

Building a dependency-based grammar for parsing informal mathematical discourse

Magdalena Wolska Ivana Kruijff-Korbayová

Computerlinguistik, Universität des Saarlandes

Building 17, Postfach 15 11 50

66041 Saarbrücken, Germany

{magda,korbay}@coli.uni-sb.de

Abstract

Discourse in formal domains, such as mathematics, is characterized by a mixture of telegraphic natural language and embedded formal expressions. Little is known about the suitability of input analysis methods for mathematical discourse in a dialog setting, due to the lack of empirical data. In this paper, we report on the development of a dependency-based lexicalist grammar for parsing input in dialogs on mathematics. We investigate the semantic relations that build the linguistic meaning of mathematical text in order to inform construction of the grammar. The ultimate goal is to provide a uniform analysis of texts with different degrees of verbalization: ranging from symbolic alone to fully worded mathematical expressions.

1 Introduction

Language understanding in dialog systems, be it with speech or text interface, is commonly performed using shallow syntactic analysis combined with keyword spotting. Statistical methods can be employed, however, they require pre-constructed scripts as gold-standard answers (Graesser et al., 2000). These techniques remain oblivious of such aspects of discourse meaning as causal relations, modality, negation, or scope of quantifiers. When precise understanding is needed, closed-questions are used to elicit short answers of little syntactic variation (Glass, 2001). Relying on restricted language input, however, goes against empirical findings which show that flexible natural language dialog supports active learning (Moore, 1993).

In the DIALOG¹ project, we aim at constructing a prototype of a flexible dialog system for tutoring mathemat-

ical theorem proving (Benzmüller et al., 2003a). To conduct an investigation into the use of natural language in written dialogs on mathematical proofs and to identify linguistic phenomena that will impose specific requirements on input understanding and dialog management, we collected a corpus of dialogs with a simulated tutoring system for proofs in naive set theory. Analysis of the collected data revealed tight interleaving of natural language and mathematical expressions in both student and tutor turns. The level of mathematical formality of the propositional content varied from formula(e) alone, through formula(e) with a minimal amount of natural language (e.g. verbalizing only logical connectives), up to fully worded descriptions of propositions. Moreover, the language used by the students was often informal and imprecise.

Given the complexity of the language phenomena observed in the collected corpus, we adopted a methodology of deep analysis of the input text. Our objective has been to develop an approach to analyzing informal mathematical text in which (i) phenomena related to the tight interleaving of natural and mathematical languages would be accounted for and (ii) different degrees of the mathematical content verbalization would be treated uniformly, and (iii) imprecise formulations would be interpreted using domain knowledge. To this end, we are building a grammar that enables deep analysis of the symbolic content on a par with the natural language. We adopt a dependency-based framework as a representation of the deep structure level of the utterance. To inform the development of the grammar, we are annotating the linguistic meaning of the utterances in our corpus. As the annotation paradigm, we take on an existing taxonomy of semantic dependency relations, *tectogrammatical relations* (TRs) used in the Prague Dependency Treebank, and attempt to apply it to our specialized domain.

On the other hand, given the telegraphic nature of the language and common ungrammaticality, we also started

¹The DIALOG project is part of the Collaborative Research Center on *Resource-Adaptive Cognitive Processes* (SFB 378) at Universität res Saarlandes; <http://www.coli.uni-sb.de/sfb378/>

to investigate how to combine the deep semantically oriented analysis with shallow techniques. However, we do not discuss ways of combining the deep and shallow analysis approaches here. In this paper, we concentrate on analyzing well-formed sentences.

The paper is organized as follows: in Sect. 2, we present the corpus and the language phenomena; in Sect. 3, we place the task of input analysis in the system setup; in Sect. 4, we present the annotation effort that guides the grammar development; in Sect. 5, we show our input analysis approach that captures the mathematical and the linguistic content in a uniform way; in Sect. 6, we show example analyses; in Sect. 7, we mention other work related to understanding mathematical discourse on the one hand, and to deep level annotation in corpora on the other; finally, we present conclusions in Sect. 8.

2 Linguistic data

2.1 Corpus collection

24 subjects with varying educational backgrounds and little to fair prior mathematical knowledge participated in a *Wizard-of-Oz* experiment (Benzmüller et al., 2003b). At the tutoring session, they were asked to prove 3 theorems²: (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) When $A \subseteq K(B)$, then $B \subseteq K(A)$. The subjects were instructed to enter proof steps to encourage dialog with the system. Buttons were available in the interface for inserting mathematical symbols, while the text was typed on the keyboard. The subjects and the tutor were free in the linguistic expression of their turns. The dialogs were carried out in German.

