Modeling Human Proof Checking in the Naproche-SAD System

by Peter Koepke

University of Bonn, Germany

Big Proof Workshop, ICMS Edinburgh, 28 May 2019

Theorem Proving

– Proving that premisses Φ entail a conclusion ψ:

$$\Phi \vdash \psi .$$

– The entailment relation $\Phi \vdash \psi$ is generated by a simple calculus (Completeness Theorem).

– $\Phi \vdash \psi$ can in principle be proved by exhibiting a derivation in the calculus.

– $\Phi \vdash \psi$ is undecidable.

– Algorithmically approximations of $\Phi \vdash \psi$ lead to very high complexities.

– Automatic Theorem Provers (ATPs) can only prove simple instances.

– $\Phi \vdash \psi$ is in general difficult to unsolvable for humans.

Interactive Theorem Proving (ITP)

- Human user and machine collaborate to prove $\Phi \vdash \psi$.
- Human provides a *formal proof* text which supports finding a derivation of $\Phi \vdash \psi$.
- Current ITPs favour procedural proof texts consisting of proof commands.

Formal Proof of the Kepler Conjecture in HOL Light and Isabelle (T. Hales et. al., 2014)

```
let the_kepler_conjecture_def = new_definition’
  ‘the_kepler_conjecture <=>
     (!V. packing V
            ==> (?c. !r. &1 <= r
                         ==> &(CARD(V INTER ball(vec 0,r))) <=
                             pi * r pow 3 / sqrt(&18) + c * r pow 2))‘;;
...
let kepler_conjecture_with_assumptions = prove_by_refinement(
  ‘!a:((((A)list)list)list). tame_classification a /\
    good_linear_programming_results a /\
    the_nonlinear_inequalities
    ==> the_kepler_conjecture
    ‘,
  (* {{{ proof *)
  [
  REPEAT WEAKER_STRIP_TAC;
  ASSUME_TAC Reduction5.restricted_hypermaps_are_planegraphs_thm;
...
```

LEMMA 1.3. *If there exists a negligible* fcc*-compatible function* $A : \Lambda \to \mathbb{R}$ *for a saturated packing* Λ*, then there exists a constant* C *such that for all* $r \geq 1$ *and all* $x \in \mathbb{R}^3$,

$$\delta(x, r, \Lambda) \leq \pi/\sqrt{18} + C/r.$$

The constant C *depends on* Λ *only through the constant* C_1.

Proof. The numerator $\operatorname{vol} B(x, r, \Lambda)$ of $\delta(x, r, \Lambda)$ is at most the product of the volume of a ball $4\pi/3$ with the number $|\Lambda(x, r + 1)|$ of balls intersecting $B(x, r)$. Hence

$$\operatorname{vol} B(x, r, \Lambda) \leq |\Lambda(x, r + 1)|4\pi/3. \tag{1.1}$$

Ordinary Proof Texts

- Combination of grammatically correct natural language and symbolic material.
- Texts are read and processed sentence by sentence.
- The language is often close to formal logical languages.
- Usually, every variable is typed.
- Statements have to be type-correct.

LEMMA 1.3. *If there exists a negligible* fcc*-compatible function* $A : \Lambda \to \mathbb{R}$ *for a saturated packing* Λ*, then there exists a constant* C *such that for all* $r \geq 1$ *and all* $x \in \mathbb{R}^3$,

$$\delta(x, r, \Lambda) \leq \pi/\sqrt{18} + C/r.$$

The constant C *depends on* Λ *only through the constant* C_1.

Proof. The numerator $\operatorname{vol} B(x, r, \Lambda)$ of $\delta(x, r, \Lambda)$ is at most the product of the volume of a ball $4\pi/3$ with the number $|\Lambda(x, r + 1)|$ of balls intersecting $B(x, r)$. Hence

$$\operatorname{vol} B(x, r, \Lambda) \leq |\Lambda(x, r + 1)|4\pi/3. \tag{1.1}$$

LEMMA 1.3. *If there exists a negligible* fcc*-compatible function* $A : \Lambda \to \mathbb{R}$ *for a saturated packing* Λ*, then there exists a constant* C *such that for all* $r \geq 1$ *and all* $x \in \mathbb{R}^3$,

$$\delta(x, r, \Lambda) \leq \pi/\sqrt{18} + C/r.$$

The constant C *depends on* Λ *only through the constant* C_1.

Proof. The numerator $\operatorname{vol} B(x, r, \Lambda)$ of $\delta(x, r, \Lambda)$ is at most the product of the volume of a ball $4\pi/3$ with the number $|\Lambda(x, r + 1)|$ of balls intersecting $B(x, r)$. Hence

$$\operatorname{vol} B(x, r, \Lambda) \leq |\Lambda(x, r + 1)|4\pi/3. \tag{1.1}$$

LEMMA 1.3. *If there exists a negligible* fcc*-compatible function* $A : \Lambda \to \mathbb{R}$ *for a saturated packing* Λ*, then there exists a constant* C *such that for all* $r \geq 1$ *and all* $x \in \mathbb{R}^3$*,*

$$\delta(x, r, \Lambda) \leq \pi/\sqrt{18} + C/r.$$

The constant C *depends on* Λ *only through the constant* C_1*.*

Proof. The numerator $\operatorname{vol} B(x, r, \Lambda)$ of $\delta(x, r, \Lambda)$ is at most the product of the volume of a ball $4\pi/3$ with the number $|\Lambda(x, r + 1)|$ of balls intersecting $B(x, r)$. Hence

$$\operatorname{vol} B(x, r, \Lambda) \leq |\Lambda(x, r + 1)|4\pi/3. \tag{1.1}$$

LEMMA 1.3. *If there exists a negligible* fcc*-compatible function* $A : \Lambda \to \mathbb{R}$ *for a saturated packing* Λ*, then there exists a constant* C *such that for all* $r \geq 1$ *and all* $x \in \mathbb{R}^3$*,*

$$\delta(x, r, \Lambda) \leq \pi/\sqrt{18} + C/r.$$

The constant C *depends on* Λ *only through the constant* C_1*.*

Proof. The numerator $\operatorname{vol} B(x, r, \Lambda)$ of $\delta(x, r, \Lambda)$ is at most the product of the volume of a ball $4\pi/3$ with the number $|\Lambda(x, r + 1)|$ of balls intersecting $B(x, r)$. Hence

$$\operatorname{vol} B(x, r, \Lambda) \leq |\Lambda(x, r + 1)|4\pi/3. \tag{1.1}$$

