Lexical-Semantic Interpretation of Language Input in Mathematical Dialogs

Magdalena Wolska[1] Ivana Kruijff-Korbayová[1] Helmut Horacek[2]

[1]Fachrichtung Computerlinguistik [2]Fachrichtung Informatik
Universität des Saarlandes, Postfach 15 11 50
66041 Saarbrücken, Germany
{magda,korbay}@coli.uni-sb.de, horacek@ags.uni-sb.de

Abstract

Discourse in formal domains, such as mathematics, is characterized by a mixture of telegraphic natural language and embedded (semi-)formal symbolic mathematical expressions. Due to the lack of empirical data, little is known about the suitability of input analysis methods for mathematical discourse in a dialog setting. We present an input understanding method for a tutoring system teaching mathematical theorem proving. The adopted deep analysis strategy is motivated by the complexity of the language phenomena observed in a corpus collected in a *Wizard-of-Oz* experiment. Our goal is a uniform input interpretation, in particular, considering different degrees of formality of natural language verbalizations.

1 Introduction

In the DIALOG[1] project (Benzm ̈uller et al., 2003a), we are investigating and modeling semantic and pragmatic phenomena in tutorial dialogs focused on problem solving skills in mathematics. Our goal is (i) to empirically investigate the use of flexible natural language dialog in tutoring mathematics, and (ii) to develop a dialog-based tutoring system for teaching mathematical theorem proving. The experimental system will engage in a dialog in written natural language to help a student understand and construct mathematical proofs. In this paper, we address a strategy for user input interpretation in our setting.

Because of the lack of empirical dialog-data on the use of natural language in formal domains, such as mathematics, we conducted a *Wizard-of-Oz* experiment to collect a corpus of dialogs with a simulated system teaching proofs in naive set theory. An investigation of the corpus reveals language phenomena that present challenges to the existing commonly used input understanding methods. The challenges lie in (i) the tight interleaving of natural and symbolic language, (ii) varying degree of natural language verbalization of the formal mathematical content, and (iii) informal and/or imprecise reference to mathematical concepts and relations.

These phenomena motivate the use of deep syntactic and semantic analysis of user input. We developed a grammar that allows a uniform treatment of the linguistic content on a par with the mathematical content and thus supports analysis of inputs of different degrees of verbalization. We employ a domain-motivated semantic lexicon to mediate between the domain-independent semantic representation obtained through semantic construction during parsing and domain-specific interpretation. This serves to achieve a consistent semantic analysis while avoiding example-based heuristics.

The paper is organized as follows: In Sect. 2, we present the setup of the system and the corpus collection experiment. In Sect. 3 and show examples of language phenomena from our dialogs. In Sect. 4, we first summarize the approach to parsing the mixed symbolic and natural language input and then present a lexical-semantic interface to a domain-specific interpretation of the input. We show example analyzes in Sect. 5. In Sect. 6, we summarize relevant existing approaches to input analysis in (tutorial) dialog systems on the one hand and analysis of mathematical discourse on the other. Sect. 7 is the conclusion.

2 System setup and corpus collection

Our system scenario is illustrated in Fig. 1:

- *Learning Environment*: Students take an interactive course in the relevant subfield of mathematics.
- *Mathematical Proof Assistant (MPA)*: Checks the appropriateness of user specified inference steps with respect to the problem-solving goal; based on ΩMEGA.
- *Proof Manager (PM)*: In the course of tutoring session the student may explore alternative

[1] The DIALOG project is a collaboration between the Computer Science and Computational Linguistics departments of University of the Saarland, and is a part of the Collaborative Research Center on *Resource-Adaptive Cognitive Processes*, SFB 378 (`www.coli.uni-sb.de/sfb378`).

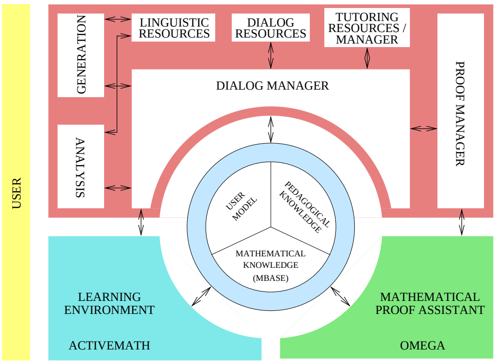


Figure 1: DIALOG project scenario.

proofs. The PM builds and maintains a representation of constructed proofs and communicates with the *MPA* to evaluate the appropriateness of the student's contributions for the proof construction.