The collected corpus consists of 66 dialog logfiles, containing on average 12 turns. The total number of sentences is 1115, of which 393 are student sentences. More details on the corpus and the ongoing corpus annotation are presented in (Wolska et al., 2004).

2.2 Language phenomena

Below, we present some of the identified characteristics of the language of written mathematical dialogs. Example utterances from the corpus are shown in Figure 1.

- Mathematical language, often semi-formal, is interleaved with natural language informally verbalizing proof steps (1). Mathematical objects (or parts thereof) lie within scope of quantifiers or negation expressed in natural language (as in (2)).
- Domain relations and concepts are described informally using imprecise and/or ambiguous natural language expressions. In (3) and (4), **be-in** is ambiguous between the domain relations of “subset” and

“element”, and **be_outside_of**, **be_different**, and **have-no-common-elements** are informal intuitive descriptions of empty set intersection.

- Sometimes, “actions” involving terms, formulae or parts thereof are verbalized before the appropriate formal operation is performed as in (7). The meaning of the “action verbs” is needed for the interpretation of the intended proof-step.
 - Generic and specific references appear within one utterance as in (5) 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 in discourse.
 - Co-reference phenomena specific to informal mathematical discourse involve (parts of) formulae. In particular, entities denoted with the same literals may not co-refer, as in (6). Here, co-referential and non-coreferential use of symbolic identifiers A and B is mixed: 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 in the second sentence.
 - Metonymic expressions are used to refer to structural sub-parts of formulae, resulting in predicate structures incompatible in terms of selection restrictions; in (8), the predicate **be_valid_for**, in this domain, normally takes an argument of sort CONSTANT, TERM or FORMULA, rather than LOCATION; in (9), **apply** takes two arguments: one of sort RULE and the other of sort TERM or FORMULA, not OPERATION ON SETS.
 - Discourse deictic expressions include references to structural parts of terms and formulae such as “the left side” or “inner bracket” (in (8) and (10)) which are incomplete specifications: the former refers to a part of an equation, the latter, metonymic, to an expression enclosed in parenthesis. Moreover, these expressions require discourse referents for the sub-parts of mathematical expressions to be available.
- In the work so far, we have been concentrating on developing an approach to analyzing informal mathematical text in which (i) phenomena related to the tight interleaving of natural and mathematical languages would be accounted for and (ii) different degrees of the mathematical content verbalization would be treated uniformly, and (iii) imprecise formulations would be interpreted using domain knowledge. In the next sections, we place the input understanding task in the overall system setup and discuss our approach to a uniform analysis of input with a mixture of natural language and mathematical expressions.

² K stands for set complement and P for power set.

- (1) $A \cap B$ auf der linken Seite **ist** \in **von** $C \cup (A \cap B)$
‘ $A \cap B$ on the left-hand side **is** \in **of** $C \cup (A \cap B)$ ’
- (2) B enthaelt **kein** $x \in A$
‘ B contains **no** $x \in A$ ’
- (3) 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 ’
- (4) dann sind A und B (**vollkommen**) **verschieden**, haben **keine gemeinsamen Elemente**
‘then A and B are (**completely**) **different**, have **no common elements**’
- (5) **Potenzmenge** enthaelt alle Teilmengen, also auch $(A \cap B)$
‘**A power set** contains all subsets, hence also $(A_i \cap B_j)$ ’
- (6) 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 $K(C \cup D)$
‘DeMorgan-Regel-2 means: $K(A \cap B) = K(A) \cup K(B)$. In this case: e.g. $K(A) =$ the term $K(A \cup B)$
 $K(B) =$ the term $K(C \cup D)$ ’
- (7) Ich **zerlege** jetzt die Potenzmenge: $P(C \cup (A \cap B)) \supseteq P(C) \cup P(A \cap B)$
‘Now I’m **splitting** the power set: $P(C \cup (A \cap B)) \supseteq P(C) \cup P(A \cap B)$ ’
- (8) Dann gilt fuer **die linke Seite**, wenn $C \cup (A \cap B) = (A \cup C) \cap (B \cup C)$, der Begriff $A \cap B$ dann ja schon dadrin und ist somit auch Element **davon**.
‘Then for **the left side** holds when $C \cup (A \cap B) = (A \cup C) \cap (B \cup C)$, the term $A \cap B$ is already there, and so an element of **it**’
- (9) de morgan regel 2 **auf beide komplemente angewendet**
‘de morgan rule 2 **applied to both complements**’
- (10) ... $A \cap B \in P((A \cup C) \cap (B \cup C))$... Distributivitaet von Vereinigung ueber Durchschnitt: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Hier dann also: $C \cup (A \cap B) = (A \cup C) \cap (B \cup C)$ Dies fuer **die innere Klammer**.
‘... $A \cap B \in P((A \cup C) \cap (B \cup C))$... Distributivity of union over intersection: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ So here: $C \cup (A \cap B) = (A \cup C) \cap (B \cup C)$ This for **the inner bracket**.’