LEMMA 1.3. *If there exists a negligible* fcc*-compatible function* $A : \Lambda \to \mathbb{R}$ *for a saturated packing* Λ*, then there exists a constant* C *such that for all* $r \geq 1$ *and all* $x \in \mathbb{R}^3$*,*

$$\delta(x, r, \Lambda) \leq \pi/\sqrt{18} + C/r.$$

The constant C *depends on* Λ *only through the constant* C_1*.*

Proof. The numerator $\operatorname{vol} B(x, r, \Lambda)$ of $\delta(x, r, \Lambda)$ is at most the product of the volume of a ball $4\pi/3$ with the number $|\Lambda(x, r + 1)|$ of balls intersecting $B(x, r)$. Hence

$$\operatorname{vol} B(x, r, \Lambda) \leq |\Lambda(x, r + 1)|4\pi/3. \tag{1.1}$$

LEMMA 1.3. *If there exists a negligible* fcc*-compatible function* $A : \Lambda \to \mathbb{R}$ *for a saturated packing* Λ*, then there exists a constant* C *such that for all* $r \geq 1$ *and all* $x \in \mathbb{R}^3$*,*

$$\delta(x, r, \Lambda) \leq \pi/\sqrt{18} + C/r. \quad \text{<--- Type correct?}$$

The constant C *depends on* Λ *only through the constant* C_1*.*

Proof. The numerator $\operatorname{vol} B(x, r, \Lambda)$ of $\delta(x, r, \Lambda)$ is at most the product of the volume of a ball $4\pi/3$ with the number $|\Lambda(x, r + 1)|$ of balls intersecting $B(x, r)$. Hence

$$\operatorname{vol} B(x, r, \Lambda) \leq |\Lambda(x, r + 1)|4\pi/3. \tag{1.1}$$

Checking Ordinary Proof Texts

Texts are processed sentence by sentence:

- Reading
- Understanding
- Reasoning
  - Identifying *smaller* proof tasks
  - Discharging tasks by "high level" reasoning
  - Discharging tasks by "low level" derivation search, possibly with multiple attempts

Checking Ordinary Proof Texts

Texts are processed sentence by sentence:

- Reading [tokenizer, parser]
- Understanding [parser, typechecking]
- Reasoning [main process]
  - Identifying *smaller* proof tasks [tactics, heuristics]
  - Discharging tasks by "high level" reasoning [lookup with small tableau prover]
  - Discharging tasks by "low level" derivation search, possibly with multiple attempts [ATPs, term rewriting]

Checking Ordinary Proof Texts

Texts are processed sentence by sentence:

- Reading [tokenizer, parser]
- Understanding [parser, typechecking]
- Reasoning [main process]
  - Identifying *smaller* proof tasks [tactics, heuristics]
  - Discharging tasks by "high level" reasoning [lookup with small tableau prover]
  - Discharging tasks by "low level" derivation search, possibly with multiple attempts [ATPs, term rewriting]

Several feedback loops; results of previous reasoning can influence all components of further checking.

Evidence Algorithm

By the end of 1960s Academician V. Glushkov advanced a programme on investigating automated theorem proving, which was later called the Evidence Algorithm, EA (first mentioned in "Kibernetika", 2, 1970). V. Glushkov proposed to make investigation simultaneously into formalized languages for presenting mathematical texts in the form most appropriate for a user, formalization and evolutional development of computer-made proof step, EA information environment having an influence on a current evidence of computer-made proof step, and interactive man-assistant search of proof.

SAD system
Explanations
Download
Our Team

Since then, a lot of investigations were made in all the above spheres. Russian and English versions of the formalized mathematical languages were developed. Their syntactical analyzers were designed. At present time, a translator of the English-based Formal Theory Language (ForTheL) into the first-order language is implemented.

A sequential formalism was developed for construction of an efficient technique of proof search in an initial theory (without preliminary skolemization). A special approach was offered for applying definitions and auxiliary propositions that takes into account the neighbourhood of the proposition to be proved. Basing on this formalism, a first-order prover was implemented.

As a result, the System for Automated Deduction (SAD) appeared.

Theses of the EA programme promise to be helpful in attacking such problems as distributed automated theorem proving, verification of mathematical texts, remote training in mathematical disciplines, and construction of databases for mathematical theories.

THE SYSTEM FOR AUTOMATED DEDUCTION

Evidence Algorithm

SAD system | Explanations | Download | Our Team
Inference Search | Theorem Proving | Text Verification | TPTP Problems

[Help] [Examples]

```
[number/-s]

Signature NatSort.  A natural number is a notion.

Let i,j,k,l,m,n denote natural numbers.

Signature SortsC.  0 is a natural number.

Let x is nonzero stand for x != 0.

Signature SortsC.  1 is a nonzero natural number.

Let x is trivial stand for x = 0 \/ x = 1.
Let x is nontrivial stand for x != 0 and x != 1.

Signature SortsB.  m + n is a natural number.
Signature SortsB.  m * n is a natural number.

Axiom AddComm.   m + n = n + m.
Axiom AddAsso.  (m + n) + l = m + (n + l).
```

