Linguistics and Logic of Common Mathematical Language

Peter Koepke and Bernhard Schröder
University of Bonn, Germany

Brouwer Seminar

Nijmegen, December 12, 2006

Contents

- The Gödel completeness theorem
- MIZAR proof of the Gödel completeness theorem
- The NAPROCHE project (NAtural language PROof CHEcking)
- Linguistics of mathematical language
- TeX-quality interface: $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$
- Demonstration

Gödel’s Completeness Theorem

Every valid formula of the
special function calculus
is provable

(Kurt Gödel, *Die Vollständigkeit der Axiome des logischen Funktionenkalküls*, 1930)

P. Braselmann and K., A formal proof of Gödel's completeness theorem, a series of 7 articles in:
Formalized Mathematics 13 (2005), 5-53,

corresponding to the MIZAR articles

1. `SUBSTUT1.MIZ`: Definition of substitution
2. `SUBSTUT2.MIZ`: Technical facts about substitutions
3. `SUBLEMMA.MIZ`: The substitution lemma
4. `CALCUL_1.MIZ`: Sequent calculus; correctness
5. `CALCUL_2.MIZ`: Technical facts about the sequent calculus
6. `HENMODEL.MIZ`: Consistency; construction of Henkin-models
7. `GOEDELCP.MIZ`: Proof of the Gödel Completeness Theorem

The MIZAR system (1973 -) of Andrzej Trybulec

Language modeled after
“mathematical vernecular”

Natural deduction style

Automatic proof checker

Large mathematical library

Journal
Formalized Mathematics

www.mizar.org

Original MIZAR in GOEDELCP.MIZ:

```
begin :: Goedel's Completeness Theorem,
:: Ebb et al, Chapter V, Completeness Theorem 4.1
theorem
  still_not-bound_in X is finite & X |= p implies X |- p
  proof
    assume A1: still_not-bound_in X is finite & X |= p;
    now assume not X |- p; then
      reconsider CX = X \/ {'not' p} as Consistent Subset of CQC-WFF
        by HENMODEL:9;
A2:  for A,J,v holds not J,v |= CX
...
...
...
        hence not J,v |= CX;
      end;
      still_not-bound_in 'not' p is finite by CQC_SIM1:20; then
      still_not-bound_in {'not' p} is finite by Th26; then
      still_not-bound_in X \/
      still_not-bound_in {'not' p} is finite by A1,FINSET_1:14; then
      still_not-bound_in CX is finite by Th27; then
      consider CZ,JH1 such that A8: JH1,valH |= CX by Th34;
      thus contradiction by A2,A8;
    end;
    hence thesis;
  end;
```

Formal (First-Order) Language $\longleftrightarrow$ Common Mathematical Language - ''Narrowing the Gap''

$\longrightarrow$ Mathematical logic: design formal languages which are similar to

the common mathematical language

$\longleftarrow$ Linguistics: extract the formal content of common mathematical texts

$\longleftarrow$ Linguistics: extract the formal content of common mathematical texts

... From a linguistic perspective, the Language of Mathematics is distinguished by the fact that its core mathematical meaning can be fully captured by an intelligent translation into first-order predicate logic. ...

$\longrightarrow$ Mathematical logic: design formal languages which are similar to

the common mathematical language

(controlled languages: MIZAR, ...)

The ... project NAPROCHE aims at constructing a system which accepts a controlled but rich subset of ordinary mathematical language including TeX-style typeset formulas and transforms them into formal statements. We adapt linguistic techniques to allow for common grammatical constructs and to extract mathematically relevant implicit information about hypotheses and conclusions. Combined with proof checking software we obtain NAtural language PROof CHEckers which are prototypically used ... to teach mathematical proving.

NAPROCHE: NAtural language PROof CHEcker

- formal language using natural language constructs, grammatically correct and varied

- allowing mathematical formulas

- input through TeX quality WYSIWYG editor

- interactive proof checking

Layers of a NAPROCHE system:

Mathematical text

$$\updownarrow$$

TeX-style internal format with editing information

$$\updownarrow$$

Tokenized format

$$\updownarrow$$

First-order logic format

$$\updownarrow$$

“Accepted”/“Not accepted”, with error messages

Layers of a NAPROCHE system:

Mathematical text

$\updownarrow$ WYSIWYG editor $\TeX_{\text{MACS}}$

TeX-style internal format with editing information

$\updownarrow$ Tokenizer

Tokenized format

$\updownarrow$ NLP (natural language processing)

First-order logic format

$\updownarrow$ Proof checker

“Accepted”/“Not accepted”, with error messages

NLP: translations between natural language and first order logic:

1 divides every integer.

$$\updownarrow$$

For every $y \in \mathbb{Z}$ holds $1|y$.

