a predicate structure acceptable informally, yet incompatible in terms of selection restrictions: The predicate **be_valid_for**, in this domain, normally takes an argument of sort CONSTANT, TERM or FORMULA, rather than LOCATION.

(3) Dann **gilt fuer die linke Seite**, wenn $C \cup (A \cap B) = (A \cup C) \cap (B \cup C)$, der Begriff A ∩ B dann ja schon dadrin und ist somit auch Element davon.

(4,5) illustrate discourse deixis typical for the descriptions of subsequent reasoning steps.

(4) **T:** Ist einer dieser Sätze die Antwort auf meine Frage? Falls ja, welcher?
S: Der letzte Satz.

(5) **T3:** Aus welcher Regel haben Sie Ihre erste Schlußfolgerung abgeleitet ?
S4: aus **der regel in der zweiten zeile**
T4: Die zweite Zeile wovon?
S6: ich habe versucht, meine erste schlussfolgerung mit **der zweiten zeile meiner Aussagen** zu erklaeren. ich meinte **die zweite zeile meiner aussage**. also ist auch $A \cap B \in P((A \cup C) \cap (B \cup C))$

To handle this range of anaphoric reference phenomena, we want to address the question which parts of mathematical expressions and of the informal mathematical text should have explicit semantic representation, and thus be available for anaphoric reference. We want to motivate this choice empirically by attested systematic occurrences of natural language references to parts of math expressions (e.g., “the left/right side”, “the parenthesis”, and “the inner parenthesis”) and by the syntactic contexts in which they occur (e.g., the partitioning <[x][∈A]> seems well motivated in “B contains no x∈A”; [x ∈] is a constituent in “x ∈ of complement of B.”)

Two occurrences of the same symbolic identifier need not be co-referential. In our initial approach to discourse modeling in the first project phase, we treated identifiers of mathematical expressions the same way as proper nouns in DRT: as rigid designators, i.e., all occurrences of the same symbolic identifier throughout a dialog (or throughout a segment where an identifier is declared) are co-referential.

This treatment presupposed that symbolic identifiers were used in a formally proper way. However, the experimentally collected data reveal much more liberal use of identifiers, and therefore it becomes necessary to determine when two occurrences are co-referential and when not.

(6) (a) Da, wenn $A \subseteq K(B)$ sein soll, A Element von $K(B)$ sein muss. (b) Und wenn $B \subseteq K(A)$ sein soll, muss es auch Element von $K(A)$ sein.

(7) DeMorgan-Regel-2 besagt: $K(A \cap B)$ = $K(A) \cup K(B)$ In diesem Fall: z.B. $K(A)$ = dem Begriff $K(A \cup B)$ $K(B)$ = dem Begriff $K(C \cup D)$

In (6), A and B are indeed co-referential within the individual utterances (6a) and (6b). Across the two utterances, the symbols are not used coreferentially. In DRT terms, A and B introduce discourse referents in separate universes of two parallel implicative conditions. (7) is a particularly tricky example of mixing co-referential and non-coreferential use of symbolic identifiers: since $K(A)$ in the De Morgan rule is to be substituted with the expression $K(A \cup B)$, A is clearly used non-coreferentially.

Discourse relations indicate intended proof structure and co-determine reference. Discourse relations, whether signaled explicitly by discourse connectives or not, are one aspect of discourse structuring: in our genre, the rhetorical structure of the discourse often closely mirrors the structure of the proof and the involved reasoning, even though the mapping between the discourse level and the domain specific proof prepresentation is not necessarily one-to-one.

(6) above is a nice illustration of the interplay between discourse relations/structure and reference: At the rhetorical structure level, it consists of two parallel condition-result pairs that express two independent premise-conclusion pairs at the proof level. This explains the use of the same symbols in a non-coreferential way. We want to model such correlations and make use of them in discourse-level input interpretation.

Task 1.3 Ontology-based domain-specific interpretation

The input interpretation we have developed in the first project phase separated domain-independent lexical semantics (assigning general semantic interpretation of lexical items) from domain-specific interpretation. For example, the verb "enthalten" (as in *A enthält B*) introduces a lexical concept of CONTAINMENT. Subsequently, the domain-specific interpretations (w.r.t. naive-set theory) of CONTAINMENT are assigned, namely, the relations of (strict) subset or element (of a set). So far we have implemented such mappings on case-by-case basis.

We will systematically extend the domain-specific interpretation to cover input containing informal and/or imprecise names of domain concepts and relations. For example, the domain-independent lexical concept of **be_in** (as in *A muss in B sein*) is an informal realization of the domain-specific containment relations of either subset or element. Similarly, **be_outside_of** (as in B **vollstaendig ausserhalb von** A **liegen** *muss*) and **be_different** (as in *dann sind A und B* **(vollkommen) verschieden**), are informal descriptions of the empty intersection of sets. We will also address semantically complex operators such as 'vice-versa' (as in *Wenn alle* A *in* $K(B)$ *enthalten sind und dies auch* **umgekehrt** *gilt*, . . . , which can be interpreted as "alle $K(B)$ in A enthalten sind" or "alle B in $K(A)$ enthalten sind".) Finally, we want to explore the possibility of using the lexical resources developed within the SALSA project (Erk, Kowalski, & Pinkal, 2003) to assign semantic interpretation in cases which are not covered by our semantic lexicon and domain-specific resources.

