The Language of Proofs

Peter Koepke, University of Bonn

Proof theory treats mathematical proofs as formal proofs which proceed by syntactic manipulations of sequences of symbols. To the human reader proofs rather appear as tales about numbers, figures and other mathematical objects. We discuss whether standard linguistic techniques for the understanding of simple discourse are able to generate formal proofs from natural language proofs. This work may lead to natural interfaces for proof-checkers and provers and to a better understanding of the natural language and logic used in mathematical discourse. See also http://www.math.uni-bonn.de/people/naproche

Trimestre on Methods of Proof Theory in Mathematics

Max-Planck-Institut for Mathematics, Bonn, March 22, 2007

Contents

- Proofs in natural language
- The *curt* system
- Mathematical language
- Naproche: Natural language proof checking
- Details
- TEXMACS
- Examples and demonstration
- Experiences, further plans, applications

Proofs in natural language

> Every man is mortal.
> Socrates is a man.
> Socrates is mortal.

Aristotelean syllogism (Barbara).

Proofs in computer linguistics

Blackburn, Bos: Representation and Inference for Natural Language

Implementations: curt (clever use of reasoning tools)

(demo of sensitiveCurt)

Structure of the curt system:

Input text ("`Every man dies`")

↕ Parser, Tokenizer (readline.pl)

Tokenized format (`[all, man, die]`)

↕ NLP (natural language processing)

Internal representation (...)

↕

First-order logic format (`all(A, imp(man(A), die(A)))`)

↕ (fol2otter.pl, fol2mace.pl)

Input format for theorem prover Otter / model builder Mace (...)

More curt examples

"Natural language"

Every man that hates Mia hates Butch.
Marsellus is a man and hates Mia.
Marsellus hates Butch.

Mia: proper name
man: noun
hates: transitive verb
⋮

"Mathematics"

Every number that divides 10 divides 20
5 is a number and divides 10.
5 divides 20.

10: constant
is a number: unary relation
divides: binary relation
⋮

The language of mathematics I

- combination of natural language and "mathematical formulas"
- specific, defined words and figures of speech
- hypothetical constructions ("assume", "define", "let", ...)
- definitions, theorems, proofs
- typography ($\alpha, \beta, ..., \frac{a}{b}, \sqrt{\ }, \ ...$)
- graphics (diagrams, pictures, ...)
- ...