Figure 1: Example utterances.

3 Tasks of input understanding

The central component of our system is the Dialog Manager (DM). The user’s input is captured by the DM and passed on to the input understanding module. The task of utterance and discourse interpretation is three-fold. First, it is to construct a representation of the utterance’s *linguistic meaning* and to interpret the utterance (using ontological information and a semantic lexicon of the domain) thus assigning it a domain-specific *sense*. Second, it is to identify and separate within the utterance: (i) parts which constitute student’s (partial) proof contribution that can be verified by a theorem prover; (ii) parts which constitute meta-communication with the tutor (e.g., “Ich habe die Aufgabenstellung nicht verstanden.”; eng. “I did not understand the task.”) that are not to be processed by a prover. Finally, it is to update the discourse representation in the Informaton State (IS)³. The analyzed input is returned to the DM and merged with the discourse context maintained in the IS. The DM subsequently either passes the proof-relevant part (if any) to the Proof Manager (PM) for evaluation by the theorem prover or carries out the dialog according to its specified rules. The task of the PM is to: (i) communicate directly with the the-

³The structure of IS is modelled after Trindi (Traum and Larsson, 2003).

orem prover⁴; (ii) build and maintain a representation of the proof constructed by the student; (iii) if necessary, check type compatibility of proof-relevant entities introduced as new in discourse; (iv) check consistency and appropriateness of each of the interpretations constructed by the analysis module, with the proof context.

We are building a lexically-based dependency grammar for syntactic and semantic analysis of the input utterances. At the same time, we are analyzing the corpus and annotating it with semantic dependency relations. The investigation of the semantic relations that build the linguistic meaning of mathematical text informs the construction of the grammar.

4 Linguistic Meaning Annotation

By linguistic meaning (LM), we understand deep semantics in the sense of the Prague School notion of sentence meaning as employed in the Functional Generative Description (FGD) (Sgall et al., 1986; Kruijff, 2001). LM represents the literal meaning of an utterance, rather than an interpretation within a specific domain. It 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 in-

⁴We are using a version of Ω MEGA adapted for assertion-level proving (Vo et al., 2003).

interpretation of plurals, synonymy, or ambiguity are not resolved.

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.

To derive our set of semantic relations we generalise and simplify the collection of Praguian tectogrammatical relations in (Hajičová et al., 2000). The reason for this simplification is, among others, to distinguish which of the roles have to be understood metaphorically given our specific sub-language domain. In order to allow for ambiguity in the recognition of TRs, we organize them hierarchically into a taxonomy.

The most commonly occurring relations in our context (aside from the inner participant relations of Actor and Patient) are Cause, Condition, and Result-Conclusion⁵:

1. Da $[A \subseteq K(B)]_{\langle \text{CAUSE} \rangle}$, 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 '
- 1'. Da $A \subseteq K(B)$ gilt, [alle x , die in A sind sind nicht in B] _{$\langle \text{RES} \rangle$}
2. Wenn $[A \subseteq K(B)]_{\langle \text{COND} \rangle}$, dann $A \cap B = \emptyset$
'If $A \subseteq K(B)$, then $A \cap B = \emptyset$ '
- 2'. Wenn $A \subseteq K(B)$, dann $[A \cap B = \emptyset]_{\langle \text{RES} \rangle}$

These coincide with the rhetorical relations with which we would like to model the argumentative structure of the proof. At this point, we do not disambiguate between the Cause/Condition relations and the Result relation within one sentence. We are investigating the correlations between the Cause, Condition, and Result relations, and the student's structure of reasoning (forward or backward). For example, the Condition relation may indicate that the proposition has not yet been proven, whereas using Cause the student considers it proven.