Parse | Verify with Moses | Verify with | SPASS 3.7 | Clear

Time limit (1–600 sec) 3 3 Verbosity level (0–6)

to Russian

Last modified: 3 Aug 2008

History of Naproche-SAD

1970 Victor Glushkov: *Evidence Algorithm*

1980 Victor Glushkov: *System for Automated Deduction (SAD)*

2008 Andrei Paskevich *SAD* (PhD project, Kiev, Paris)

2013 Marcos Cramer *Naproche* (PhD project, Natural Proof Checking, Bonn)

2018 Steffen Frerix *SAD* (Master project, Bonn)

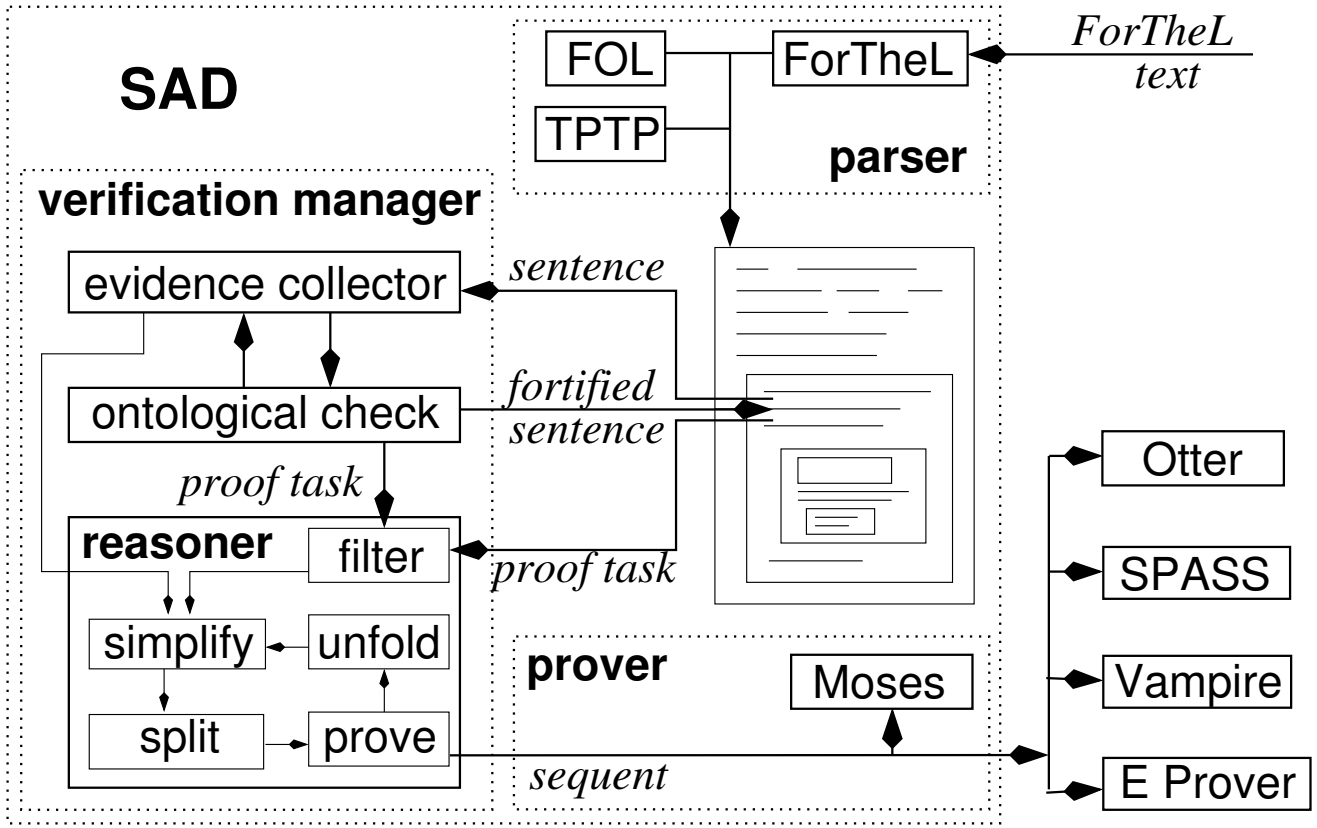
Further development of SAD within the Naproche programme:

improving and extending the SAD code; larger formalizations; LaTeX-typesetting

2018/19 with Makarius Wenzel: Embedding Naproche-SAD into the Isabelle PIDE

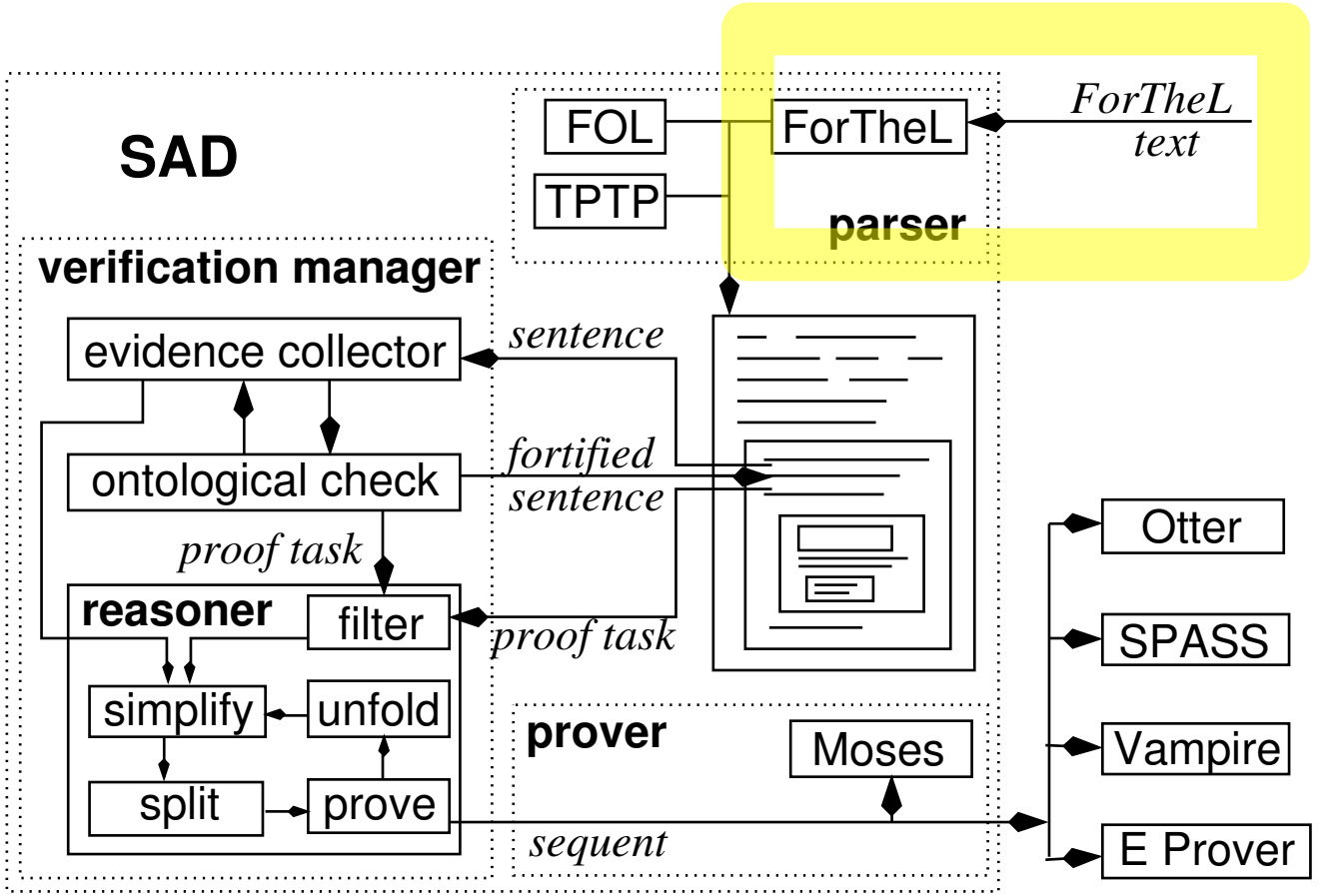
Demo

System for Automated Deduction



- **manager**: decompose input text into separate proof tasks
- **reasoner**: big steps of reasoning, heuristic proof methods
- **prover**: inference search in a sound and complete calculus

System for Automated Deduction



- **manager**: decompose input text into separate proof tasks
- **reasoner**: big steps of reasoning, heuristic proof methods
- **prover**: inference search in a sound and complete calculus

ForTheL (Formula Theory Language)

Signature. A *real number* is a notion.

Let x, y, z stand for real numbers.

Definition. $\mathbb{R}$ is the set of real numbers.

Signature. $x \cdot y$ is a real number.

Axiom. $x \cdot y = y \cdot x$.

Signature. A *positive integer* is a real number.

Theorem 7. (120a) *If $x \in \mathbb{R}$ and $y \in \mathbb{R}$ and $x > 0$ then there is a positive integer n such that*

$$n \cdot x > y.$$

ForTheL (Formula Theory Language)

ForTheL is a weakly typed language: variables belong to notions ($\sim$ types).

Functions and relations are typed: $x \cdot y$ is a real number; multiplication is of type $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$.

First-order logic is used internally. The parser translates types into type-guards:

a / the real number $x \Longrightarrow \mathsf{realNumber}(x)$

$x \cdot y$ is a real number $\Longrightarrow \mathsf{realNumber}(x) \wedge \mathsf{realNumber}(y) \to \mathsf{realNumber}(x \cdot y)$

A positive integer is a real number $\Longrightarrow \mathsf{posInt}(x) \to \mathsf{realNumber}(x)$

ForTheL (Formula Theory Language)

Function and relations can be introduced by symbolic patterns.

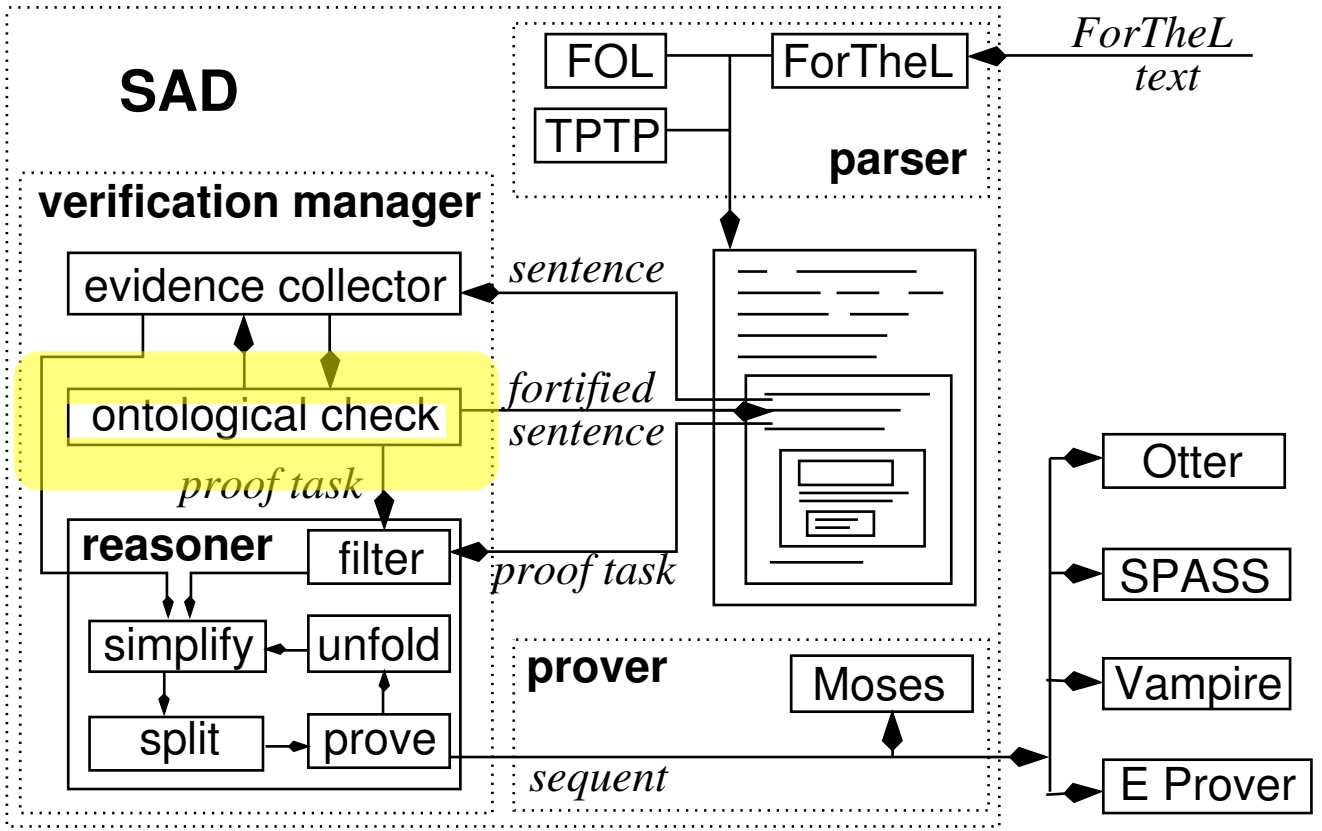
Signature. `\Prod{m}{n}{f} is a real number. Let the product of f from m to n stand for \Prod{m}{n}{f}.`

`\Prod{m}{n}{f}` complies with the LaTeX syntax for macros. With an appropriate macro definition it will, e.g., be typeset as

$$\prod_{i=m}^{n} f_i$$

It is easy to filter LaTeX-files to ForTheL-files. We are currently embedding ForTheL into LaTeX.