- *Dialog Manager*: We employ the Information-State (IS) Update approach developed in the TRINDI project[2]
- *Tutorial Manager (TM)*: This component incorporates extensions to handle tutorial-specific dialog moves, such as hinting.
- *Knowledge Resources*: This includes *pedagogical knowledge* (teaching strategies), and *mathematical knowledge*.

In order to empirically investigate the use of natural language in mathematics tutoring, we collected and analyzed a corpus of dialogs with a simulated tutoring system.

24 subjects with varying educational background and little/fair prior mathematical knowledge participated in a *Wizard-of-Oz* experiment (Benzm¨uller et al., 2003b). The experiment consisted of 3 phases: (i) *preparation and pre-test* (on paper), (ii) *tutoring session* (mediated by a *WOz* tool (Fiedler and Gabsdil, 2002)), (iii) *post-test and evaluation questionnaire* (on paper). At the tutoring session, they were asked to prove 3 theorems[3]: (i) $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$; (ii) $A \cap B \in P((A \cup C) \cap (B \cup C))$; (iii) *If* $A \subseteq K(B)$, *then* $B \subseteq K(A)$. The subjects were instructed to enter proof steps, rather than complete proofs at once, to encourage interaction with the system. The subjects and the tutor were free in formulating their turns.[4]

The collected corpus consists of 66 dialog log-files, containing on average 12 turns. The total number of sentences is 1115, of which 393 are student sentences. The students' turns consisted on average of 1 sentence, the tutor's of 2. More details on the corpus itself and annotation efforts that guide the development of the system components can be found in (Wolska *et al.*, 2004).

3 Linguistic data

In this section, we present an overview of the language phenomena prominent in the collected dialogs to indicate the overall complexity of input understanding in our setting.[5]

Interleaved natural language and formulas The following examples illustrate how the mathematical language, often semi-formal, is interleaved with the natural language informally verbalizing proof steps.

A auch $\subseteq$ $K(B)$	[$A also \subseteq K(B)$]
A∩B **ist** ∈ **von** C∪(A∩B)	[... *is* ∈ *of* ...]
(da ja A∩B=∅**)**	[*(because A∩B=∅)*]
B enthaelt **kein x**∈A	[*B contains no x∈A*]

The mixture affects the way parsing needs to be conducted: mathematical content has to be identified before it is interpreted within the utterance. In particular, mathematical objects (or parts thereof) may lie within the scope of quantifiers or negation expressed in natural language (as in the last example above).

Imprecise or informal naming Domain relations and concepts are described informally using imprecise and/or ambiguous expressions.

A **enthaelt** B	[*A contains B*]
A muss **in** B sein	[*A must be in B*]

B **vollstaendig ausserhalb von** A **liegen** muss, also im Komplement von A
[*B has to be entirely outside of A, so in the complement of A*]

dann sind A und B **(vollkommen) verschieden**, haben keine **gemeinsamen Elemente**
[*then A and B are (completely) different, have no common elements*]

In the above examples, **contain** and **be_in** can express domain relations of (strict) subset or element, while **be_outside_of** and **be_different** are informal descriptions of the empty intersection of sets.

To handle imprecision and informality, we have designed an ontological knowledge base that includes domain-specific interpretations of conceptual relations that have corresponding formal counterparts in the domain of naive set theory.

[2] http://www.ling.gu.se/research/projects/trindi/

[3] K stands for set complement and P for power set.

[4] Buttons were available in the interface for inserting mathematical symbols, while literals were typed on the keyboard.

The dialogs were typed in German.

[5] As the tutor was also free in wording his turns, we include observations from both student and tutor language behavior.

Metonymy Metonymic expressions are used to refer to structural sub-parts of formulas, resulting in predicate structures acceptable informally, yet incompatible in terms of selection restrictions.

> 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
> [*Then for the left hand side it is valid that..., the term A ∩ B is already there, and so an element of it*]

where the predicate **be_valid_for**, in this domain, normally takes an argument of sort CONSTANT, TERM or FORMULA, rather than LOCATION;

> de morgan regel 2 auf **beide komplemente angewendet**
> [*de morgan rule 2 applied to both complements*]

where the predicate **apply** takes two arguments: one of sort RULE and the other of sort TERM or FORMULA, rather than OPERATION ON SETS.