Other commonly found TRs include Norm-Criterion:

3. [nach deMorgan-Regel-2] _{$\langle \text{NORM} \rangle$} ist
 $K((A \cup B) \cap \dots) = \dots$
'according to deMorgan-rule-2 it holds that...'
4. $K(A \cup B)$ ist [laut DeMorgan-1] _{$\langle \text{NORM} \rangle$} $K(A) \cap K(B)$
' $K(A \cup B)$ equals, according to deMorgan-1, $K(A) \cap K(B)$ '

We group other modifications into sets of HasProperty, GeneralRelation (e.g. for adjectival and clausal modification), and Other (a catch-all category), for example:

5. dann muessen alla A und B [in C] _{$\langle \text{PROP-LOC} \rangle$} enthalten sein
'then all A and B have to be contained in C '
6. Alle x , [die in B sind] _{$\langle \text{GENREL} \rangle$} ...
'All x that are in B ...'

⁵The presentation of the annotation is schematic.

7. alle elemente [aus A] _{$\langle \text{PROP-FROM} \rangle$} sind in $K(B)$ enthalten
'all elements from A are contained in $K(B)$ '
8. Aus $A \subseteq U \setminus B$ folgt [mit $A \cap B = \emptyset$] _{$\langle \text{OTHER} \rangle$} , $B \subseteq U \setminus A$.
'From $A \subseteq U \setminus B$ follows with $A \cap B = \emptyset$, that $B \subseteq U \setminus A$.'

where PROP-LOC denotes the HasProperty relation of type Location, GENREL is a general relation as in relative clause modification, and PROP-FROM is a HasProperty relation of type Direction-From or From-Source.

Using TRs rather than surface grammatical roles provides a generalized view of the correlations between domain-specific content and its linguistic realization. By annotating the corpus with the dependency-based semantic relations that build up the linguistic meaning of the utterances, we hope to be able to investigate these correlations in a systematic way. Currently, the main purpose of the annotation is to guide the development of the deep parser grammar.

5 Syntactic and Semantic Analysis

Our goal is to provide a uniform analysis of inputs of varying degrees of verbalization. This is achieved by the use of one grammar for analyzing utterances that contain both natural language and mathematical expressions.

The analysis proceeds in 2 stages: (i) At the pre-processing stage, aside from standard pre-processing⁶, mathematical expressions are identified, analysed, categorized, and substituted with default lexicon entries encoded in the grammar; (ii) Next, the input is syntactically parsed and a representation of its LM is constructed along with the parse. The LM is subsequently embedded in the discourse context and interpreted using an ontology and a semantic lexicon of the domain.

We address the interpretation procedure for syntactically well-formed utterances in the following sections.

5.1 Parsing math expressions

The task of the mathematical expression parser is to identify mathematical expressions. The identified mathematical expressions are subsequently verified as to syntactic validity, and categorized. Identification of mathematical expressions within word-tokenized text is performed using simple indicators: single character tokens (with the characters P and K standing for power set and set complement respectively), mathematical symbol unicodes, and new-line characters. The tagger converts the infix notation used in the input into an expression tree from which the following information is available: surface sub-structure (e.g., "left side" of an expression, list of sub-expressions, list of bracketed sub-expressions) and expression type based on the top level operator (e.g.,

⁶Standard pre-processing includes sentence and word tokenization, etc.

CONSTANT, TERM, FORMULA, 0_FORMULA (formula missing left argument), etc.).

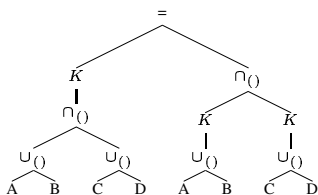


Figure 2: Tree representation of the formula $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cap K(C \cup D))$

For example, the expression $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cap K(C \cup D))$ is represented by the formula tree in Fig. 2. The bracket subscripts indicate the operators heading sub-formulae enclosed in parentheses. Given the expression’s top node operator, =, it 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: $A \cup B$, $C \cup D$, $(A \cup B) \cap (C \cup D)$, etc.