WP2: Proof Management and Proof-Step Evaluation

In this work package, we will develop a proof manager that evaluates the appropriateness of proof steps based on the criteria *soundness*, *granularity*, and *relevance*. The analysis of these criteria is important for guiding the tutor's response and it also supports the resolution of ambiguities when interpreting the user input in the first place. The latter aspect will be addressed in WP3.

WP2.1 Proof Representation Languages.

We will extend the proof representation languages defined in the first project phase: The proof representation language LU serves as an interface between the analysis module (WP1) and the proof manager. The proof representation language LS provides an interface to the mathematical domain reasoners; it is furthermore employed within the proof manager to maintain a partial proof object representing the proof under development (actually it even represents alternative interpretations of the proof under development by employing OR-branches; see also WP3).

The proof representation language LU is described in Figure 1 and presented in more detail in (Autexier, Benzmüller, Fiedler, Horacek, & Vo, 2004). LU is inspired by the work of Abel and colleagues (Abel, Chang, & Pfenning, 2001). The aim of their assertion-level proof language is to support teaching of constructive first-order proofs.

For the purpose of this proposal, it is sufficient to view LS as an underspecification-free sublanguage of LU. In practice, however, our aim is to develop and employ a language LS that not only serves the requirements of the DIALOG project but at the same time provides an independent interface supporting the integration of different reasoning paradigms (proof planning, interactive proof, agent-based reasoning, external specialist reasoners) in proof assistants. Our most recent design of the proof representation language LS is presented in (Hübner, Autexier, Benzmüller, & Meier, 2004). In the course of the project we also investigate whether it is a good or bad idea to work with different languages LU and LS. We will merge them if it appears useful.

We assume that we already have grammars for the non-terminals name, formula and occurrence. Furthermore we assume a countably infinite set of metavariable symbols $MV = \{M_1, \dots, M_i, \dots\}$ ($i \in Nat$); these metavariables are employed as placeholders for underspecified pieces of information. We define:

$$\begin{array}{rcl} \text{Names } \mathsf{N} & ::= & \mathsf{name} \mid MV \\ \text{Formulae } \mathsf{F} & ::= & \mathsf{formula} \mid MV \\ \text{Occurrences } \mathsf{O} & ::= & \mathsf{occurrence} \mid MV \end{array}$$

Next, we define references as triples:

$$\text{References } \mathsf{R} \quad ::= \quad (\mathsf{N}, \mathsf{F}, \mathsf{O})$$

Now the proof language LU is defined as:

$$\begin{array}{llcl} \text{Step} & S & ::= & MV \\ & & & \mid \texttt{Trivial} \\ & & & \mid \texttt{Fact } N : F \texttt{ from } R^*; S \\ & & & \mid \texttt{Subgoals } (N : F)^+ \texttt{ for } R \texttt{ by } R^* \texttt{ in } S^+ \texttt{ End} \\ & & & \mid \texttt{Assume } H^* \texttt{ prove } N : F \texttt{ (from } R\texttt{) in } S \texttt{ End} \\ & & & \mid \texttt{Assign } (SUBST \mid ABBRV); S \\ & & & \mid \texttt{Or}(S_1 \parallel \dots \parallel S_n) \\ & & & \mid \texttt{Cases } F^+ : (\texttt{Case } N : F : S \texttt{ End})^+ \texttt{ End} \\ & & & \\ \text{Hypotheses} & H & ::= & N : F \mid CONST : TYPE? \mid VAR : TYPE? \\ \text{Substitutions} & SUBST & ::= & \texttt{Let } VAR := TERM \\ \text{Abbreviations} & ABBRV & ::= & \texttt{Let } CONST := TERM \\ \text{Constants} & CONST & ::= & \text{const } N \\ \text{Variables} & VAR & ::= & \text{var } N \\ \text{Types} & TYPE & ::= & \dots \end{array}$$

Figure 1. The proof representation language LU.

The proof representation languages need to be enhanced in several ways. For instance, we want to explicitly represent preference rankings among alternative proof branches. Also we have to differentiate between proof information stemming from user input and proof information that has been contributed by the mathematical domain reasoning tools (in WP3 we will explain how the domain reasoning tools support the resolution of ambiguities and underspecification; thereby new proof information will be generated and instantiated for underspecified parts in the partial proof).

WP2.2: Proof Step Evaluation

Proof step evaluation is an interesting novel application for theorem proving systems. A (next) proof step uttered by a student within a tutorial context has to be analyzed with respect to the following criteria:

Assertions already introduced
(A1) $A \wedge B$.
(A2) $A \Rightarrow C$.
(A3) $C \Rightarrow D$.
(A4) $F \Rightarrow B$.
(G) $D \vee E$.

Alternative proof step directives.
(a) Aus den Annahmen folgt D.
(b) B gilt.
(c) Es genügt D zu zeigen.
(d) Wir zeigen E.