System for Automated Deduction



- **manager**: decompose input text into separate proof tasks
- **reasoner**: big steps of reasoning, heuristic proof methods
- **prover**: inference search in a sound and complete calculus

Notions: Ontological Checking

Naproche-SAD accepts statements only if they are type correct (ontologically correct)

Theorem 8. (120a) *If $x \in \mathbb{R}$ and $y \in \mathbb{R}$ and $x > 0$ then there is a positive integer n such that*

$$n \cdot x > y.$$

The subterm $n \cdot x$ has to be type correct at the place where it is stated. The variables are typed as realNumber(x) and posInt(n). Using the universal implication posInt$(x) \rightarrow$ realNumber(x) we can "prove" that realNumber(n). Hence $n \cdot x$ is legitimate.

Type correctness has to be proved *before* proving the existence of n.

Notions: Treatment of Undefinedness

Division can be introduced as:

Signature. Assume $y \neq 0$. Then $\frac{x}{y}$ is a real number.

The term $\frac{a}{b}$ is ontologically correct within a text if one can prove $b \neq 0$ from the statements and assumptions made *before* the position of the term.

This check corresponds to the "dynamic" correctness checking common in mathematics. It is not captured by static typechecking as in programming languages.

Notions can be defined by arbitrary first-order formulas

Definition. A *prime number* is a positive integer such that [first-order condition].

Notions can model ascending number systems:

Signature. A rational number is a real number.

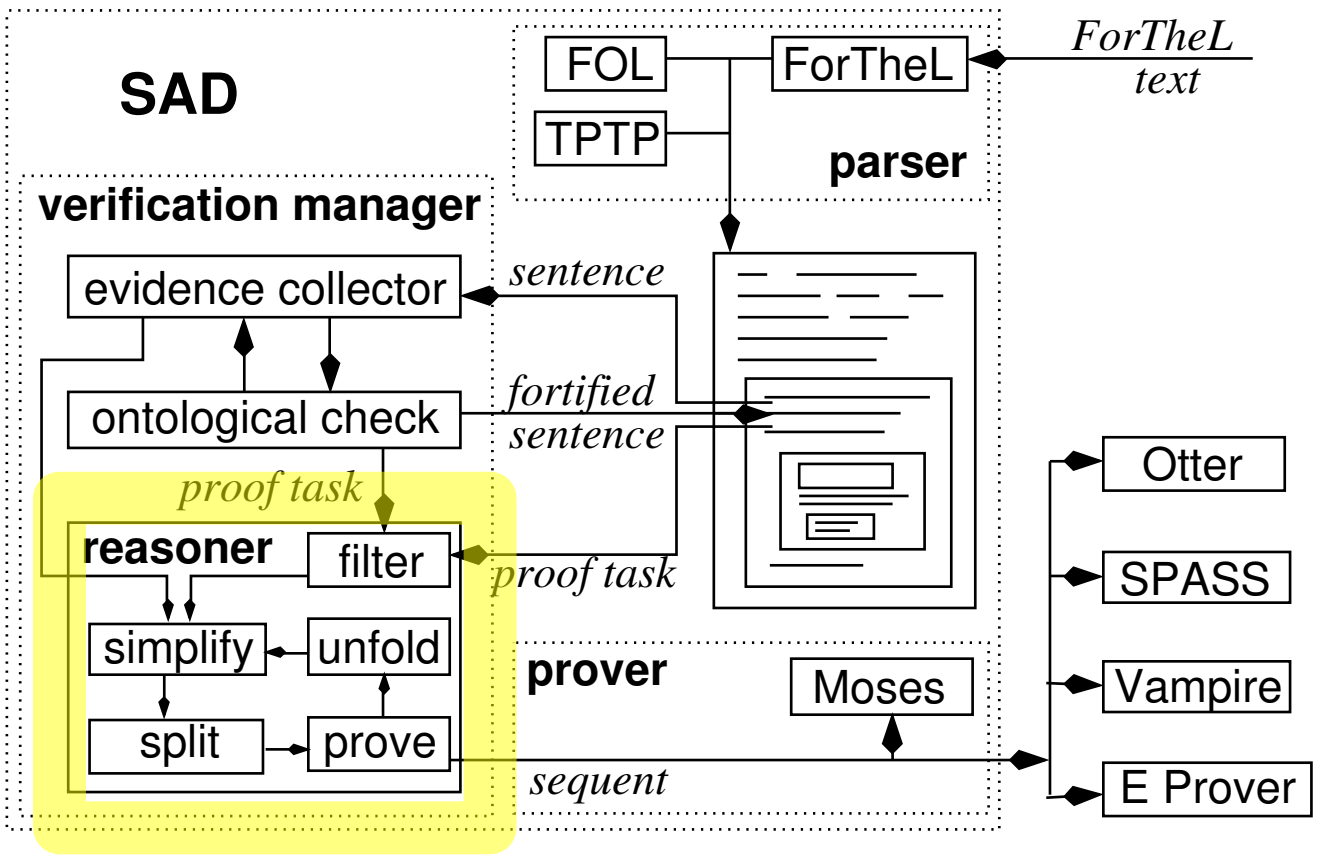
Signature. A positive integer is a rational number.

This introduces natural subtypes

$$\mathbb{N} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$$

Notions $\sim$ weak types (similar to the types in Mizar). Notions do not satisfy one of the standard type theories.

System for Automated Deduction



- **manager**: decompose input text into separate proof tasks
- **reasoner**: big steps of reasoning, heuristic proof methods
- **prover**: inference search in a sound and complete calculus

First-Order Proving

Break down ForTheL statements into smaller first-order proof obligations.