5.2 Semantic analysis

The task of the deep parser is to produce an LM representation of syntactically well-formed sentences. The analysis is performed using openCCG⁷, an open source Multi-Modal Combinatory Categorical Grammar (MM-CCG) parser. MMCCG is a lexicalist grammar formalism in which application of combinatory rules is controlled through context-sensitive specification of modes on slashes (Baldrige 2002; Baldrige and Kruijff, 2003). The linguistic meaning, built in parallel with the syntax, is represented using Hybrid Logic Dependency Semantics (HLDS), a hybrid logic representation that allows a compositional, unification-based construction of HLDS terms with CCG (Baldrige and Kruijff 2002). Dependency relations between heads and dependents are explicitly encoded in the lexicon as modal relations.

Default lexical entries (e.g. CONSTANT, TERM, FORMULA, 0_FORMULA; cf. 5.1) are encoded in the grammar for the mathematical expression categories. Syntactic signs corresponding to the mathematical expressions are treated in the same way as those of linguistic lexical entries: they are part of the deep analysis, enter into dependency relations. The signs for a lexical entry FORMULA are S , NP , and N .

For example, depending on the discourse context, the utterance “B enthaelt $x \in A$ ” (eng. ‘B contains $x \in A$ ’) can have multiple interpretations. In one of the interpretations, B is a FORMULA⁸ that contains the mathematical

⁷<http://openccg.sourceforge.net>

⁸In prior discourse, there may have been an assignment $B := \phi$, where ϕ is a formula, in which case, B would be known from discourse context to be of type FORMULA (similarly for

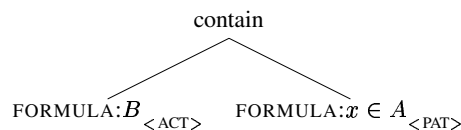


Figure 3: TRs in the reading ‘Formula B contains a formula $[x \in A]$ ’ of the utterance “ B contains $[x \in A]$ ”.

expression $[x \in A]$. This reading is shown schematically in Fig. 3. In another interpretation, B is a set CONSTANT⁹ in prior discourse, and the utterance reads ‘Set B contains no x such that x is an element of the set A ’. The latter interpretation is obtained by structurally partitioning the mathematical expression at its top node operator in the following way: $[x] [\in A]$, where the expression $[\in A]$ is categorized as a formula missing left argument (0_FORMULA; cf. 5.1). This reading is shown in Fig. 4.

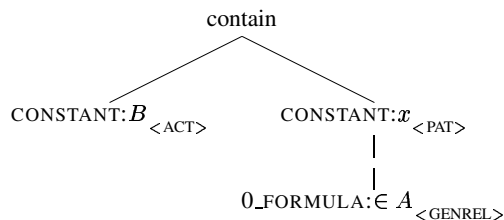


Figure 4: TRs in the reading ‘Set B contains x that is an element of the set A .’ of “ B contains $[x] [\in A]$ ”.

In both readings, the verb “enthaelt” represents the meaning **contain** and takes dependents in the relations Actor and Patient. In the second reading, Fig. 4, the identified CONSTANT, x , takes a dependent 0_FORMULA in the GeneralRelation.

The linguistic meaning of the first reading, ‘Formula B contains a formula $[x \in A]$ ’, is represented by the following hybrid logic formula:

$$@h1(\text{contain} \wedge \langle \text{ACT} \rangle (f1 \wedge \text{FORMULA}:B) \wedge \langle \text{PAT} \rangle (f2 \wedge \text{FORMULA}:x \in A))$$

where $h1$ is the state where the proposition **contain** is true, and the nominals $f1$ and $f2$ represent the discourse referents for the dependents of the head **contain**, in the relations Actor and Patient, respectively. The LM of the reading ‘Set B contains x that is an element of the set A ’ is represented by the following formula:

$$@h2(\text{contain} \wedge \langle \text{ACT} \rangle (c1 \wedge \text{CONSTANT}:B) \wedge \langle \text{PAT} \rangle (c2 \wedge \text{CONSTANT}:x \wedge \langle \text{GENREL} \rangle (f3 \wedge 0_FORMULA:\in A))$$

where $f3$ represents the GeneralRelation dependent of CONSTANT x whose discourse referent is given by nominal $c2$.

term assignment).

⁹By CONSTANT we mean a set or element variable, such as A , x , denoting a set A or an element x respectively.