Figure 2. Example Scenario for Proof Step Evaluation.

- **Soundness[11]:** Can the proof step be verified by a formal inference system?
- **Granularity:** Is the granularity (i.e., ’logical size’ or ’argumentative complexity’) of the proof step acceptable?
- **Relevance:** Is the proof step needed or useful in achieving the goal?

Each criterion calls for different theorem proving techniques. Moreover, the most challenging criterion of *relevance* raises novel and interesting proof theoretical questions.

We will illustrate the criteria with a simple example (see Figure 2):

Soundness. Verification of an uttered proof step by a formal inference system is the easiest task. Consider utterance (a) in Figure 2: this proof step describes a forward reasoning step exploiting available knowledge. Verification of the soundness of this utterance boils down to proving the theorem:

$$(A \wedge B), (A \Rightarrow C), (C \Rightarrow D), (F \Rightarrow B) \vdash D \qquad (8)$$

We call this theorem the *proof task for proof step utterance* (a). Analogously, for the backward reasoning step given in (d) we get the proof task:

$$E \vdash (D \vee E) \qquad (9)$$

Solving these proof tasks confirms the soundness of the utterances, and there are no specific requirements imposed on the proof system $\vdash$, i.e., they can be shown by any kind of first-order theorem prover answering “yes” or “no” as the case may be.

---

[11] In some of our papers we called this criterion accuracy.

Soundness is thus a rather simple criterion for which different techniques are already available. Soundness investigation is also addressed in the work of Zinn (Zinn, 2003). Proof step evaluation with respect to the criteria *granularity* and *relevance*, however, are novel and challenging research tasks.

Granularity. Instead of asking for the mere existence of proofs, granularity evaluation requires analyzing the 'complexity' or 'size' of proofs. For utterances (a) and (d) above, it thus boils down to judging about the complexity of the proof tasks (8) and (9).

Let us use Gentzen's natural deduction calculus (Gentzen, 1935) as proof system $\vdash$. As a granularity measure, we determine the number of $\vdash$-steps in the smallest $\vdash$-proof of the proof task for the proof step utterance in question; this number is taken as the argumentative complexity of the uttered proof step.

We consider proof step utterance (a) as an example: here, the smallest natural deduction proof has '3' proof steps (we need one 'conjunction left elimination' step to extract A from $A \wedge B$, one 'modus ponens' step to obtain B from A and $A \Rightarrow B$, and another 'modus ponens' step to obtain C from B and $B \Rightarrow C$). As another example we consider utterance (b): here, the smallest proof requires only '1' step (B follows from assertion $A \wedge B$ by 'conjunction-elimination').

If we now fix a threshold that characterizes, in this sense, the maximally acceptable size of an argumentation then we can distinguish between proof steps whose granularity is acceptable and those which are not. This threshold may be treated as a parameter determined by the tutorial setting.

We doubt, however, that the natural deduction calculus, which we have used above only for illustration purposes, will provide a suitable basis for granularity analysis. The reason is that two intuitively very similar user proof steps may actually expand into natural deduction proofs of completely different size. Thus, our approach would probably rate one step as appropriately granular and the other one as not, while they would be considered as equally complex by human mathematicians. Therefore, the notion of 'argumentative complexity' itself has to be investigated in more depth in order to develop a cognitively adequate measure. An important question concerns the appropriate choice of a proof system $\vdash$. Our hypothesis is that the abstract-level reasoning systems will provide more adequate measures for analyzing argumentative complexity of user proof steps since they better reflect human reasoning.

Relevance. Relevance asks about the usefulness and importance of a proof step with respect to the original proof task. We briefly mention some alternative approaches to approximate the relevance concept; they are listed according to increasing difficulty:

- We could statically choose one or a few "golden proofs" and match the uttered partial proofs against them.
- We first generate from the initially chosen golden proofs larger sets modulo, for instance, (allowed) re-orderings of proof steps and match against this extended set.
- We dynamically support relevance analysis with domain reasoning. For this, we test whether a proof can still be obtained from the new proof situation (using an abstract-level proof system). This alone is not quite sufficient and (resource-bound) enumeration of possible proofs and proof step matching is still required (see below). As the granularity of the human-constructed proof steps in our tutorial setting can be expected to be sufficiently low, we expect the enumeration to be tractable. This approach is our preferred choice for the project and will be illustrated in more detail below.

To illustrate the latter approach we again employ example utterances from Figure 2. In the backward reasoning case (c) the proof goal $D \vee E$ is refined to the new proof goal D, i.e., the previously open goal $D \vee E$ is closed and justified by the new goal. An approach to answer the relevance question in this case is to check whether a proof can still be generated in the new proof situation. In our case the task is thus identical to proof task (8) as before. A backward proof step that is not relevant according to this criterion is (d) since it reduces to the proof task

(10) $(A \wedge B), (A \Rightarrow C), (C \Rightarrow D), (F \Rightarrow B) \vdash E$

for which no proof can be generated. Thus, (d) is a sound refinement step that is not relevant, in contrast to utterance (c).