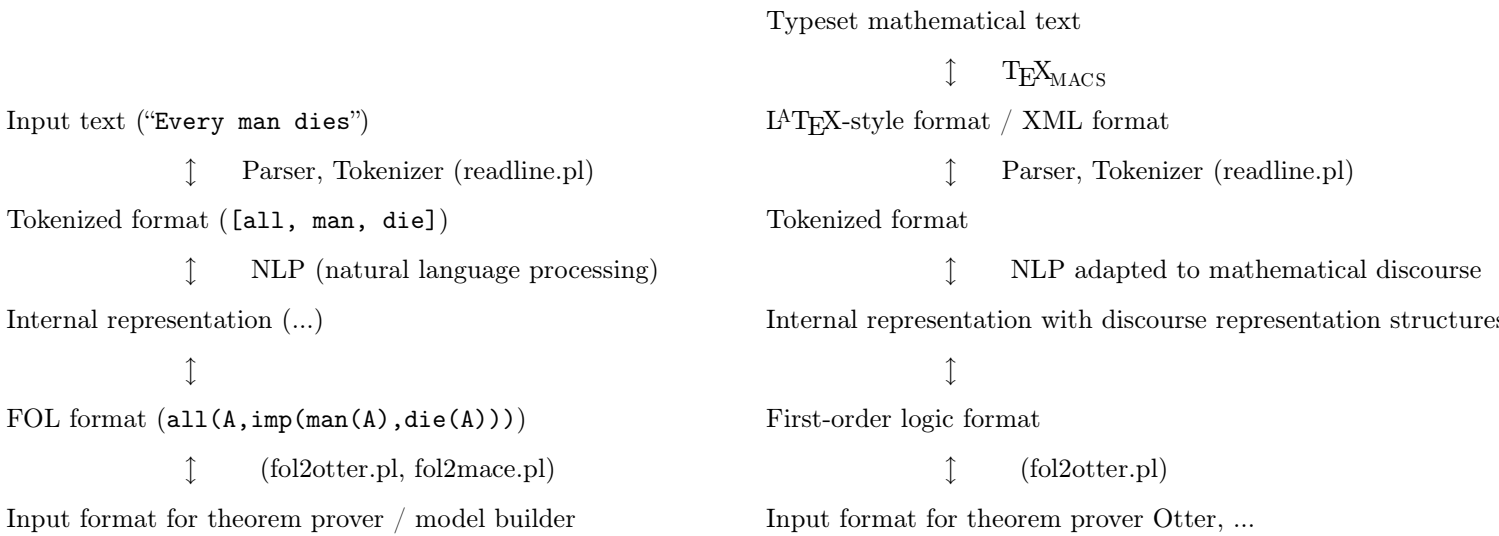
The language of mathematics II

- very concise, partially incomplete
- relying on implicit assumptions, intuitions, traditions, ...
- but: a proper mathematical text has, in principle, a definite meaning which can be expressed formally (in first-order logic and Zermelo-Fraenkel set theory).
- this may help with problems of ambiguity in mathematical texts

Linguistic studies of mathematical language

- ...
- some normative texts: D. Knuth et al: *Mathematical writing*, 1988; L. Lamport: *How to write proofs*, 1993; ...
- natural language at man/machine interfaces: P. Abrahams: *Machine verification of mathematical proofs*, 1963; ...
- checking natural mathematical language: D. Simon: *Checking natural language proofs*, 1988, and *Checking number theory proofs in natural language*, 1990; C. Zinn: *Understanding informal mathematical discourse*, 2004
- B. Löwe, B. Schröder, *Thetic Phrases in Semi-Formal Usage*
- *ProofML* - annotating mathematical proofs (B. Fisseni, B. Schröder, ...)
- ...

A "mathematical Curt"



The Naproche project (Natural language proof checking)

M. Carl, B. Fisseni, M. Klein, P. Koepke, N. Kolev, Th. Räsch, B. Schröder, J. Veldman

`www.math.uni-bonn.de/people/`

Typeset mathematical text

↕ $\TeX_{\text{MACS}}$

adapted XML format

↕ Parser, Tokenizer (readline.pl)

Tokenized format

↕ NLP adapted to mathematical discourse

Internal representation with discourse representation structures

↕

First-order logic format

↕ (fol2otter.pl)

Input format for theorem prover Otter

↕ Otter

accepted / not accepted

... 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. ...

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.

Prover / proof checker

- proof checking: e.g. MIZAR, home-grown Prolog checker
- proof checking = proving every statement from available premises and methods, with e.g. Otter, Bliksem
- Problem: how to determine the available premises
  - explicit declaration of premises: *By Theorem 5.7 ...*
  - underspecified declarations which can resolved in context: *By induction hypothesis ...*
  - closely preceding statements
- Solution (?): define a metric between statement in text and background knowledge, use premises with small distance

Discourse representation structures (DRS)

- describing the semantics of sequences of sentences, i.e., discourse (H. Kamp: *A theory of truth and semantic representation*, 1981)

- dealing with quantifiers (universal, existential, scope) and anaphora (pronouns, ...): Define a function $f: A \rightarrow B$. This function satisfies ...

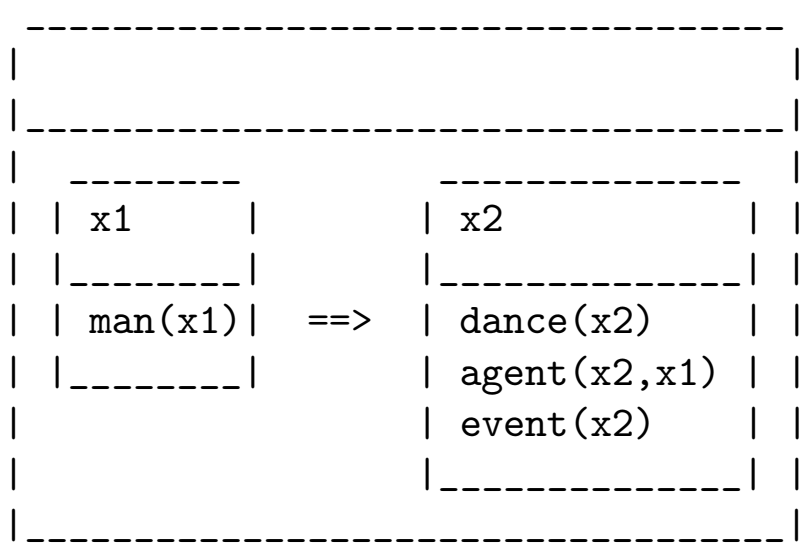
- graphical presentation of DRS: variables and properties of variables