$$\updownarrow$$

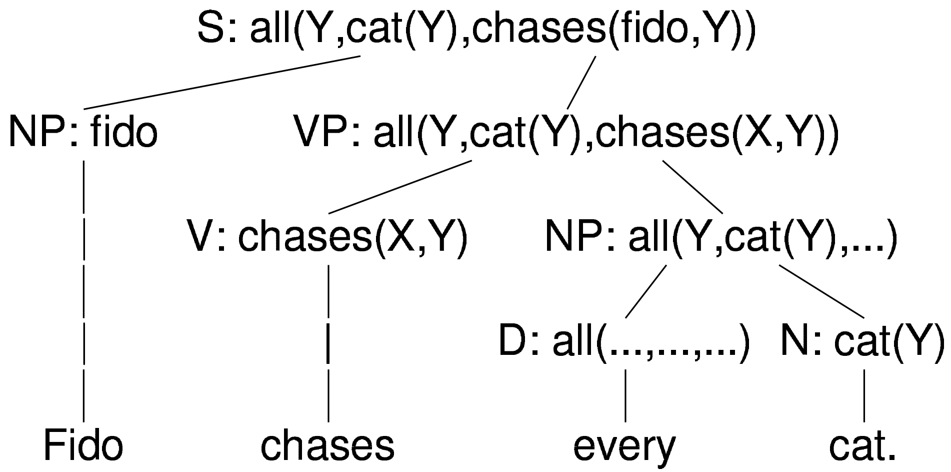
$$\forall y \in \mathbb{Z}\, 1|y$$

$$\updownarrow$$

all(Y,integer(Y),divides(1,Y))

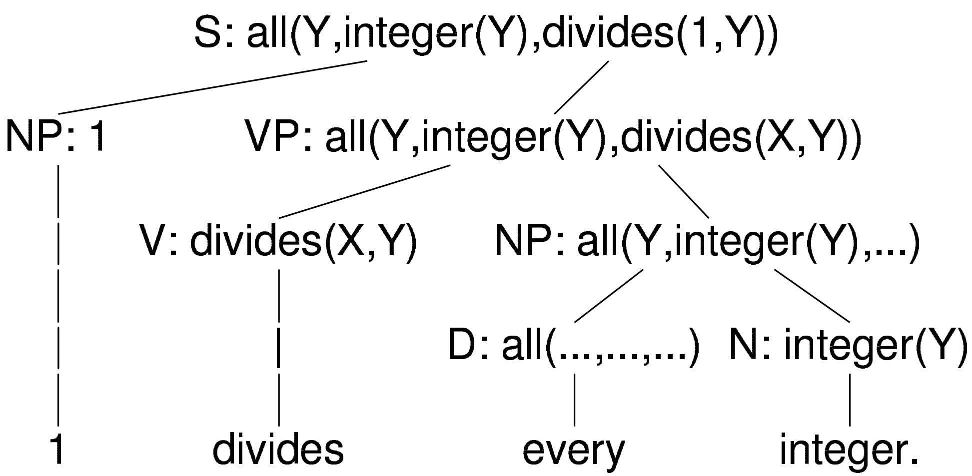
NLP: Semantics of simple natural language

“Fido chases every cat”



NLP: Semantics of simple mathematical language

“1 divides every integer.”



i.e., $\forall y \in \mathbb{Z}\ 1|y$

Further linguistic issues in (controlled) mathematical texts

- ensure grammatical correctness by grammars
- mixture of text and mathematical formulas:
  pass the formulas unchanged through the NLP layer
- resolution of anaphors: *let* X *be a set of integers and let* m *be its maximal element*. Use standard NLP methods
- identification of mathematical *keywords* structuring a text:
  Proof, qed, define, ...
- handling of ellipses: $1, 2, ..., n$

The mathematical WYSIWYG editor TeXmacs

- www.texmacs.org, GNU General Public License, under development

- TeX/LaTeX-like file format and instant on-screen rendering using the TeX font system and TeX typesetting algorithms α

- on-screen editing

- uses `scheme` as extension language

Theorem. $(\neg\varphi\vee\psi)\rightarrow(\varphi\rightarrow\psi)$.
Proof.
Let $(\neg\varphi\vee\psi)$.
Let $\neg\varphi$. Let φ. Contradiction. ψ. Thus $\varphi\rightarrow\psi$. Thus $\neg\varphi\rightarrow(\varphi\rightarrow\psi)$.
Let ψ. Let φ. ψ. Thus $\varphi\rightarrow\psi$. Thus $\psi\rightarrow(\varphi\rightarrow\psi)$.
$\varphi\rightarrow\psi$. Thus $(\neg\varphi\vee\psi)\rightarrow(\varphi\rightarrow\psi)$.
Qed.

Internal representation (.tm file)

```
<TeXmacs|1.0.6>
<style|generic>
<\body>
  Example:
  <\quotation>
    Theorem. <with|mode|math|(\<neg\>\<varphi\>\<vee\>\<psi\>)\<rightarrow\>
             (\<varphi\>\<rightarrow\>\<psi\>)>.\
    Proof.
    Let <with|mode|math|(\<neg\>\<varphi\>\<vee\>\<psi\>)>.
    Let <with|mode|math|\<neg\>\<varphi\>>. Let <with|mode|math|\<varphi\>>.
    Contradiction. <with|mode|math|\<psi\>>. Thus
    <with|mode|math|\<varphi\>\<rightarrow\>\<psi\>>. Thus
    <with|mode|math|\<neg\>\<varphi\>\<rightarrow\>(\<varphi\>\<rightarrow\>\<psi\>)>.
    Let <with|mode|math|\<psi\>>. Let <with|mode|math|\<varphi\>>.
    <with|mode|math|\<psi\>>. Thus <with|mode|math|\<varphi\>\<rightarrow\>\<psi\>>.
    Thus <with|mode|math|\<psi\>\<rightarrow\>(\<varphi\>\<rightarrow\>\<psi\>)>.
    <with|mode|math|\<varphi\>\<rightarrow\>\<psi\>>. Thus
<with|mode|math|(\<neg\>\<varphi\>\<vee\>\<psi\>)\<rightarrow\>
(\<varphi\>\<rightarrow\>\<psi\>)>.
    Qed.
  </quotation>
```

```
</body>
```