This simple approach appears plausible but needs to be refined. The challenge is to exclude detours and to take tutorial aspects into account (in a tutorial setting we are often interested in teaching particular styles of proofs, particular proof methods, etc.). This also applies to the more challenging forward reasoning case discussed next.

In example (a) we generate along the same idea the proof task

(11) $(A \wedge B), (A \Rightarrow C), (C \Rightarrow D), (F \Rightarrow B), D \vdash (D \vee E)$

where the inferred assertion D is added to the directly available assertions. The question whether D is relevant reduces to the question whether there exists a proof for the given task that employs D and which is shorter than the best proof that can be obtained when deleting D from the available knowledge. According to this approach utterance (b) describes an non-relevant proof step.

Note that we do not just ask about the existence of an arbitrary proof but about the existence of a proof with particular properties. This requires techniques, such as (resource-bound and heuristic guided) enumeration of proofs, which are non-trivial to realize and which are hardly supported in traditional automated theorem provers.

Proof step relevance is an interesting and ambitious challenge and we will start research in this direction. Generally, the relevance challenge may even stimulate new research in proof theory. For instance, it would be nice to compactly and tractably represent the proofs in the proof space explicitly. Ideally only ‘normal form’ proofs (i.e. proofs without detours representing the whole equivalence class of proofs that reduce to the same normal form) would be stored in a (semi-)lattice structure that would encode some notion of ‘quality’ of proofs and thereby support relevance analysis.

WP2.3: Domain Reasoning support for Proof Step Evaluation

What are the “right” theorem provers to support the evaluation of the different criteria? The task of this work package is to answer this question and to accordingly improve and adapt the selected technology.

For the soundness criterion traditional automated theorem provers are perfectly suited. For granularity, however, we will consider abstract-level proof planning, abstract-level assertion reasoning, and logic-level proof systems such as natural deduction. These proof systems will then be empirically evaluated by mathematicians in small relevance-judgment experiments.

For relevance and underspecification (see WP3) we need an approach that can (resource-bound and heuristically guided) enumerate at least some of the proofs in the proof space. For similar reasons as above we assume that this mechanism should ideally operate on an abstract-level. Therefore, agent-based assertion level reasoning will be our first choice. As for granularity, we will investigate this hypothesis within small empirical experiments.

Our approach to analyze relevance as described above is still very simple and an appropriate solution will have to take plan recognition aspects also into account (knowledge-based proof planning is a good choice of a proof system to support this).

In case no single proof system turns out to be optimally suited to support the evaluation of proof step granularity or relevance we will also investigate combined solutions.

WP3: Domain Reasoning for Ambiguity Resolution

Our corpus contains a broad range of ambiguity and underspecification phenomena. Ambiguity and underspecification can occur and can be resolved at different levels of interpretation (sentence, discourse, domain). For instance, linguistically ambiguous input may obtain an unambiguous domain-specific interpretation or instantiation w.r.t. to the current proof state; on the other hand, linguistically unambiguous input may result in an underspecified domain-specific interpretation or in multiple instantiations w.r.t. to the current proof state. We illustrate this below with examples from the corpus. The general conclusion is that the interpretation process must take into account the global interplay between different aspects of linguistic structure and domain-specific interpretation.

Ambiguities arising at the level of linguistic meaning are introduced by well-known syntactic phenomena, such as attachment and coordination (e.g., $x \in B$ *und somit* $x \subseteq K(B)$ *und* $x \subseteq K(A)$ *wegen Vorraussetzung*), as well as lexical semantics (e.g., the verb “enthalten” can mean the lexical concept of HAVE-PART, when A and B are formulae, or a concept of CONTAINMENT, when A and B are sets). Additional ambiguities arise due to the informal character of the discourse: For example, the parentheses omitted by the student on the right-hand side in (12), can be cooperatively filled in by the system in two ways: $P(C) \cup (A \cap B)$ or $P(C \cup (A \cap B))$.

(12) P((A∪C)∩(B∪C))=PC ∪(A∩B)

More cases like this can be expected when processing spoken input, where parentheses are likely to be left out alltogether.

Ambiguities arise also at the discourse level: Anaphoric expressions have multiple possible antecedents, e.g. “es” in (6b), “die linke Seite” in (3), “daraus” in *daraus folgt* in (1), or “die Zweite Zeile” in (5). Additional ambiguities arise due to specific vs. generic reference (especially when such expressions occur in anaphoric chains). Finally, ambiguities concerning rhetorical relations at the discourse level typically lead to underspecified proof steps, e.g.:

(13) A und B müßen disjunkt sein.

(13) is ambiguous as there are two possible readings: a non-subordinating (parallel) rhetorical relation to a preceding utterance vs. a subordinating one. This ambiguity is reflected in an underspecified status of a proof step w.r.t. a proof, e.g., new premise vs. conclusion. The necessity modality, too, is compatible with asserting a prerequisite condition or a necessary conclusion.

