Formalize, Naturally!

by Peter Koepke

University of Bonn

Talk at Formalize!(?), 16 January 2021, Zurich/online

From the announcement of Formalize!(?):

... The derivation indicator view says that all proofs stand in some relation to a derivation, i.e. a mechanically checkable syntactical objects following fixed rules, that would not have any gaps.

...

Interactive and automated theorem provers promise to make the construction of a justification without any gaps feasible for complex mathematics.

...

Is this promise justified? Will the future of mathematical practice shift to more formal mathematics? Should it? ...

Jody Azzouni, 2004: *The Derivation-Indicator View of Mathematical Practice*

… I take a proof to indicate an 'underlying' derivation. How *proofs do this is a somewhat complicated matter which I'll say more about shortly. …*

.

Jody Azzouni, 2004: *The Derivation-Indicator View of Mathematical Practice*

... I take a proof to indicate an 'underlying' derivation. How *proofs do this is a somewhat complicated matter which I'll say more about shortly. ...*

$\mathbb{N}$aproche:

- Natural proof assistant
- partial implementation of the derivation indicator view
- perfectly natural proof text is transformed into a formal derivation in some proof calculus
- natural language processing, logical transformations and automatic theorem proving

Martin Aigner, Günter M. Ziegler, 1998: *Proofs from THE BOOK*

Euclid's Proof. For any finite set $\{p_1, \ldots, p_r\}$ of primes, consider the number $n = p_1 p_2 \cdots p_r + 1$. This n has a prime divisor p. Put p is not one of the p_i: otherwise p would be a divisor of n and of the product $p_1 p_2 \cdots p_r$, and thus also of the difference $n - p_1 p_2 \cdots p_r = 1$, which is impossible. So a finite set $\{p_1, \ldots, p_r\}$ cannot be the collection of *all* prime numbers. $\square$

Aigner and Ziegler:

Euclid's Proof. For any finite set $\{p_1, \dots, p_r\}$ of primes,

consider the number $n = p_1 p_2 \cdots p_r + 1$.

This n has a prime divisor p.

But p is not one of the p_i:

otherwise

p would be a divisor of n and of the product $p_1 p_2 \cdots p_r$,

and thus also of the difference $n - p_1 p_2 \cdots p_r = 1$,

which is impossible.

So a finite set $\{p_1, \dots, p_r\}$ cannot be the collection of *all* prime numbers. □

ForTheL text, accepted by $\mathbb{N}$aproche:

Theorem 1. (Euclid) $\mathbb{P}$ *is infinite.*

Proof. Assume that r is a natural number and p is a sequence of length r and $\{p_1, \dots, p_r\}$ is a subset of $\mathbb{P}$.

Consider $n = p_1 \cdots p_r + 1$.

Take a prime divisor q of n.

Let us show that q is not equal to p_i for all i such that $1 \leq i$ and $i \leq r$.

Assume the contrary. Take i such that $1 \leq i$ and $i \leq r$ and $q = p_i$.

q is a divisor of n and q is a divisor of $p_1 \cdots p_r$ (by 1).

Thus q divides 1.

Contradiction. qed.

Hence $\{p_1, \dots, p_r\}$ is not the class of prime natural numbers. □

The Natural Proof Assistant $\mathbb{N}$aproche

- (Original mathematical text)
- Input text in ForTheL (Formula Theory Language)
- Natural language and familiar symbolic terms
- LaTeX format allows mathematical typesetting
- Natural language processing into first-order text
- Logical processing cuts up text into proof tasks
- Proof tasks are given to Automatic Theorem Prover (eprover)
- Eprover searches for superposition proofs
- (Eprover outputs derivations in superposition calculus)
- (Partial derivations can be combined into a complete derivation of the original text)

$\mathbb{N}$aproche (Natural Proof Checking)

- Evidence Algorithm (Victor Glushkov, $\sim$ 1970), ForTheL Input Language (Konstantin Vershinin, $\sim$ 1975), System for Automated Deduction (Andrei Paskevich, $\sim$ 2007)
- Naproche Project (Bernhard Schröder, PK, $\sim$ 2002), (Old) Naproche System (Marcos Cramer, 2013)
- Naproche-SAD (Steffen Frerix, PK, 2018)
- Isabelle-Naproche (Steffen Frerix, Makarius Wenzel, PK, 2019): `https://files.sketis.net/Isabelle_Naproche-20190611/`

The Natural Proof Assistant $\mathbb{N}$aproche

- (Original mathematical text)
- Input text in ForTheL (Formula Theory Language)
- Natural language and familiar symbolic terms
- LaTeX format allows mathematical typesetting
- Natural language processing into first-order text
- Logical processing cuts up text into proof tasks (goals)
- Proof tasks are given to Automatic Theorem Prover (eprover)
- Eprover searches for superposition proofs
- (Eprover outputs derivations in superposition calculus)
- (Partial derivations can be combined into a complete derivation of the original text)