Presently, our grammar supports the syntactic constructions that occur most frequently in the corpus. We are extending the grammar to cover more syntactic phenomena (e.g. word order phenomena). At the same time, we are working on applying the HLDS-based approach to discourse representation, as presented in (Kruijff and Kruijff-Korbayová 2001).

5.3 Domain interpretation

The linguistic meaning representations obtained from the parser are interpreted with respect to a knowledge base (KB). Mathematical knowledge bases are typically highly structured into mathematical sub-domains and usually form a dependency/inheritance graph (e.g. the MBase system (Kohlhase and Franke, 2000)). To be able to interface to an existing KB resource, such as MBase, we are constructing a domain ontology that reflects the domain concepts database, and is augmented to allow resolution of ambiguities introduced by natural language. The domain objects and relations in the constructed ontology are organized in a specialization hierarchy where prominent aspects of their semantics are expressed as properties with value restrictions.

For example, the previously mentioned predicate **contain** 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 specializations of the ambiguous semantic relations are encoded in the ontology, while a semantic lexicon provides interpretations of the predicates. At the domain interpretation stage, the semantic lexicon is consulted to translate the tectogrammatical frames of the predicates into the semantic relations represented in the domain ontology. For the predicate **contain**, the lexicon contains the following rules:

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”

where, in the ontology, FORMULA is a STRUCTURED OBJECT that allows an embedding (a SUB-FORMULA), and **container** and **containee** are specializations of the semantic relation CONTAINMENT. Translation rules that consult the ontology expand the meaning of the predicates to all their 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 within the given proof context, this task is delegated to the Proof Manager. Simple pattern-based rules

translate the meaning interpretations into FOL formula and those, in turn, into statements in a proof representation language (Autexier et al.2004) used for input to the Proof Manager.

The present work concentrates on extending the semantic lexicon of the domain and the ontology. In particular, we will address semantically complex operators such as ‘vice-versa’ as in “Wenn alle A in $K(B)$ enthalten sind und dies auch **umgekehrt** gilt, ...” (eng. ‘If all A are contained in $K(B)$ and this also holds vice-versa’) which can be interpreted as ‘all $K(B)$ are contained in A ’ or as ‘all B are contained in $K(A)$ ’.

6 Example analysis

In this section, we illustrate the mechanics of the approach on the following examples.

(1) B enthaelt kein $x \in A$ [B contains no $x \in A$]

(2) $A \cap B \in \{A \cap B\}$

(3) A enthaelt keinesfalls Elemente, die auch in B sind.

’A contains no elements that are also in B’

Example (1) shows the tight interaction of natural language and mathematical formulae. The intended reading of the scope of negation is over a part of the formula following it, rather than the whole formula. The analysis proceeds as follows.

The formula tagger first identifies the formula $\langle x \in A \rangle$ and substitutes it with the generic entry FORMULA represented in the lexicon. If there was no prior discourse entity for B to verify its type, the type is ambiguous between CONSTANT, TERM, and FORMULA (cf. the example in Sect. 5.2). The sentence is assigned four alternative readings: (i) “CONST contains no FORMULA”, (ii) “TERM contains no FORMULA”, (iii) “FORMULA contains no FORMULA”, (iv) “CONST contains no CONST 0_FORMULA”.

The last reading is obtained by partitioning an entity of type FORMULA in meaningful ways, taking into account possible interaction with preceding modifiers. Here, given the quantifier “no”, the expression $\langle x \in A \rangle$ has been split into its surface parts as follows: $\langle [x][\in A] \rangle$ ¹⁰. $[x]$ has been substituted with a generic lexical entry CONSTANT, and $[\in A]$ with a symbolic entry for a formula missing its left argument (cf. Sect. 5.1). The readings (i) and (ii) are rejected because of sortal incompatibility.

¹⁰There are other ways of constituent partitioning of the formula at the top level operator to separate the operator and its arguments: $\langle [x][\in][A] \rangle$ and $\langle [x \in][A] \rangle$. Each of the partitions obtains its appropriate type corresponding to a lexical entry available in the grammar (e.g., the $[x \in]$ chunk is of type FORMULA_0 for a formula missing its right argument). Not all the readings, however, compose to form a syntactically and semantically valid parse of the given sentence.