Finally, ambiguities arise at the level of domain-specific interpretation, where concepts and relations that are unambiguous at the level of linguistic meaning, may have multiple domain-specific interpretations. For example, the lexical concept of CONTAINMENT (expressed, e.g., as *A enthaelt B* or *A muss in B sein*), has the domain-specific interpretation of either subset or element-of relation. This could be resolved in the larger discourse context where A or B are known to be of type set or individual object, respectively. Otherwise, the proof context is needed. Moreover, linguistically unambiguous utterances resulting in underspecified proof-step instantiation are exemplified below:

(14) **T1:** Bitte zeigen Sie: $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$!
S1: nach deMorgan-Regel-2 ist $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cup K(C \cup D))$

This utterance may be a re-writing step applied to the right-hand side of the source equation, or an immediate instantiation of DeMorgan-Regel-2 with $(A \cup B)$ and $(C \cup D)$.

Task 3.1: Resolution of Ambiguities by Proof Step Evaluation

We have seen that the right-hand side of the equation in (12) has two alternative readings: (a) $P(C) \cup (A \cap B)$ and (b) $P(C \cup (A \cap B))$. First, both alternatives are represented as alternative proof branches (OR-branches) in the LU proof language and then translated into the LS proof object maintained in the proof manager. The task is now to rule out some branches in the proof tree and to compute a preferential ranking for the remaining ones. Simple type checking is already sufficient to exclude the former alternative since ‘powerset’ may map the set ‘C’ into a set of higher type; this leads to a type-checking clash for ‘$\cup$’. When we replace the powerset symbol ‘P’ in (12) by the complement symbol ‘K’ we obtain:

(15) K((A∪C)∩(B∪C))=KC ∪(A∩B)

The right-hand side has again two readings: (a) $K(C) \cup (A \cap B)$ and (b) $K(C \cup (A \cap B))$, but type-checking is no longer sufficient to exclude the first reading.

Ambiguity resolution can additionally be supported by proof step evaluation: The first criterion we check for is soundness. In (15), soundness already excludes the alternative proof branch corresponding to (a) since it cannot be proved by a domain reasoner, while (b) can. Generally, however, the soundness check may only reduce the set of alternative interpretaions instead of collapsing it to a single one. We, therefore, employ relevance as an additional criterion, which will ideally yield a preferential ranking for the OR-branches (and thus for the remaining alternative input interpretations).

Example (14) illustrates a situation where two different sound interpretations are possible: this proof step utterance mentions neither to what assertion(s) the deMorgan rule is actually to be applied nor at which subterm-prositions it is used; the student may just mean the *instantiation* of the deMorgan-Rule or he may mean the twofold *application* of the deMorgan-Rule at different positions within the given premise.

As has been mentioned in the report, our approach to resolve underspecification is to further improve and employ an abstract-level agent-oriented proof mediator that heuristically enumerates assertion-level proofs (Vo, 2003, To appear; Vo, Benzmüller, & Autexier, 2003). In our example, this proof mediator is capable of generating two proofs that correspond to the mentioned interpretation alternatives, i.e., from the generated proof alternatives we read off information that instantiates the underspecified data. To this set of generated alternatives we may then apply the relevance-analysis as described above in order to compute a preferential ranking. A soundness check is not needed since the generated proof alternatives are already guaranteed to be sound by construction.

When proofs are described in a dialog (or in a text), preferential ordering over possible proof-alternatives is influenced not only by the proof-structure and other proof-related information, but also by discourse factors. For example, proof steps (or their parts) that are constructed by the domain reasoner, but not mentioned explicitly in discourse, may not be available to anchor subsequent proof steps (i.e., such proof alternatives should be dispreferred); also parallelism, embedding, or sequencing signaled through discourse connectives indicates the intended proof progression (i.e., such proof alternatives should be ranked higher).

Task 3.2 Underspecified descriptions and their processing without enumeration

The general picture about the interaction between linguistic interpretation and domain reasoning arising in WP 3.1 is as follows: Linguistic interpretation finds the set of possible readings, and passes them to the proof manager, making use of meta-variables to represent disjunctions between readings in a packed format. The proof manager unpacks the linguistic interpretation output, and sends the readings one by one to the reasoning module, which checks soundness and relevance. On the basis of the domain reasoner's results, the proof manager filters the set of alternatives, and produces an ordered short list of the remaining ones, on the basis of preference values from different sources.

Since one user utterance may contain several ambiguities, the number of readings can be considerably large, and accordingly the required processing time for the domain reasoner may be far beyond the time constraints for a tutorial dialog setting. This problem has been extensively discussed in underspecification semantics research (cf. quasi-logical form (Alshawi & Crouch, 1992), underspecified DRT (Reyle, 1993; Poessio, 1995), Minimal Recursion Semantics (Copestake, Flickinger, Sag, & Pollard, 1999), hole-semantics (Bos, 2002)). In this work package we will investigate, how the techniques for direct processing and resolution of underspecified representations developed in the CHORUS project (Egg, Koller, & Niehren, 2001) can be applied in the DIALOG project. The basic idea is to reduce the set of readings by evaluating alternatives in local case distinctions of an underspecified representation, without enumerating all disambiguations. CHORUS has in particular developed very fast algorithms to process dominance constraints, which are useful to model scope underspecification.