Informal descriptions of proof-step actions

> **Wende** zweimal die DeMorgan-Regel an
> [*I'm applying DeMorgan rule twice*]
> damit kann ich den oberen Ausdruck wie folgt **schreiben**:...
> [*given this I can* **write** *the upper term as follows:*...]

Sometimes, "actions" involving terms, formulae or parts thereof are verbalized before the appropriate formal operation is performed. The meaning of the "action verbs" is needed for the interpretation of the intended proof-step.

Discourse deixis

der obere Ausdruck	[*the above term*]
der letzte Satz	[*the last sentence*]
Folgerung aus **dem Obigen**	[*conclusion from the above*]
aus der regel in **der zweiten Zeile**	[*from the rule in the second line*]

This class of referring expressions includes also references to structural parts of terms and formulas such as "the left side" or "the inner parenthesis" which are incomplete specifications: the former refers to a part of a formula, the latter, metonymic, to an expression enclosed in parenthesis. Moreover, they require discourse referents for sub-parts of mathematical expressions to be available.

Generic vs. specific reference

> **Potenzmenge** enthaelt alle Teilmengen, also auch **(A∩B)**
> [*A power set contains all subsets, hence also(A∩B)*]

Generic and specific references can appear within one utterance as above, where "a power set" is a generic reference, whereas "$A \cap B$" is a specific reference to a subset of a specific instance of a power set introduced earlier.

Co-reference[6]

> Da, wenn $A_i \subseteq K(B_j)$ sein soll, A_i Element von $K(B_j)$ sein muss. Und wenn $B_k \subseteq K(A_l)$ sein soll, muss es_k auch Element von $K(A_l)$ sein.
> [*Because if it should be that $A_i \subseteq K(B_j)$, A_i must be an element of $K(B_j)$. And if it should be that $B_k \subseteq K(A_l)$, it must be an element of $K(A_l)$ as well.*]

> DeMorgan-Regel-2 besagt: $K(A_i \cap B_j)$ = $K(A_i) \cup K(B_j)$
> In diesem Fall: z.B. $K(A_i)$ = dem Begriff
> $K(A_k \cup B_l)$ $K(B_j)$ = dem Begriff $K(C \cup D)$
> [*DeMorgan-Regel-2 means:*
> $K(A_i \cap B_j) = K(A_i) \cup K(B_j)$ *In this case: e.g.* $K(A_i)$ = *the term* $K(A_k \cup B_l)$ $K(B_j)$ = *the term* $K(C \cup D)$]

Co-reference phenomena specific to informal mathematical discourse involve (parts of) mathematical expressions within text. In particular, entities denoted with the same literals may not co-refer, as in the second utterance.

In the next section, we present the input interpretation procedure up to the level of lexical-semantic interpretation. We concentrate on the interface between the linguistic meaning representation (obtained from the parser) and the representation of domain-knowledge (encoded in a domain ontology), which we realize through a domain-motivated semantic lexicon.

4 Interpretation strategy

The task of the input interpretation component is two-fold. Firstly, it is to construct a representation of the utterance's linguistic meaning. Secondly, it is to identify within the utterance, separate, and construct interpretations of:

(i) parts which constitute meta-communication with the tutor (e.g., "Ich habe die Aufgabenstellung nicht verstanden." [*I don't understand what the task is.*] that are not to be processed by the domain reasoner; and

(ii) parts which convey domain knowledge that should be verified by a domain reasoner; for example, the entire utterance "$K((A \cup B))$ ist laut deMorgan-1 $K(A) \cap K(B)$" [... *is, according to deMorgan-1,...*] can be evaluated in the context of the proof being constructed; on the other hand, the reasoner's knowledge base does not contain appropriate representations to evaluate the appropriateness of the focusing particle "also" in "Wenn A = B, dann ist A auch $\subseteq K(B)$ und B $\subseteq K(A)$." [*If A = B, then A is also* $\subseteq K(B)$ *and B* $\subseteq K(A)$.].

Domain-specific interpretation(s) of the proof-relevant parts of the input are further processed by

[6] To indicate co-referential entities, we inserted the indices which are not present in the dialog logfiles.