Input

```
\begin{theorem}[Euclid]
$\Primes$ is infinite.
\end{theorem}
\begin{proof}
Assume that $r$ is a natural number and $p$ is a sequence of length $r$
and $\Set{p}{1}{r}$ is a subset of $\Primes$.
Consider $n=\Prod{p}{1}{r}+1$.
Take a prime divisor $q$ of $n$.
Let us show that $q$ is not equal to $\val{p}{i}$ for all $i$ such that
$1 \leq i$ and $i \leq r$.
Assume the contrary.
Take $i$ such that $1 \leq i$ and $i \leq r$ and $q=\val{p}{i}$.
$q$ is a divisor of $n$ and $q$ is a divisor of $\Prod{p}{1}{r}$ (by
1).
Thus $q$ divides $1$. Contradiction. qed.
Hence $\Set{p}{1}{r}$ is not the class of prime natural numbers.
\end{proof}
```

Self-Contained Axiomatic Text (2-3 Pages LaTeX Printout)

[dump on]

Let $x \neq y$ stand for x is nonequal to y.

[synonym number/-s] [synonym divide/-s] [synonym set/-s] [synonym element/-s] [synonym belong/-s] [synonym subset/-s]

1 Natural Numbers

Signature 2. *A natural number is a notion.*

Let i, k, l, m, n, p, q, r denote natural numbers.

Signature 3. *0 is a natural number.*

Signature 4. *1 is a nonzero natural number.*

...

Signature 5. *$m + n$ is a natural number.*

Signature 6. *$m * n$ is a natural number.*

Axiom 7. *$m + n = n + m$.*

Axiom 8. *$(m + n) + l = m + (n + l)$.*

2 The Natural Order

Definition 9. *$m \leq n$ iff there exists a natural number l such that $m + l = n$.*

3 Division

4 Primes

Definition 10. *n is prime iff n is nontrivial and for every divisor m of n $m = 1$ or $m = n$.*

Lemma 11. *Every nontrivial k has a prime divisor.*

Proof. Proof by induction. □

5 Sets

Definition 12. *$\mathbb{N}$ is the class of natural numbers.*

6 Sequences and Products

Signature 13. *Let F be a sequence of length n such that $\{F_1, \ldots, F_n\} \subseteq \mathbb{N}$. $F_1 \cdots F_n$ is a nonzero natural number.*