Split $\varphi \wedge \psi$ into proof of φ from the given assumptions Γ and a subsequent proof of ψ from Γ, φ.

($\varphi \wedge \psi$ is handled differently from $\psi \wedge \varphi$: $\varphi \wedge \psi$ may be accepted, but ψ not).

Goal-orientated proving: if the goal is $\varphi \to \psi$ then "Assume φ" reduces the goal to ψ.

Proof methods like induction or case distinctions can be stated in ForTheL and automatically generate proof obligations:

Induction: to prove $\forall x\, \varphi$ it suffices to prove $\forall y\, (y \prec x \to \varphi(y)) \to \varphi(x)$.

Cases: prove the goal under the case assumptions and prove that the assumptions exhaust all possibilities.

Term rewriting for proving equalities; this may generate further obligations like $b \neq 0$.

E Prover

Proof obligations that cannot be resolved trivially by the Reasoner are sent to E Prover.

The proof obligations generated by the Reasoner should be within reach of E Prover.

If E Prover fails, an obligation can be sent again with more premisses (unfolding of definitions).

Theorem 1. *(a) If $x \in \mathbb{R}$, $y \in \mathbb{R}$, and $x > 0$, then there is a positive integer n such that*

$$n\,x > y.$$

Proof. Let A be the set of all $n\,x$, where n runs through the positive integers. If (a) were false, then y would be an upper bound of A. But then A has a *least* upper bound in $\mathbb{R}$. Put $\alpha = \sup A$. Since $x > 0$, $\alpha - x < \alpha$, and $\alpha - x$ is not an upper bound of A. Hence $\alpha - x < m\,x$ for some positive integer m. But then $\alpha < (m+1)\,x \in A$, which is impossible, since α is an upper bound of A. □

Theorem 2. (120a) *If $x \in \mathbb{R}$ and $y \in \mathbb{R}$ and $x > 0$ then there is a positive integer n such that*

$$n \cdot x > y.$$

Proof. Define $X = \{n \cdot x |\ n \text{is a positive integer}\}$. Assume the contrary. Then y is an upper bound of X. Take a least upper bound α of X. $\alpha - x < \alpha$ and $\alpha - x$ is not an upper bound of X. Take an element z of X such that not $z \leq \alpha - x$. Take a positive integer m such that $z = m \cdot x$. Then $\alpha - x < m \cdot x$ (by 15b).

$$\alpha = (\alpha - x) + x < (m \cdot x) + x = (m+1) \cdot x.$$

$(m+1) \cdot x$ is an element of X. Contradiction. Indeed α is an upper bound of X. □

Formalizing Rudin in Naproche-SAD

Theorem 3. *If $x \in \mathbb{R}$, $y \in \mathbb{R}$, and $x < y$, then there exists a $p \in \mathbb{Q}$ such that $x < p < y$.*

Proof. Since $x < y$, we have $y - x > 0$, and (a) furnishes a positive integer n such that

$$m\,(y - x) > 1.$$

Apply (a) again, to obtain positive integers m_1 and m_2 such that $m_1 > n\,x$, $m_2 > -n\,x$. Then

$$-m_2 < n\,x < m_1.$$

Hence there is an integer m (with $-m_2 \leq m \leq m_1$) such that

$$m - 1 \leq n\,x < m.$$

If we combine these inequalities, we obtain

$$n\,x < m \leq 1 + n\,x < n\,y.$$

Since $n > 0$, it follows that

$$x < \frac{m}{n} < y.$$

This proves (b), with $p = m/n$. □

Theorem 4. (120b) *If $x \in \mathbb{R}$ and $y \in \mathbb{R}$ and $x < y$ then there exists a rational number p such that $x < p < y$.*

Proof. Assume $x < y$. We have $y - x > 0$. Take a positive integer n such that $n \cdot (y - x) > 1$ (by 120a). Take an integer m such that $m - 1 \leq n \cdot x < m$. Then

$$n \cdot x < m = (m - 1) + 1$$

$$\leq (n \cdot x) + 1 < (n \cdot x) + (n \cdot (y - x))$$

$$= n \cdot (x + (y - x)) = n \cdot y.$$

$m \leq (n \cdot x) + 1 < n \cdot y$. $\frac{m}{n} < \frac{n \cdot y}{n}$. Indeed $m < n \cdot y$ and $1/n > 0$. Then

$$x = \frac{n \cdot x}{n} < \frac{m}{n} < \frac{n \cdot y}{n} = y.$$

Let $p = \frac{m}{n}$. Then $p \in \mathbb{Q}$ and $x < p < y$. □

Naproche-SAD Formalizations

Further parts of Rudin's *Analysis*.

The Appendix of Kelley's *General Topology* about Kelley-Morse Set Theory.

Zermelo-Fraenkel Set Theory up to ordinals and cardinals

Small proofs from areas like elementary number theory, complex analysis, ...

ORDERED PAIRS: RELATIONS

This section is devoted to the properties of ordered pairs and relations. The crucial property for ordered pairs is theorem 55: if x and y are sets, then $(x,y) = (u,v)$ iff $x = u$ and $y = v$.

48 DEFINITION $(x,y) = \{\{x\}\{xy\}\}$.

The class (x,y) is an *ordered pair.*

49 THEOREM *(x,y) is a set if and only if x is a set and y is a set; if (x,y) is not a set, then $(x,y) = \mathfrak{U}$.*

50 THEOREM *If x and y are sets, then* $\bigcup(x,y) = \{xy\}$, $\bigcap(x,y) = \{x\}$, $\bigcup\bigcap(x,y) = x$, $\bigcap\bigcap(x,y) = x$, $\bigcup\bigcup(x,y) = x \cup y$ *and* $\bigcap\bigcup(x,y) = x \cap y$.

If either x or y is not a set, then $\bigcup\bigcap(x,y) = 0$, $\bigcap\bigcap(x,y) = \mathfrak{U}$, $\bigcup\bigcup(x,y) = \mathfrak{U}$, *and* $\bigcap\bigcup(x,y) = 0$.