Proof Manager, a component that directly communicates with a domain-reasoner[7]. The task of the Proof Manager is to: (i) build and maintain a representation of the proof constructed by the student;[8] (ii) check appropriateness of the interpretation(s) found by the input understanding module with the state of the proof constructed so far; (iii) given the current proof state, evaluate the utterance with respect to soundness, relevance, and completeness.

The semantic analysis proceeds in 2 stages:

(i) After standard pre-processing[9], mathematical expressions are identified, analyzed, categorized, and substituted with default lexicon entries encoded in the grammar. The input is then syntactically parsed, and an formal abstract representation of its meaning is constructed compositionally along with the parse;

(ii) The obtained meaning representation is subsequently merged with discourse context and interpreted by consulting a semantic lexicon of the domain and a domain-specific ontology.

In the next sections, we first briefly summarize the syntactic and semantic parsing part of the input understanding process[10] and show the format of meaning encoding constructed at this stage (Sect. 4.1). Then, we show the lexical-semantic interface to the domain ontology (Sect. 4.2).

4.1 Linguistic Meaning

By linguistic meaning (LM), we understand the dependency-based deep semantics in the sense of the Prague School sentence meaning as employed in the Functional Generative Description (FGD) (Sgall et al., 1986; Kruijff, 2001). It represents the literal meaning of the utterance rather than a domain-specific interpretation.[11] In FGD, the central frame unit of a sentence/clause is the head verb which specifies the *tectogrammatical relations* (TRs) of its dependents (*participants*). Further distinction is drawn into *inner participants*, such as Actor, Patient, Addressee, and *free modifications*, such as Location, Means, Direction. Using TRs rather than surface grammatical roles provides a generalized view of the correlations between the conceptual content of an utterance and its linguistic realization.

At the pre-processing stage, mathematical expressions embedded within input are identified, verified as to syntactic validity, categorized, and substituted with default lexical entries encoded in the parser grammar for mathematical expression categories. For example, the expression $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cap K(C \cup D))$ given its top node operator, =, is of type formula, its “left side” is the expression $K((A \cup B) \cap (C \cup D))$, the list of bracketed sub-expressions includes: A∪B, C∪D, $(A \cup B) \cap (C \cup D)$, etc.

Next, the pre-processed input is parsed with a lexically-based syntactic/semantic parser built on Multi-Modal Combinatory Categorial Grammar (Baldridge, 2002; Baldridge and Kruijff, 2003). The task of the deep parser is to produce an FGD-based linguistic meaning representation of syntactically well-formed sentences and fragments. The linguistic meaning is represented in the formalism of Hybrid Logic Dependency Semantics. Details on the semantic construction in this formalism can be found in (Baldridge and Kruijff, 2002).

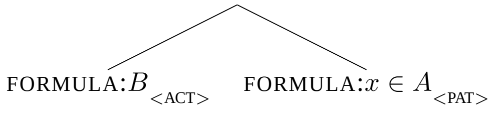
To derive our set of TRs we generalize and simplify the collection of Praguian tectogrammatical relations from (Hajičová et al., 2000). One reason for simplification is to distinguish which relations are to be understood metaphorically given the domain-specific sub-language. The most commonly occurring relations in our context (aside from the roles of Actor and Patient) are Cause, Condition, and Result-Conclusion (which coincide with the rhetorical relations in the argumentative structure of the proof):

Da [A $\subseteq K(B)$ gilt]$_{\text{<CAUSE>}}$, alle x, die in A sind sind nicht in B
[*As A$\subseteq K(B)$ applies, all x that are in A are not in B*]

Wenn [A $\subseteq K(B)$]$_{\text{<COND>}}$, dann A ∩ B=∅
[*If A$\subseteq K(B)$, then A∩B=∅*]

For example, in one of the readings of “B enthaelt $x \in A$”, the verb “enthaelten” represents

Figure 2: TRs in “B contains $x \in A$”.



[7] We are using a version of ΩMEGA adapted for assertion-level proving (Vo et al., 2003)

[8] The discourse content representation is separated from the proof representation, however, the corresponding entities must be co-indexed in both.

[9] Standard pre-processing includes sentence and word tokenization, (spelling correction and) morphological analysis, part-of-speech tagging.