The remaining linguistic meanings and readings of the sentence are:¹¹

- for ‘FORMULA contains no FORMULA’:
 $s: (@k1(\text{kein} \wedge \langle \text{RESTR} \rangle f2 \wedge \langle \text{BODY} \rangle (e1 \wedge \text{enthalten} \wedge \langle \text{ACT} \rangle (f1 \wedge \text{FORMULA}) \wedge \langle \text{PAT} \rangle f2)) \wedge @f2(\text{FORMULA}))$
 ‘‘formula B embeds no subformula $x \in A$ ’’
- for ‘CONST contains no CONST 0_FORMULA’:
 $s: (@k1(\text{kein} \wedge \langle \text{RESTR} \rangle x1 \wedge \langle \text{BODY} \rangle (e1 \wedge \text{enthalten} \wedge \langle \text{ACT} \rangle (c1 \wedge \text{CONST}) \wedge \langle \text{PAT} \rangle x1)) \wedge @x1(\text{CONST} \wedge \langle \text{HASPROP} \rangle (x2 \wedge 0_{\text{FORMULA}})))$
 ‘‘B contains no x such that x is an element of A ’’

Next, the semantic lexicon is consulted to translate these readings into their domain interpretations. The relevant lexical semantic entries were presented in Sect. 5.3. Using the linguistic meaning, the semantic lexicon, and the ontology, we obtain four interpretations paraphrased below:

- for ‘FORMULA contains no FORMULA’:
 (1.1) ‘it is not the case that $\langle \text{PAT} \rangle$, the formula, $x \in A$, is a subformula of $\langle \text{ACT} \rangle$, the formula B’;
- for ‘CONST contains no CONST 0_FORMULA’:
 (1.2a) ‘it is not the case that $\langle \text{PAT} \rangle$, the constant x , $\subseteq \langle \text{ACT} \rangle$, B, and $x \in A$,
 (1.2b) ‘it is not the case that $\langle \text{PAT} \rangle$, the constant x , $\in \langle \text{ACT} \rangle$, B, and $x \in A$,
 (1.2c) ‘it is not the case that $\langle \text{PAT} \rangle$, the constant x , $\subset \langle \text{ACT} \rangle$, B, and $x \in A$.’

The interpretation (1.1) is verified in the discourse context with information on structural parts of the discourse entity B of type FORMULA, while (1.2a-c) are translated into FOL formulae and into the proof representation language on the input side of the Proof Manager.

Example (2) contains one mathematical formula. Such utterances are the simplest to analyze: The formulae identified by the mathematical expression tagger are translated directly into the PM input language.

The example (3) shows an utterance with domain-relevant content linguistically verbalized. The analysis of fully verbalized utterances proceeds similarly to the example (1): The mathematical items are substituted with the appropriate generic lexical entries (here, A and B are substituted with their three alternative readings: CONSTANT, TERM, and FORMULA, yielding several utterance readings: ‘CONST contains no elements that are also in CONST’, ‘TERM contains no elements that are also in TERM’, etc.). Next, the sentence is analyzed by the grammar. The semantic roles of Actor and Patient associated with the verb ‘‘contain’’ are taken by A and ‘‘elements’’ respectively; quantifier ‘‘no’’ is in the relation Restrictor with A ; the relative clause is in the GeneralRelation with ‘‘elements’’, etc. Then the hybrid logic representations of the linguistic meaning are built, and the KB is consulted to translate each meaning into its interpreted results, which are, in this case, very similar to the ones of example (1).

¹¹Irrelevant parts of the meaning representation are omitted; glosses of the hybrid formulae are provided.

7 Related work

(Zinn 2003) addresses mathematical text processing using an extended Discourse Representation Theory (DRT) approach applied to underspecification resolution. Like in the earlier work, (Zinn, 1999; Baur, 1999), analyzed are complete, carefully structured textbook proofs, and the analysis 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 differences between presentation in textbooks and in a tutorial dialog. The language in dialog is more informal than in textbooks: natural language and symbolic mathematical expressions are mixed more freely, there is a higher degree and more variety of verbalization, instantiation of variables in applied theorems is informal. Our input does not contain typesetting information for math symbols, formulae, and proof steps.

With regard to annotation of deep level relations in corpora, in the Negra Corpus¹², only surface grammatical roles are annotated. The Prague Dependency Treebank¹³ does encode deep relations of the tectogrammatical level. The Praguian annotation manual does not, however, provide definitions of the particular tectogrammatical relations, only examples in Czech. We are attempting to apply the Praguian set of tectogrammatical relations to a narrow and specialized domain. At the same time, we are trying to specify definitions of the particular relations.