Example (curtPPDRT)

```
> Mia dances.

> interpretations

 -------------
| x2 x1       |
|-------------|
| mia(x2)     |
| dance(x1)   |
| agent(x1,x2)|
| event(x1)   |
|-------------|
```

Example (curtPPDRT)

```
> Every man dances.

> interpretations
```

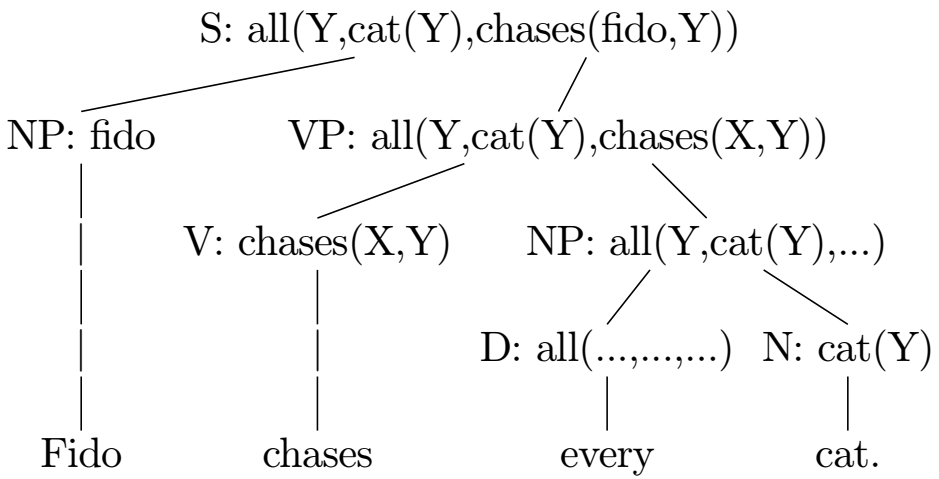


Formal Grammar: XML –> DRS semantics

- as in Blackburn-Bos, with mathematical features added

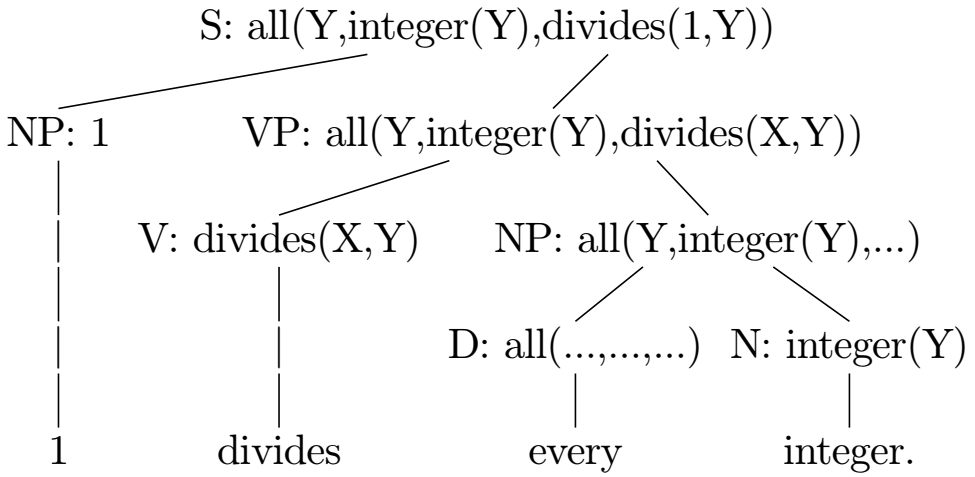
NLP: Semantics of simple natural language

"Fido chases every cat"



NLP: Semantics of simple mathematical language

"1 divides every integer."