[10] We are concentrating on syntactically well-formed utterances. In this paper, we are not discussing ways of combining deep and shallow processing techniques for handling malformed input.

[11] LM is conceptually related to logical form, however, differs in coverage: while it does operate on the level of deep semantic roles, such aspects of meaning as the scope of quantifiers or interpretation of plurals, synonymy, or ambiguity are not resolved.

the meaning **contain** and in this frame takes dependents in the relations Actor and Patient, shown schematically in Fig. 2 (FORMULA represents the default lexical entry for the identified mathematical expressions categorized as formulas). The linguistic meaning of this utterance returned by the parser obtains the following representation:

@h1(contain ∧ <ACT>(f1 ∧ FORMULA:B) ∧ <PAT>(f2 ∧ FORMULA: $x \in A$)

where h1 is the state where the proposition **contain** is true, and the nominals f1 and f2 represent dependents of the head **contain**, in the relations Actor and Patient, respectively.

More details on our approach to parsing interleaved natural and symbolic expressions can be found in (Wolska and Kruijff-Korbayová, 2004a) and more information on investigation into tectogrammatical relations that build up linguistic meaning of informal mathematical text can be found in (Wolska and Kruijff-Korbayová, 2004b).

4.2 Conceptual Semantics

At the final stage of input understanding, the linguistic meaning representations obtained from the parser are interpreted with respect to the given domain. We encode information on the domain-specific concepts and relations in a domain ontology that reflects the knowledge base of the domain-reasoner, and which is augmented to allow resolution of ambiguities introduced by natural language (Horacek and Wolska, 2004). We interface to the domain ontology through an upper-level ontology of concepts at the lexical-semantics level.

Domain specializations of conceptual relations are encoded in the domain ontology, while a semantic lexicon assigns conceptually-oriented semantics in terms of linguistic meaning frames and provides a link to the domain interpretation(s) through the domain ontology. Lexical semantics in combination with the knowledge encoded in the ontology allows us to identify those parts of utterances that have an interpretation in the given domain. Moreover, productive rules for treatment of metonymic expressions are encoded through instantiation of type compatible concepts. If more than one lexical-semantic interpretation is plausible, no disambiguation is performed. Alternative conceptual representations are further interpreted using the domain ontology, and passed on to the Proof Manager for evaluation. Below we explain some of the entries the semantic lexicon encodes:

Containment The Containment relation specializes into the domain relations of (strict) SUBSET and ELEMENT. Linguistically, it can be realized, among others, with the verb “enthalten” (“contain”). The tectogrammatical frame of “enthalten” involves the roles of *Actor* (ACT) and *Patient* (PAT):

contain($ACT_{type:FORMULA}$, $PAT_{type:FORMULA}$) $\equiv$ (SUBFORMULA$_{PAT}$, embedding$_{ACT}$)
contain($ACT_{type:OBJECT}$, $PAT_{type:OBJECT}$) $\equiv$ CONTAINMENT(container$_{ACT}$, containee$_{PAT}$)

Location The Location relation, realized linguistically by the prepositional phrase introduced by “in”, involves the tectogrammatical relations *HasProperty-Location* (LOC) and the *Actor* of the predicate “sein”. We consider *Location* in our domain as synonymous with *Containment*. Another realization of this relation, dual to the above, occurs with the adverbial phrase “außerhalb von ...(liegen)” (“lie outside of”) and is defined as negation of *Containment*:

in($ACT_{type:OBJECT}$,$LOC_{type:OBJECT}$)
$\equiv$ CONTAINMENT(container$_{LOC}$, containee$_{ACT}$)
outside($ACT_{type:OBJECT}$,$LOC_{type:OBJECT}$)
$\equiv$ not(in($ACT_{type:OBJECT}$,$LOC_{type:OBJECT}$))

Common property A general notion of “common property” we define as follows:

common(Property, $ACT_{plural(A:SET,B:SET)}$)
$\equiv Property(p1, A) \wedge Property(p1, B)$

Property is a meta-object that can be instantiated with any relational predicate, for example as in “(A und B)$_{<ACT>}$ haben (gemeinsame Elemente)$_{<PAT>}$” (“A and B have common elements”):

common(ELEMENT, $ACT_{plural(A:SET,B:SET)}$)
$\equiv$ ELEMENT(p_1,A) $\wedge$ ELEMENT(p_1, B)