8 Conclusions

We presented an approach to input understanding in a system for tutoring mathematical proofs. The analysis module uses a dependency-grammar parser to build meaning representation of input utterances. Incremental construction of the grammar is accompanied by annotation of semantic relations in a corpus collected in an experiment. The annotation guides the grammar development. At the same time, we are constructing our own dependency treebank which, in the future, we are planning to make available to the computational linguistics community.

Our current work concentrates on: (i) developing a discourse representation within the HLDS formalism used for linguistic meaning representation; (ii) extending the domain-specific knowledge resources for interpretation; (iii) allowing underspecification in the HLDS representations; (iv) combining deep and shallow analysis techniques to obtain robust analysis of ill-formed/out-of-grammar utterances. We plan to achieve a robust interpretation by a combination of methods: On the one hand,

¹²<http://www.coli.uni-sb.de/sfb378/negra-corpus/negra-corpus.html>

¹³<http://quest.ms.mff.cuni.cz/pdt/>

by using partial analyses from a chart (built by the parser during deep syntactic analysis) and constructing under-specified semantic representations from them. On the other hand, we will attempt to parse the input with an external syntactic parser or a shallow chunk parser, and then apply deep syntactic analysis to the chunks.

References

- S. Autexier, C. Benzmüller, A. Fiedler, H. Horacek and B. Q., Vo. 2004. Assertion-level proof representation with under-specification. *Electronic Notes in Theoretical Computer Science*. (In print).
- J.M. Baldridge. 2002. Lexically specified derivational control in combinatory categorial grammar. Ph.D. thesis, Institute for Communicating and Collaborative Systems, School of Informatics, 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)*, 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 EACL*, Budapest.
- J. Baur. 1999. *Syntax und Semantik mathematischer Texte*. Diplomarbeit, Fachrichtung Computerlinguistik, Universität des Saarlandes, Saarbrücken.
- C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Q. Vo, and M. Wolska. 2003a. Tutorial dialogs on mathematical proofs. In *IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*.
- C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Q. Vo, and M. Wolska. 2003b. A Wizard-of-Oz experiment for tutorial dialogues in mathematics. In *Proceedings of the AIED Workshop on Advanced Technologies for Mathematics Education*, Sydney, Australia.
- M. Glass. 2001. Processing language input in the CIRCSIM-Tutor intelligent tutoring system. In *Proc. of the 10th AIED Conference*, 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. Hajičová, J. Panevová, and P. Sgall. 2000. A manual for teletogrammatical tagging of the Prague Dependency Treebank. TR-2000-09, Charles University, Prague.
- M. Kohlhase and A. Franke. 2000. MBase: Representing knowledge and context for the integration of mathematical software systems. *Journal of Symbolic Computation*, 32(4).
- G.-J. Kruijff and I. Kruijff-Korbayová. 2001. A hybrid logic formalization of information structure sensitive discourse interpretation. In V. Matoušek, P. Mautner, R. Mouček, K. Taušer (eds.) *Proc. of the 4th International Conference on Text, Speech and Dialogue (TSD'2001)*, Springer. Železná Ruda.
- G.-J. M. Kruijff. 2001. *A Categorical-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. Ph.D. thesis, Charles University, Prague.
- 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. Hajičová, and J. Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. Reidel Publishing Company, Dordrecht.
- D. R. Traum and S. Larsson. 2003. The information state approach to dialogue management. In *Current and New Directions in Discourse and Dialogue*. Kluwer.
- B. Q. Vo, C. Benzmüller, and S. Autexier. 2003. Assertion application in theorem proving and proof planning. In *Proc. of the 18th International Joint Conference on Artificial Intelligence*, Acapulco.
- M. Wolska, B. Q. Vo, D. Tsovaltzi, I. Kruijff-Korbayová, E. Karajosova, H. Horacek, M. Gabsdil, A. Fiedler, and C. Benzmüller. 2004. An annotated corpus of tutorial dialogs on mathematical theorem proving. In *Proc. of 4th International Conference on Language Resources and Evaluation*, Lisbon. (In print)
- C. Zinn. 1999. Understanding mathematical discourse. In *Proc. of Amstelogue'99*, Amsterdam.
- C. Zinn. 2003. Computational framework for understanding mathematical discourse. *Logic Journal of the IGPL*, 11(4).