51 DEFINITION 1^{st} coord $z = \bigcap\bigcap z$.

52 DEFINITION 2^{nd} coord $z = (\bigcap\bigcup z) \cup ((\bigcup\bigcup z) \sim \bigcup\bigcap z)$.

These definitions will be used, with one exception, only in the case where z is an ordered pair. The *first coordinate* of z is 1^{st} coord z and the *second coordinate* of z is 2^{nd} coord z.

53 THEOREM 2^{nd} coord $\mathfrak{U} = \mathfrak{U}$.

54 THEOREM *If x and y are sets* 1^{st} coord $(x,y) = x$ *and* 2^{nd} coord $(x,y) = y$. *If either of x and y is not a set, then* 1^{st} coord $(x,y) = \mathfrak{U}$ *and* 2^{nd} coord $(x,y) = \mathfrak{U}$.

PROOF If x and y are sets, then the equality for 1^{st} *coord* is immediate from 50 and 51. The equality for 2^{nd} *coord* reduces to showing that $y = (x \cap y) \cup ((x \cup y) \sim x)$, by 50 and 52. It is straightforward to see that $(x \cup y) \sim x = y \sim x$ and by the distributive law $(y \cap x) \cup (y \cap \sim x)$ is $y \cap (x \cup \sim x) = y \cap \mathfrak{U} = y$. If either x or y is not a set, then, using 50 it is easy to compute 1^{st} *coord* (x,y) and 2^{nd} *coord* (x,y). ▮

55 THEOREM *If x and y are sets and $(x,y) = (u,v)$, then $x = u$ and $y = v$.*

ELEMENTARY SET THEORY

An SAD3 Formalisation of the Appendix of "General Topology" by John L. Kelley

October 26, 2018

0.1 The Classification Axiom Scheme

Let $a, b, c, d, e, r, s, t, x, y, z$ stand for *classes*.
Let $a \in x$ stand for a is an *element* of x.

Axiom (I)**.** *For each x for each y $x = y$ iff for each z $z \in x$ iff $z \in y$.*

[set/-s]

Definition (1)**.** *A* set *is a class x such that for some y $x \in y$.*

0.2 Elementary Algebra of Classes

Definition (2)**.** $x \cup y = \{set\ u \mid u \in x\ or\ u \in y\}$.

Definition (3)**.** $x \cap y = \{set\ u \mid u \in x\ and\ u \in y\}$.

Let the *union* of x and y stand for $x \cup y$. Let the *intersection* of x and y stand for $x \cap y$.

Theorem (4a)**.** $z \in x \cup y$ *iff* $z \in x$ *or* $z \in y$.

Theorem (4b)**.** $z \in x \cap y$ *iff* $z \in x$ *and* $z \in y$.

Theorem (5a)**.** $x \cup x = x$.

Theorem (5b)**.** $x \cap x = x$.

Theorem (6a)**.** $x \cup y = y \cup x$.

Theorem (6b)**.** $x \cap y = y \cap x$.

Theorem (7a)**.** $(x \cup y) \cup z = x \cup (y \cup z)$.

Theorem (7b)**.** $(x \cap y) \cap z = x \cap (y \cap z)$.

Theorem (8a)**.** $x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$.

Theorem (8b)**.** $x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$.

Let $a \notin b$ stand for a is not an element of b.

Definition (10)**.** $\sim x = \{set\ u \mid u \notin x\}$. *Let the* complement *of* x *stand for* $\sim x$.

Theorem (11)**.** $\sim (\sim x) = x$.

Theorem (12a)**.** $\sim (x \cup y) = (\sim x) \cap (\sim y)$.

Theorem (12b)**.** $\sim (x \cap y) = (\sim x) \cup (\sim y)$.

Definition (13)**.** $x \sim y = x \cap (\sim y)$.

Theorem (14)**.** $x \cap (y \sim z) = (x \cap y) \sim z$.

Definition (15)**.** $0 = \{set\ u \mid u \neq u\}$. *Let the* void class *stand for* 0. *Let* zero *stand for* 0.

Theorem (16)**.** $x \notin 0$.

Theorem (17a)**.** $0 \cup x = x$.

Theorem (17b)**.** $0 \cap x = 0$.

Definition (18)**.** $\mathcal{U} = \{set\ u \mid u = u\}$. *Let the* universe *stand for* $\mathcal{U}$.

Theorem (19)**.** $x \in \mathcal{U}$ *iff* x *is a set.*

Theorem (20a)**.** $x \cup \mathcal{U} = \mathcal{U}$.

Theorem (20b)**.** $x \cap \mathcal{U} = x$.

Theorem (21a)**.** $\sim 0 = \mathcal{U}$.

Theorem (21b)**.** $\sim \mathcal{U} = 0$.

Definition (22)**.** $\bigcap x = \{setu \mid$ *for each* y *if* $y \in x$ *then* $u \in y\}$. *Let the* intersection *of* x *stand for* $\bigcap x$.

Definition (23)**.** $\bigcup x = \{setu \mid$ *for some* $y(y \in x$ *and* $u \in y)\}$. *Let the* union *of* x *stand for* $\bigcup x$.

Theorem (24a)**.** $\bigcap 0 = \mathcal{U}$.

Theorem (24b)**.** $\bigcup 0 = 0$.

Definition (25)**.** *A* subclass *of* y *is a class* x *such that each element of* x *is an element of* y. *Let* $x \subset y$ *stand for* x *is a subclass of* y. *Let* x *is* contained *in* y *stand for* $x \subset y$.

Proposition. $0 \subset 0$ *and* $0 \notin 0$.

Theorem (26a)**.** $0 \subset x$.

Theorem (26b)**.** $x \subset \mathcal{U}$.

Theorem (27)**.** $x = y$ *iff* $x \subset y$ *and* $y \subset x$.

Theorem (28)**.** *If* $x \subset y$ *and* $y \subset z$ *then* $x \subset z$.

Theorem (29)**.** $x \subset y$ *iff* $x \cup y = y$.

Theorem (30)**.** $x \subset y$ *iff* $x \cap y = x$.

Theorem (31a)**.** *If* $x \subset y$ *then* $\bigcup x \subset \bigcup y$.

Theorem (31a)**.** *If* $x \subset y$ *then* $\bigcap y \subset \bigcap x$.

Theorem (32a)**.** *If* $x \in y$ *then* $x \subset \bigcup y$.