Difference The Difference relation, realized linguistically by the predicates “verschieden (sein)” (“be different”; for COLLECTION or STRUCTURED OBJECTS) and “disjunkt (sein)” (“be disjoint”; for objects of type COLLECTION) involves a plural *Actor* (e.g. coordinated noun phrases) and a *HasProperty* TRs. Depending on the type of the entity in the Actor relation, the interpretations are:

different($ACT_{plural(A:SET,B:SET)}$) $\equiv A \neq B$
different($ACT_{plural(A:SET,B:SET)}$)
$\equiv (e_1$ ELEMENT $A \wedge e_2$ ELEMENT $B \Rightarrow e_1 \neq e_2)$
different($ACT_{plural(A:STRUCTUREDOBJECT,B:STRUCTUREDOBJECT)}$)
$\equiv (Property_1(p1, A) \wedge Property_2(p2, B) \wedge Property_1 = Property_2 \Rightarrow p1 \neq p2)$

Mereological relations Here, we encode part-of relations between domain objects. These concern both physical surface and ontological properties of objects. Commonly occurring part-of relations in our domain are:

hasComponent(STRUCTURED OBJECT$_{term,formula}$,
STRUCTURED OBJECT$_{SUBTERM,SUBFORMULA}$)
hasComponent(STRUCTURED OBJECT$_{term,formula}$,
STRUCTURED
OBJECT$_{ENCLOSEDTERM,ENCLOSEDFORMULA}$)
hasComponent(STRUCTURED OBJECT$_{term,formula}$,
STRUCTURED
OBJECT$_{TERMCOMPONENT,FORMULACOMPONENT}$)

Moreover, from the ontology we have:

Property(STRUCTURED OBJECT$_{term,formula}$,
component$_{term-side,formula-side}$)

Using these definitions and polysemy rules such as polysemous(Object, Property), we can obtain interpretation of utterances such as “Dann gilt f¨ur die linke Seite, ...” (“Then for the left side it holds that ...”) where the predicate “gilt” normally takes two arguments of types STRUCTURED OBJECT$_{term,formula}$, rather than an argument of type Property.

For example, the previously mentioned predicate **contain** (Fig. 2) represents the semantic relation of *Containment* which, in the domain of naive set theory, is ambiguous between the domain relations ELEMENT, SUBSET, and PROPER SUBSET. The alternative specializations are encoded in the domain ontology, while the semantic lexicon provides the conceptual structure of the head predicate. At the domain interpretation stage, the semantic lexicon is consulted to translate the tectogrammatical frame of the predicate into a semantic relation represented in the domain ontology. For the predicate **contain**, from the semantic lexicon, we obtain:

contain($ACT_{type:FORMULA}$, $PAT_{type:FORMULA}$)
$\equiv$ (SUBFORMULA$_{PAT}$, embedding$_{ACT}$)
[*’a Patient of type* FORMULA *is a* **subformula** *embedded within a* FORMULA *in the Actor relation with respect to the head* **contain***’*]
contain($ACT_{type:OBJECT}$, $PAT_{type:OBJECT}$)
$\equiv$ CONTAINMENT(container$_{ACT}$, containee$_{PAT}$)
[*’the* **Containment** *relation involves a predicate* **contain** *and its Actor and Patient dependents, where the Actor and Patient are the* **container** *and* **containee** *parameters respectively’*]

Translation rules that consult the domain ontology expand the conceptual structure representation into alternative domain-specific interpretations preserving argument structure. As it is in the capacity of neither sentence-level nor discourse-level analysis to evaluate the appropriateness of the alternative interpretations in the proof context, this task is delegated to the Proof Manager.

5 Example analysis

In this section, we illustrate the mechanics of the approach on the following example:

A enthaelt keinesfalls Elemente, die auch in B sind.
[*A contains no elements that are also in B*]

The analysis proceeds as follows.

The mathematical expression tagger first identifies the expressions A and B. If there was no prior discourse entity for “A” and “B” to verify their types, they are ambiguous between *constant*, *term*, and *formula*[12]. The expressions are substituted with generic entries FORMULA, TERM, CONST represented in the parser grammar. The sentence is assigned alternative readings: “CONST contains no elements that are also in CONST”, “CONST contains no elements that are also in TERM”, “CONST contains no elements that are also in FORMULA”, etc. Here, we continue only with “CONST contains no elements that are also in CONST”; the other readings would be discarded at later stages of processing because of sortal incompatibilities.