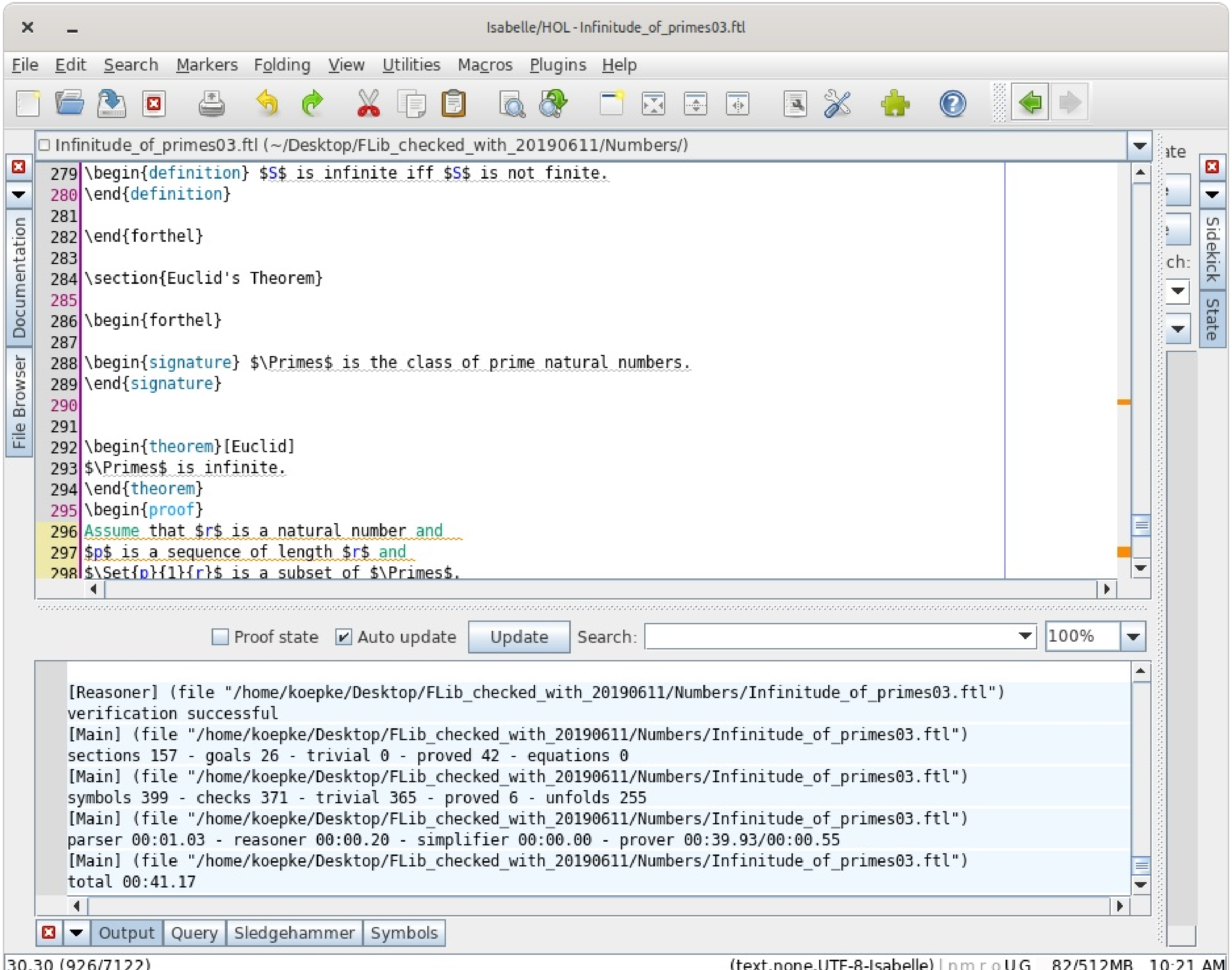
7 Finite and Infinite Sets

8 Euclid's Theorem

Signature 14. *$\mathbb{P}$ is the class of prime natural numbers.*

Theorem 15. *[Euclid] $\mathbb{P}$ is infinite.*

Proof. ... □


Isabelle/HOL - Infinitude_of_primes03.ftl
File Edit Search Markers Folding View Utilities Macros Plugins Help
Infinitude_of_primes03.ftl (~/Desktop/FLib_checked_with_20190611/Numbers/)
\begin{definition} S is infinite iff S is not finite.
\end{definition}
\end{forthel}
\section{Euclid's Theorem}
\begin{forthel}
\begin{signature} $\Primes$ is the class of prime natural numbers.
\end{signature}
\begin{theorem}[Euclid]
$\Primes$ is infinite.
\end{theorem}
\begin{proof}
Assume that r is a natural number and
p is a sequence of length r and
$\Set{p}{1}{r}$ is a subset of $\Primes$.
Documentation
File Browser
Sidekick
State
Proof state
Auto update
Update
Search:
100%
[Reasoner] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/Infinitude_of_primes03.ftl")
verification successful
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/Infinitude_of_primes03.ftl")
sections 157 - goals 26 - trivial 0 - proved 42 - equations 0
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/Infinitude_of_primes03.ftl")
symbols 399 - checks 371 - trivial 365 - proved 6 - unfolds 255
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/Infinitude_of_primes03.ftl")
parser 00:01.03 - reasoner 00:00.20 - simplifier 00:00.00 - prover 00:39.93/00:00.55
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/Infinitude_of_primes03.ftl")
total 00:41.17
Output Query Sledgehammer Symbols
30,30 (926/7122)
(text,none,UTF-8-Isabelle) | n m r o U G 82/512MB 10:21 AM

First-Order Translation

```
koepke@dell:~/TEST/Naproche-SAD$ stack exec Naproche-SAD -- -T ~/Desktop/
FLib_checked_with_20190611/Numbers/Infinitude_of_primes03.ftl
......
hypothesis.
  assume forall v0 ((HeadTerm :: v0 = Primes) implies (aClass(v0) and forall v1
(aElementOf(v1,v0) iff (aNaturalNumber(v1) and isPrime(v1))))).

conjecture Euclid.
  isInfinite(Primes).
  proof.
    assume ((aNaturalNumber(r) and aSequenceOfLength(p,r)) and aSubsetOf(Set{p}{1}{r},
Primes)).
    n = Prod{p}{1}{r}+1.
    ((aNaturalNumber(q) and doDivides(q,n)) and isPrime(q)).
    forall v0 ((aNaturalNumber(v0) and (doLeq(1,v0) and doLeq(v0,r))) implies not q =
sdlbdtrb).
    proof.
      assume not thesis.
      (aNaturalNumber(i) and ((doLeq(1,i) and doLeq(i,r)) and q = sdlbdtrb)).
      ((aNaturalNumber(q) and doDivides(q,n)) and (aNaturalNumber(q) and doDivides(q,
Prod{p}{1}{r}))).
      doDivides(q,1).
      contradiction.
    qed.
    not (aClass(Set{p}{1}{r}) and forall v0 (aElementOf(v0,Set{p}{1}{r}) iff
(aNaturalNumber(v0) and isPrime(v0)))).
  qed.
```