Theorem (32b)**.** *If* $x \in y$ *then* $\bigcap y \subset x$.

0.3 Existence of Sets

Axiom (III)**.** *If* x *is a set then there is a set* y *such that for each* z *if* $z \subset x$ *then* $z \in y$.

Theorem (33)**.** *If* x *is a set and* $z \subset x$ *then* z *is a set.*

Theorem (34a)**.** $0 = \bigcap \mathcal{U}$.

Theorem (34b)**.** $\mathcal{U} = \bigcup \mathcal{U}$.

Theorem (35)**.** *If* $x \neq 0$ *then* $\bigcap x$ *is a set.*

Definition (36)**.** $2^x = \{set\ y \mid y \subset x\}$.

Theorem (37)**.** $\mathcal{U} = 2^{\mathcal{U}}$.

Theorem (38a)**.** *If* x *is a set then* 2^x *is a set.*

Proof. Let x be a set. Take a set y such that for each z if $z \subset x$ then $z \in y$ (by III). Then $2^x \subset y$. □

Theorem (38b)**.** *If* x *is a set then* $y \subset x$ *iff* $y \in 2^x$.

Definition. $\mathrm{R} = \{set\ x \mid x \notin x\}$.

Lemma. R *is not a set.*

Theorem (39)**.** $\mathcal{U}$ *is not a set.*

Definition (40). $\{x\} = \{set\ z \mid \ if\ x \in \mathcal{U}\ then\ z = x\}$. *Let the* singleton *of* x *stand for* $\{x\}$.

Theorem (41). *If* x *is a set then for each* y $y \in \{x\}$ *iff* $y = x$.

Theorem (42). *If* x *is a set then* $\{x\}$ *is a set.*

Proof. Let x be a set. Then $\{x\} \subset 2^x$. 2^x is a class. □

Theorem (43). $\{x\} = \mathcal{U}$ *iff* x *is not a set.*

Theorem (44a). *If* x *is a set then* $\bigcap\{x\} = x$.

Theorem (44b). *If* x *is a set then* $\bigcup\{x\} = x$.

Theorem (44c). *If* x *is not a set then* $\bigcap\{x\} = 0$.

Theorem (44d). *If* x *is not a set then* $\bigcup\{x\} = \mathcal{U}$.

Axiom (IV). *If* x *is a set and* y *is a set then* $x \cup y$ *is a set.*

Definition (45). $\{x, y\} = \{x\} \cup \{y\}$. *Let the* unordered pair *of* x *and* y *stand for* $\{x, y\}$.

Theorem (46a). *If* x *is a set and* y *is a set then* $\{x, y\}$ *is a set.*

Theorem (46b). *If* x *is a set and* y *is a set then* $z \in \{x, y\}$ *iff* $z = x$ *or* $z = y$.

Theorem (46c). $\{x, y\} = \mathcal{U}$ *iff* x *is not a set or* y *is not a set.*

Theorem (47a). *If* x, y *are sets then* $\bigcap\{x, y\} = x \cap y$.

Theorem (47b). *If* x, y *are sets then* $\bigcup\{x, y\} = x \cup y$.

Proof. Let x, y be sets. $\bigcup\{x, y\} \subset x \cup y$. $x \cup y \subset \bigcup\{x, y\}$. □

Theorem (47c). *If* x *is not a set or* y *is not a set then* $\bigcap\{x, y\} = 0$.

Theorem (47d). *If* x *is not a set or* y *is not a set then* $\bigcup\{x, y\} = \mathcal{U}$.

0.4 Ordered Pairs: Relations

Definition (48). $(x, y) = \{\{x\}, \{x, y\}\}$. *Let the* ordered pair *of* x *and* y *stand for* (x, y).

Theorem (49a). (x, y) *is a set iff* x *is a set and* y *is a set.*

Theorem (49b). *If* (x, y) *is not a set then* $(x, y) = \mathcal{U}$.

Theorem (50). *If* x *and* y *are sets then* $\bigcup(x, y) = \{x, y\}$ *and* $\bigcap(x, y) = \{x\}$ *and* $\bigcup\bigcap(x, y) = x$ *and* $\bigcap\bigcap(x, y) = x$ *and* $\bigcup\bigcup(x, y) = x \cup y$ *and* $\bigcap\bigcup(x, y) = x \cap y$.

Theorem. *If x is not a set or y is not a set then $\bigcup\bigcap(x,y) = 0$ and $\bigcap\bigcap(x,y) = \mathcal{U}$ and $\bigcup\bigcup(x,y) = \mathcal{U}$ and $\bigcap\bigcup(x,y) = 0$.*

Definition (51)**.** $1^{st}z = \bigcap\bigcap z$. *Let the* first coordinate *of z stand for $1^{st}z$.*

Definition (52)**.** $2^{nd}z = (\bigcap\bigcup z) \cup ((\bigcup\bigcup z) \sim \bigcup\bigcap z)$. *Let the* second coordinate *of z stand for $2^{nd}z$.*

Theorem (53)**.** $2^{nd}\mathcal{U} = \mathcal{U}$.

Theorem (54a)**.** *If x and y are sets then $1^{st}(x,y) = x$.*

Theorem (54b)**.** *If x and y are sets then $2^{nd}(x,y) = y$.*

Proof. Let x and y be sets. $2^{nd}(x,y) = (\bigcap\bigcup(x,y)) \cup ((\bigcup\bigcup(x,y)) \sim \bigcup\bigcap(x,y)) = (x \cap y) \cup ((x \cup y) \sim x) = y$. □

Theorem (54c)**.** *If x is not a set or y is not a set then $1^{st}(x,y) = \mathcal{U}$ and $2^{nd}(x,y) = \mathcal{U}$.*

Theorem (55)**.** *If x and y are sets and $(x,y) = (r,s)$ then $x = r$ and $y = s$.*

Demo

Discussion

- Is natural language formal mathematics viable? Is it possible to reach the efficiency and coverage of other ITPs? How would one organize libraries of natural language formal mathematics?
- Can natural language formal mathematics help the acceptance and use of formal mathematics in the mathematical community?
- Can one develop ForTheL-like languages and interfaces for standard ITPs?
- Research the language and linguistics of mathematics.
- Develop logics that combine first-order set theory and type theory.

Thank you!