The linguistic meaning of the utterance obtained from the parser is represented by the following formula[13]:

@n1(no ∧ <Restr>e1 ∧
<Body>**(p1 ∧ contain ∧ <ACT>(a1 ∧ A) ∧ <PAT> e1))** ∧
@**e1(element** ∧
<**GenRel**>**(b1 ∧ be ∧** <**ACT**>**e1 ∧** <**HasProp-Loc**>**(b2 ∧ B)))**
[‘*(set) A contains no elements that are in (set) B’*]

Next, the semantic lexicon is consulted to translate the linguistic meaning representation into a conceptual structure. The relevant lexical semantic entries are *Containment* and *Location* (see Sect. 4.2). The transformation is presented schematically below:

contain($ACT_{OBJECT:A}$, $PAT_{OBJECT:element}$) $\equiv$
CONTAINMENT(container$_{A}$, containee$_{element}$)
($ACT_{OBJECT:element}$, **HasProp-Loc**$_{OBJECT:B}$)
$\equiv$ CONTAINMENT(container$_{B}$, containee$_{element}$)

Finally, in the domain ontology, we find that the conceptual relation of *Containment*, in naive set theory, specializes into the domain relations of ELEMENT, SUBSET, STRICT SUBSET. Using the linguistic meaning, the semantic lexicon, and the domain ontology, we obtain all the combinations of interpretations, including the target one paraphrased below:

’it is not the case that there exist elements e, such that e $\in$ A and e $\in$ B’,

Using translation rules the final interpretations are translated into first-order logic formulas and passed on for evaluation to the Proof Manager.

6 Related work

Language understanding in dialog systems, be it with speech or text interface, is commonly performed using shallow syntactic analysis combined

[12] In prior discourse, there may have been an assignment A := ϕ, where ϕ is a formula, in which case, A would be known from discourse context to be of type FORMULA (similarly for term assignment); by CONST we mean a set or element variable such as A, x denoting a set A or an element x respectively.

[13] Irrelevant parts of the meaning representation are omitted; glosses of the formula are provided.

with keyword spotting. Tutorial systems also successfully employ statistical methods which compare student responses to a model built from preconstructed gold-standard answers (Graesser et al., 2000). This is impossible for our dialogs, due to the presence of symbolic mathematical expressions and because of such aspects of discourse meaning as causal relations, modality, negation, or scope of quantifiers which are of crucial importance in our setting, but of which shallow techniques remain oblivious (or handle them in a rudimentary way). When precise understanding is needed, tutorial systems use closed-questions to elicit short answers of little syntactic variation (Glass, 2001) or restricted format of input is allowed. However, this conflicts with the preference for flexible dialog do achieve active learning (Moore, 1993).

With regard to interpreting mathematical texts, (Zinn, 1999) and (Baur, 1999) present DRT analyzes of course-book proofs. The language in our dialogs is more informal: natural language and symbolic mathematical expressions are mixed more freely, there is a higher degree and more variety of verbalization, and mathematical objects are not properly introduced. Both above approaches rely on typesetting information that identifies mathematical symbols, formulas, and proof steps, whereas our input does not contain any such information. Forcing the user to delimit formulas would not guarantee a clean separation of the natural language and the non-linguistic content, while might reduce the flexibility of the system by making the interface harder to use.

7 Conclusion and Further Work

In this paper, we reported on the use of deep syntactic and semantic analysis in the interpretation of mathematical discourse in a dialog setting. We presented an approach that uses domain-motivated semantic lexicon to mediate between a domain-independent representation of linguistic meaning of utterances and their domain-specific interpretation.

We are incrementally extending the coverage of the deep analysis components. Our current parser grammar and upper-level ontology cover most of the constructions and concepts that occur most frequently in our corpus. The module will be evaluated as part of the next Wizard-of-Oz experiment.