In the math tutoring domain, there are two important types of ambiguity which relate to scope: Coreferentiality of identifiers (like A and B in examples (6) and (7)), which is indicated in a discourse semantic framework by the scope of the respective discourse referents; and anaphoric elements or null-complement constructions which refer to more or less extended portions of preceding discourse steps, like "daraus folgt" in (1) or "es folgt".

To adopt CHORUS processing techniques for dominance constraints in DIALOG, we first have to extend the scope analysis, which has mainly been done for standard predicate and higher-order logic in CHORUS, to discourse semantics, where scope is not fully determined by dominance, but by the more complex concept of accessibility. We can build on work that has been done in CHORUS already about underspecification in dynamic logic (Koller & Niehren, 2000). Also, we have to adapt semantic interpretation and the interface language LU in a way that they provide underspecified representations in the appropriate format. With respect to the HLDS formalism, (Baldridge & Kruijff, 2002) show how MRS-style underspecification is replicated in the HLDS represenitions of the linguistic meaning of sentences. Our task is to take this approach further to the HLDS representation of discourse proposed in (Kruijff & Kruijff-Korbayová, 2001), which is to be further developed in WP1 of this project.

Finally, we will collaborate with CHORUS on the integration of preference information into the underspecification framework.

WP4: Integration and Evaluation

In the first project phase, our research was inspired and guided by the ambitious but rather general math tutoring scenario as sketched in the report (see also (Benzmüller et al., 2003)). We will retain this scenario as reference framework in the second project phase, but we will focus on specific issues and extend the demonstrator with selected capabilities. We will thereby address a range of research challenges pertaining to the analysis and evaluation of proof step utterances with respect to a dynamically evolving proof context. We will use robust techniques of input processing, handling also multimodal input consisting of written and spoken language, accompanied by mouse pointing. To resolve ambiguities, we will employ proof-step relevance evaluation, taking into account both discourse and domain information.

Our demonstrator implementation will be based on the demonstrator of the first phase. We will continue to use *Rubin* (Fliedner & Bobbert, 2003) as a generic shell for our system. *Rubin* provides a language to implement the core of a ISU-based Dialog Manager (among others, the structure of the Information State (IS) and the IS update rules) and the interfaces between the modules. As an integration framework for the proof manager and the domain reasoning tools we will benefit from the infrastructure (such as the MathWeb system (Zimmer & Kohlhase, 2002)) that has been developed in the OMEGA project.

A second experiment to collect further corpus data is still planned in the current (first) project phase. After this, empirical investigations will be conducted as needed, to support specific research questions pertaining to input analysis and proof-step evaluation.

2.6 Time Plan

All tasks of WP1 and WP2 will start immediately at the project beginning. In WP3, the investigation of the applicability of CHORUS-techniques to linguistic meaning representation may also start at the project beginning. However, the other tasks in WP3 should start with a time offset of approximately one year, because they depend on initial results from WP1 and WP2. As for WP4, partially realized/extended system components should be integrated within the demonstrator as soon as they become available. Experiments supporting specific research questions will be carried out when needed during the project.

2.7 Position within the SFB

The DIALOG project is most closely related to the OMEGA project, which develops a mathematical assistance environment. There is a strong mutual fertilization between the two projects: The DIALOG project provides a novel top-down viewpoint including important empirical data for the development of proof assistants. The OMEGA project, in turn, provides the abstract-level (and logic-level) domain reasoning tools as required for the DIALOG project.

Concerning the Multi-Modal Categorial Grammar formalism with Hybrid Logic Dependency Semantics that we are using for our grammar for deep input analysis, we have cooperated with the Negra project, whose goal was to acquire such grammars for robust semantic-oriented processing from annotated corpora. We will continue this cooperation with the new NiGra project, in particular with respect to analyzing ill-formed and fragmentary input using partial analyses obtained from the deep parser.

As discussed above, we will closely cooperate with CHORUS.

Bibliography

Abel, A., Chang, B.-Y. E., & Pfenning, F. (2001). Human-readable machine-verifiable proofs for teaching constructive logic. In U. Egly, A. Fiedler, H. Horacek, & S. Schmitt (Eds.), *Proceedings of the Workshop on Proof Transformations, Proof Presentations and Complexity of Proofs (PTP'01).* Universitá degli studi di Siena.

Alshawi, H., & Crouch, R. (1992). Monotonic semantic interpretation. In *Proc. of the 30th Meeting of the ACL* (pp. 32–39).

Autexier, S., Benzmüller, C., Fiedler, A., Horacek, H., & Vo, B. Q. (2004). Assertion-level proof representation with under-specification. *Electronic Notes in Theoretical Computer Science.* (In print)

Baldridge, J. M. (2002). *Lexically specified derivational control in combinatory categorial grammar.* Unpublished doctoral dissertation, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh, Edinburgh, UK.

Baldridge, J. M., & Kruijff, G.-J. M. (2002). Coupling CCG with hybrid logic dependency semantics. In M. Diab & P. Resnik (Eds.), *Proceedings of the 40th Annual Meeting of* *the Association for Computational Linguistics (ACL).* Philadelphia PA.