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

The mathematical text editor TEXMACS

- WYSIWYG LaTeX-quality text editor
- uses the TeX and LaTeX algorithms and font handling
- Joris van der Hoeven 1999 -
- `www.texmacs.org`
- extendable system with scheme/guile as extension language
- can export to XML / MATHML
- can be used as an interface to other programs and for Naproche

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\>)>.
    ...
```

A weak Naproche prototype

- $\TeX_{\text{MACS}}$ + all other layers implemented in home-grown PROLOG
- simple keyword language
- Theorem / Proof / Qed construct
- no explicit references to assumptions and lemmas
- only simple proof rules

Example *de Morgan*:

Theorem. $\alpha \wedge \beta \rightarrow \neg(\neg\alpha \vee \neg\beta)$.

Proof. Assume $\alpha \wedge \beta$. Assume for a contradiction that $\neg\alpha \vee \neg\beta$. Assume $\neg\alpha$. α. Contradiction. Thus $\neg\alpha \rightarrow \bot$.

Assume $\neg\beta$. β. Contradiction. Thus $\neg\beta \rightarrow \bot$.

Hence contradiction. Thus $\neg(\neg\alpha \vee \neg\beta)$. Thus $\alpha \wedge \beta \rightarrow \neg(\neg\alpha \vee \neg\beta)$.

Qed.

Example *strict partial orders*:

Let $\leqslant$ be a partial order, and let $<$ be the associated *strict* relation:

> Let $\forall x \forall y \forall z (x \leqslant y \wedge y \leqslant z \rightarrow x \leqslant z)$.
>
> Let $\forall x\, x \leqslant x$.
>
> Let $\forall x \forall y (x \leqslant y \wedge y \leqslant x \rightarrow x = y)$.
>
> Define $\forall x \forall y (x < y \leftrightarrow x \leqslant y \wedge \neg x = y)$.

A `Naproche` proof of the transitivity of $<$:

> Theorem. $\forall x \forall y \forall z (x < y \wedge y < z \rightarrow x < z)$.
>
> Proof. Let $x < y$ and $y < z$. Then $x < y$. $x \leqslant y \wedge \neg x = y$. In particular $x \leqslant y$.
>
> Also $y < z$. Then $y \leqslant z$ and $\neg y = z$. $y \leqslant z$. $x \leqslant y$ and $y \leqslant z$. $x \leqslant z$.
>
> Assume for a contradiction that $x = z$. Then $z = x$. $y \leqslant x$. Hence $x \leqslant y$ and $y \leqslant x$. Hence $x = y$. But $\neg x = y$. Contradiction. Thus $\neg x = z$. Hence $x \leqslant z$ and $\neg x = z$. Hence $x < z$.
>
> Thus $\forall x \forall y \forall z (x < y \wedge y < z \rightarrow x < z)$. Qed.

Example *ordinals*:

We make some set theoretic assumptions. The *empty set* $\emptyset$ is characterized by:

> Assume that $\neg\exists x\, x \in \emptyset$.
>
> Assume that for all x not $x \in x$.

We define *ordinals* according to John von Neumann:

> Define for all x (Trans(x) if and only if $\forall u \forall v\,(u \in v \wedge v \in x \rightarrow u \in x)$).
>
> Define for all x (x is an ordinal iff Trans(x) $\wedge$ $\forall y\,(y \in x \rightarrow \text{Trans}(y))$).

We prove some basic facts about ordinals.

> Theorem. $\emptyset$ is an ordinal.
>
> Proof. Consider $u \in v$ and $v \in \emptyset$. Then $v \in \emptyset$. $\exists x\, x \in \emptyset$. Contradiction. Then $u \in \emptyset$. Thus $\forall u \forall v\,(u \in v \wedge v \in \emptyset \rightarrow u \in \emptyset)$. Hence Trans($\emptyset$).
>
> Consider $y \in \emptyset$. Then $\exists x\, x \in \emptyset$. Contradiction. Then Trans(y). Thus $\forall y\,(y \in \emptyset \rightarrow \text{Trans}(y))$.
>