We are planning to investigate the possibility of using FrameNet resources developed within the SALSA project (Erk et al., 2003) at the intermediate interpretation stage between the linguistic meaning and domain-specific interpretation. Presently, the semantic lexicon we have constructed encodes, for instance, a general conceptual relation of CONTAINMENT evoked by the verb "enthalten" ("contain"), with dependents in relations Actor and Patient, which corresponds to the FrameNet CONTAINING domain with frame elements CONTAINER and CONTENTS. In the course of further work, we would like to investigate ways of establishing interface between the linguistic meaning TRs and frame elements, and attempt to use FrameNet to interpret predicates unknown to our semantic lexicon. Taking a hypothetical example, if our parser grammar encoded the meaning of the verb "beinhalten" (with the intended meaning **contain**) in the same linguistic meaning frame as "enthalten" (**contain**), while the sense of "beinhalten" were not explicitly defined in the semantic lexicon, we could attempt to interpret it using the FrameNet CONTAINING domain and the existing lexical semantic entry for "enthalten".

References

J. Baldridge. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. Thesis, University of Edinburgh, Edinburgh.

J. M. Baldridge and G.J. M. Kruijff. 2002. Coupling CCG with hybrid logic dependency semantics. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia PA.

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

J. Baur. 1999. *Syntax und Semantik mathematischer Texte*. Diplomarbeit, Fachrichtung Computerlinguistik, Universit¨at des Saarlandes, Saarbr¨ucken, Germany.

C. Benzm¨uller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayov´a, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Q. Vo, and M. Wolska. 2003a. Tutorial dialogs on mathematical proofs. In *Proc. of IJCAI'03 Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, Acapulco, Mexico.

C. Benzm¨uller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayov´a, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Q. Vo, and M. Wolska. 2003b. A Wizard-of-Oz experiment for tutorial dialogues in mathematics. In *Proc. of the AIED'03 Workshop on Advanced Technologies for Mathematics Education*, Sydney, Australia.

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

A. Fiedler and M. Gabsdil. 2002. Supporting Progressive Refinement of Wizard-of-Oz Experiments. In *Proc. of the ITS'02 Workshop on Empirical Methods for Tutorial Dialogue*, San Sebastian, Spain.

M. Glass. 2001. Processing language input in the CIRCSIM-Tutor intelligent tutoring system. In *Proc. of the 10th Conference on Artificial Intelligence in Education (AIED'01)*, San Antonio.

A. Graesser, P. Wiemer-Hastings, K. Wiemer-Hastings, D. Harter, and N. Person. 2000. Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive Learning Environments*, 8.

E. Hajičov´a, J. Panevov´a, and P. Sgall. 2000. A manual for tectogrammatical tagging of the Prague Dependency Treebank. TR-2000-09, Charles University, Prague, Czech Republic.

H. Horacek and M. Wolska. 2004. Interpreting Semi-Formal Utterances in Dialogs about Mathematical Proofs. In *Proc. of the 9th International Conference on Application of Natural Language to Information Systems (NLDB'04)*, Salford, Manchester, Springer. To appear.

G.J.M. Kruijff. 2001. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. Ph.D. Thesis, Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

J. Moore. 1993. What makes human explanations effective? In *Proc. of the 15th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ.

P. Sgall, E. Hajičov´a, and J. Panevov´a. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. Reidel Publishing Company, Dordrecht, The Netherlands.

Q.B. Vo, C. Benzm¨uller, and S. Autexier. 2003. An approach to assertion application via generalized resolution. SEKI Report SR-03-01, Fachrichtung Informatik, Universit¨at des Saarlandes, Saarbr¨ucken, Germany.

M. Wolska and I. Kruijff-Korbayov´a. 2004. Analysis of mixed natural and symbolic language input in mathematical dialogs. In *Proc.of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain. To appear.

M. Wolska and I. Kruijff-Korbayov´a. 2004. Building a dependency-based grammar for parsing informal mathematical discourse. In *Proc. of the 7th International Conference on Text, Speech and Dialogue (TSD'04)*, Brno, Czech Republic, Springer. To appear.

M. Wolska, B. Q. Vo, D. Tsovaltzi, I. Kruijff-Korbayov´a, E. Karagjosova, H. Horacek, M. Gabsdil, A. Fiedler, C. Benzm¨uller, 2004. An annotated corpus of tutorial dialogs on mathematical theorem proving. In *Proc. of 4th International Conference On Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. To appear.

C. Zinn. 1999. Understanding mathematical discourse. In *Proc. of the 3rd Workshop on the Semantics and Pragmatics of Dialogue (Amstelogue'99)*, Amsterdam, The Netherlands.