Baldridge, J. M., & Kruijff, G.-J. M. (2003). Multi-modal combinatory categorial grammar. In *Proceedings of the 10th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL'03).* Budapest, Hungary.

Baur, J. (1999). *Syntax und semantik mathematischer texte.* Diplomarbeit, Fachrichtung Computerlinguistik, Universität des Saarlandes, Saarbrücken, Germany.

Benzmüller, C., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayová, I., Pinkal, M., Siekmann, J., Tsovaltzi, D., Vo, B. Q., & Wolska, M. (2003). Tutorial dialogs on mathematical proofs. In P. Baumgartner, P. A. Cairns, M. Kohlhase, & E. Melis (Eds.), *Proceedings of the IJCAI 03 Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems* (pp. 12–22). Acapulco, Mexico.

Bos, J. (2002). *Underspecification and resolution in discourse semantics.* Unpublished doctoral dissertation, Saarland University, Saarbruecken, Germany.

Bos, J. (2003). Implementing the the binding and accomodation theory for anaphora resolution and presupposition projection. *Computational Linguistics*.

Braun, C. (2003). Parsing german text for syntacto-semantic structures. In *Proc. of Lorraine-Saarland Workshop on Prospects and Advances in the Syntex/Semantics Interface* (pp. 99–102). Nancy, France.

Copestake, A., Flickinger, D., Sag, I., & Pollard, C. (1999). *Minimal recursion semantics.* Stanford University.

Dubey, A., & Frank, K. (2003). Parsing german with sister-head dependencies. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL).*

Egg, M., Koller, A., & Niehren, J. (2001). The constraint language for lambda structures. *Journal of Logic, Language, and Information, 10,* 457–485.

Erk, K., Kowalski, A., & Pinkal, M. (2003). A corpus resource for lexical semantics. In *Proc. of the 5th International Workshop on Computational Semantics.* Tilburg, The Netherlands.

Fliedner, G., & Bobbert, D. (2003). A framework for information-state based dialogue (demo abstract). In *Proceedings of the 7th workshop on the semantics and pragmatics of dialogue DiaBruck.* Saarbrücken.

Gentzen, G. (1935). Untersuchungen über das logische Schließen I & II. *Mathematische Zeitschrift, 39,* 176–210, 572–595.

Hübner, M., Autexier, S., Benzmüller, C., & Meier, A. (2004). Interactive theorem proving with tasks. *Electronic Notes in Theoretical Computer Science.* (To appear)

Koller, A., & Niehren, J. (2000). On underspecified processing of dynamic semantics. In *Proceedings of COLING-2000.* Saarbrücken.

Kruijff, G.-J., & Kruijff-Korbayová, I. (2001). A hybrid logis fomralization of information structure sensitive discourse interpretation. In V. Matoušek, P. Mautner, R. Mouček, & K. Taušer (Eds.), *Proceedings of the 4th International Conference on Text, Speech and Dialogue (TSD'2001), Železná Ruda, Czech Republic* (pp. 31–38). Springer.

Kruijff, G.-J. M. (2001). *A categorial-modal logical architecture of informativity: Dependency grammar logic & information structure.* Ph.d. dissertation, Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

Poessio, M. (1995). Disambiguation as (defeasible) reasoning about underspecified representations. In *Proc. of the 10th Amsterdam Colloquium.* Universiteit van Amsterdam, ILLC.

Reyle, U. (1993). Dealing with ambiguities by underspecification: Construction, representation, and deduction. *Journal of Semantics*, *10*(2), 123–179.

Skut, W., & Brants, T. (1998). A maximum-entropy partial parser for unrestricted text. In *Proc. of 6th Workshop on Very Large Corpora.* Montreal, Canada.

Vo, B. Q., Benzmüller, C., & Autexier, S. (2003). Assertion application in theorem proving and proof planning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).* Acapulco, Mexico. (poster description))

Vo, Q. B. (2003). A task-oriented agent-based mechanism for theorem proving. In J. Liu, B. Faltings, M. Klusch, & C. Liu (Eds.), *The 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003)* (p. 275-281). IEEE Computer Society/WIC.

Vo, Q. B. (To appear). A task-oriented information mediator for mathematical assistant systems. *International Journal of Web Intelligence and Agent Systems*.

Zimmer, J., & Kohlhase, M. (2002). System description: The MathWeb Software Bus for distributed mathematical reasoning. In A. Voronkov (Ed.), *Proceedings of the 18th International Conference on Automated Deduction (CADE-18)* (Vol. 2392). Springer.

Zinn, C. (1999). Understanding mathematical discourse. In *Proc. of Amstelogue'99.* Amsterdam.

Zinn, C. (2003). Computational framework for understanding mathematical discourse. *Logic Journal of the IGPL*, *11*(4), 457–484.