> Hence Trans($\emptyset$) and $\forall y\,(y \in \emptyset \rightarrow \text{Trans}(y))$. Qed.

The next theorem shows that the class Ord of all ordinals is transitive:

Theorem. For all y for all x ($x \in y$ and y is an ordinal implies x is an ordinal).

Proof. Consider $x \in y$ and y is an ordinal. Then $\mathrm{Ord}(y)$. $\mathrm{Trans}(y)$ and $\forall z(z \in y \to \mathrm{Trans}(z))$. In particular $\forall z(z \in y \to \mathrm{Trans}(z))$. Observe that $x \in y$. Hence $\mathrm{Trans}(x)$.

Consider $u \in x$. $\mathrm{Trans}(y)$. So $\forall u \forall v(u \in v \wedge v \in y \to u \in y)$. Observe that $u \in x$ and $x \in y$. Hence $u \in y$. Recall that $\forall z(z \in y \to \mathrm{Trans}(z))$. Hence $\mathrm{Trans}(u)$. Thus $\forall u(u \in x \to \mathrm{Trans}(u))$.

Together we have $\mathrm{Trans}(x)$ and $\forall u(u \in x \to \mathrm{Trans}(u))$. Hence x is an ordinal. Thus for all y for all x ($x \in y$ and y is an ordinal implies x is an ordinal). Qed.

The Burali-Forti paradoxon: the class Ord of all ordinals is *not* a set:

Theorem. Not there is x such that $\forall u(u \in x \leftrightarrow \mathrm{Ord}(u))$.

Proof. Assume for a contradiction that there is x such that $\forall u(u \in x \leftrightarrow \mathrm{Ord}(u))$. Assume $\forall u(u \in x \leftrightarrow \mathrm{Ord}(u))$.

Lemma. $\mathrm{Ord}(x)$.

Proof. Let $u \in v$ and $v \in x$. Then $u \in v$. $v \in x$. $\mathrm{Ord}(v)$. Together we have $u \in v$ and $\mathrm{Ord}(v)$. So $\mathrm{Ord}(u)$. $u \in x$. Thus $\forall u \forall v(u \in v \wedge v \in x \to u \in x)$. Hence $\mathrm{Trans}(x)$.

Consider $y \in x$. Then $\mathrm{Ord}(y)$. $\mathrm{Trans}(y) \wedge\ \forall z(z \in y \to \mathrm{Trans}(z))$. In particular $\mathrm{Trans}(y)$. Thus $\forall y(y \in x \to \mathrm{Trans}(y))$.

Together we have $\mathrm{Trans}(x) \wedge\ \forall y(y \in x \to \mathrm{Trans}(y))$. Hence x is an ordinal. Qed.

Then $x \in x$. But $\neg x \in x$. Contradiction. Thus contradiction. Thus not there is x such that $\forall u(u \in x \leftrightarrow \mathrm{Ord}(u))$. Qed.

Experiences

- the prototype Naproche defines a *controlled language* with "rather natural" features (in very restricted domains)

- it has been experimentally used in a first-year undergraduate course *Mathematics for computer scientists*

- the PROLOG proof checker tries to apply every rule to all possible premises, leading to inefficiency

- introducing terms and substitutions lead to intolerable complexities

Further aspects of Naproche formalizations

- set theoretic approach: formulas and abstraction terms $\{x|\varphi\}$; efficient handling of terms using "lazy expansions"

- what would be a good axiomatic basis for this?

- implementing common figures of argumentation like "for all $i \in I$ *choose* a_i such that ...

- a set orientated internal language representation would automatically take care of many tautologies: represent $\varphi \wedge \psi$ by $\{\wedge, \varphi, \psi\}$; then $\varphi \wedge \psi = \psi \wedge \varphi$

Possible applications

- formalization of basic domains:
  "*Logic for man and machines*"
- tutorial applications
- natural language interfaces to provers and proof checkers
- linguistics
- distinguishing explicit and implicit knowledge in
  mathematical practice
- ...