3.8 External Funding for the Project MI 3

PC:	Personnel requirements and costs (see 3.8.1 for explanation)
AC:	Administrative costs (see 3.8.2 for explanation)
I:	Investments (equipment exceeding 10K EUR; see 3.8.3 for explanation)

	Grant 2004			**2005**			**2006**			**2007**		
PC	Salary Group	No.	Amount EUR	Salary Group	No.	Amount EUR	Salary Group	No.	Amount EUR	Salary Group	No.	Amount EUR
	Bat IIa	2	117.600	Bat IIa	2	117.600	Bat IIa	2	117.600	Bat IIa	2	117.600
	SHK	1	12.000	SHK	1	12.000	SHK	1	12.000	SHK	1	12.000
	WHK	1	18.000	WHK	1	18.000	WHK	1	18.000	WHK	1	18.000
	Total:	4	147.600	Total:	4	147.600	Total:	4	147.600	Total:	4	147.600
AC				Expense category or number		Amount EUR	Expense category or number		Amount EUR	Expense category or number		Amount EUR
				Total:		0	Total:		0	Total:		0
I				Total investments			Total investments			Total investments		
				0			0			0		

3.8.1 Explanation of Personnel Requirements

	Name, degree, position	Specific field of the employee	Institute of the university or other organization	Contribution to the project in hours/week (consulting: C)	In this position in the CRC since	Proposed BAT level
University funding						
3.8.1.1 Scientific personnel (including assistants)	1. Manfred Pinkal, Prof. Dr., Professor	CL	Uni., FR Allg. Linguistik	4	1.1.96	–
	2. Jörg Siekmann, Prof. Dr., Professor	CS	Uni., FR Informatik	4	1.1.96	–
	3. Christoph Benzmüller, Dr., Hochschuldoz.	CS	Uni., FR Informatik	10	1.1.01	–
	4. Ivana Kruijff-Korbayová, Dr., Hochschulass.	CL	Uni., FR Allg. Linguistik	10	1.1.01	–
	5. Alexander Koller, Dipl.Inf. Dipl.Ling.	CL	Uni., FR Allg. Linguistik	2		–
	6. Serge Autexier, Dr.	CS	DFKI GmbH	2		–
	7. Erica Melis, Dr., PD	CS	DFKI GmbH	2	1.1.98	–
3.8.1.2 Nonscientific personnel	6. Irmtraud Stein, Verwaltungsangestellte	–	Uni., FR Informatik	2	1.1.96	–
	7. Helga Riedel, Verwaltungsangestellte	–	Uni., FR Allg. Linguistik	2	1.1.96	–
External funding						
3.8.1.3 Scientific personnel (including assistants)	8. Armin Fiedler	CS	Uni., FR Informatik	38,5	1.1.96	BAT IIa
	9. Magdalena Wolska	CL	Uni., FR Allg. Linguistik	38,5	1.1.03	BAT IIa
	10. Dimitra Tsovaltzi/Mark Buckley	CL	Uni., FR Informatik	19	1.4.04	WHK
	11. N.N.	CL	Uni., FR Allg. Linguistik	19		SHK
3.8.1.4 Nonscientific personnel						

(Positions for which funding is being applied for *for the first time* are marked with X.)

Job Descriptions of University Personnel

1. Manfred Pinkal: general project leader; linguistic and dialogue aspects of the project
2. Jörg Siekmann: general project leader; mathematical domain reasoning aspects of the project
3. Christoph Benzmüller: general project leader responsible for mathematical domain reasoning (WP2) and for CS aspects of WP3 and WP4.
4. Ivana Kruijff-Korbayová: general project leader responsible for natural language analysis (WP1) and for CL aspects of WP3 and WP4.
5. Alexander Koller: collaborating researcher, will support the project wrt. the adaptation and use of CHORUS tools (WP3).
6. Serge Autexier: collaborating researcher, will support the project in mathematical domain reasoning aspects.
7. Erica Melis: collaborating researcher, focus on tutoring in mathematics and abstract-level mathematical domain reasoning.
8. Irmtraud Stein: Secretary
9. Helga Riedel: Secretary

Job Descriptions of Externally Funded Personnel

5. Magdalena Wolska will work on WP1 and the CL aspects of WP3 and WP4.
6. Armin Fiedler will work on WP2 and the CL aspects of WP3 and WP4.
7. Dimitra Tsovaltzi/Mark Buckley will work on the integration of tutoring aspects and on dialogue modeling
8. N.N. will support the implementation and empirical studies.

Justification of requests for WHK: In Computer Science is has turned out to be very hard to attract skilled students; fortunately Dimitra Tsovaltzi (PhD student) is currently contributing to the project as WHK at the CS side and soon Mark Buckley will join the project. Mark Buckley will finish his masters degree until the end of this year. Without WHK support we cannot keep this skilled students in the project.

2.8.2 Specification and Explanation of Administrative Costs (by Fiscal Year)

	2005	2006	2007
For administrative expenses, the following amounts are expected to be available from the *university's own funds*:	1000	1000	1000
For administrative expenses, the following amounts are requested as *external funding* (corresponding with the total amounts under "administrative expenses" in the overview 3.8):			

(All amounts in EUR.)

Explanation of the Requested *External Funding* for Administrative Expenses

2.8.3 Investments (Equipment With Gross Cost Above 10.000 EUR and Vehicles)

	Requested for the fiscal year		
	2005	2006	2007
Amount:			

(All prices in EUR *including* VAT, transportation costs, etc.)

Explanation of the Requested External Funding for Investments