Proof Goals, Sent to Eprover in TPTP Format

```
...
...
[Reasoner] (line 293 of "/home/koepke/Desktop/
FLib_checked_with_20190611/Numbers/Infinitude_of_primes03.ftl")
goal:  Primes is infinite.
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
fof(m_,hypothesis,$true).
fof(m_,hypothesis,aNaturalNumber(sz0)).
fof(m_,hypothesis,(aNaturalNumber(sz1) & ( ~ (sz1 = sz0)))).
fof(m_,hypothesis,( ! [W0] : ( ! [W1] : ((aNaturalNumber(W0) &
aNaturalNumber(W1)) => aNaturalNumber(sdtpldt(W0,W1)))))).
fof(m_,hypothesis,( ! [W0] : ( ! [W1] : ((aNaturalNumber(W0) &
aNaturalNumber(W1)) => aNaturalNumber(sdtasdt(W0,W1)))))).
fof(m_,hypothesis,( ! [W0] : ( ! [W1] : ((aNaturalNumber(W0) &
aNaturalNumber(W1)) => (sdtpldt(W0,W1) = sdtpldt(W1,W0)))))).
...
...
```

Proof Goals (continued)

```
...
...
fof(m_,hypothesis,( ! [W0] : (aClass(W0) => (isInfinite(W0) <=> ( ~
isFinite(W0)))))).
fof(m_,hypothesis,(aClass(szPzrzizmzezs) & ( ! [W0] : (aElementOf(W0,
szPzrzizmzezs) <=> (aNaturalNumber(W0) & isPrime(W0)))))).
fof(m__,conjecture,(( ! [W0] : ( ! [W1] : (((aNaturalNumber(W1) &
aSequenceOfLength(W0,W1)) & aSubsetOf(szSzeztlcdtrclcz1rclcdtrc(W0,
W1),szPzrzizmzezs)) => ( ? [W2] : ((W2 =
sdtpldt(szPzrzozdlcdtrclcz1rclcdtrc(W0,W1),sz1)) & ( ? [W3] :
(((aNaturalNumber(W3) & doDivides(W3,W2)) & isPrime(W3)) & (( !
[W4] : ((aNaturalNumber(W4) & (doLeq(sz1,W4) & doLeq(W4,W1))) =>
( ~ (W3 = ssdlbdtrb(W0,W4))))) & ( ~ ( ! [W4] : (aElementOf(W4,
szSzeztlcdtrclcz1rclcdtrc(W0,W1)) <=> (aNaturalNumber(W4) &
isPrime(W4)))))))))))))) => isInfinite(szPzrzizmzezs))).
```

Proof Found!

```
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
[eprover] # No SInE strategy applied
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
[eprover] # Auto-Mode selected heuristic
G_E___208_C18_F1_SE_CS_SP_PS_S4Y
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
[eprover] # and selection function SelectMaxLComplexAPPNTNp.
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
[eprover] #
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
[eprover] # Presaturation interreduction done
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
[eprover] # Proof found!
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
[eprover] # SZS status Theorem
```

Derivations generated by Eprover

```
[Main] (file "/home/koepke/Desktop/FLib_checked_with_20190611/Numbers/
Infinitude_of_primes03.ftl")
...
[eprover] cnf(c_0_47,hypothesis,(aElementOf(X1,
szPzrzizmzezs)|~isPrime(X1)|~aNaturalNumber(X1)),
inference(split_conjunct,[status(thm)],[c_0_39])).
[eprover] cnf(c_0_48,plain,(isPrime(esk29_2(X2,X1))|~epred4_2(X1,X2)),
inference(split_conjunct,[status(thm)],[c_0_22])).
[eprover] cnf(c_0_49,plain,(aNaturalNumber(esk29_2(X2,
X1))|~epred4_2(X1,X2)), inference(split_conjunct,[status(thm)],
[c_0_22])).
[eprover] cnf(c_0_50,negated_conjecture,(~aElementOf(esk29_2(esk14_0,
esk15_0),szPzrzizmzezs)), inference(cn,[status(thm)],[inference(rw,
[status(thm)],[c_0_45, c_0_46])])).
[eprover] cnf(c_0_51,hypothesis,(aElementOf(esk29_2(X1,X2),
szPzrzizmzezs)|~epred4_2(X2,X1)), inference(csr,[status(thm)],
[inference(spm,[status(thm)],[c_0_47, c_0_48]), c_0_49])).
[eprover] cnf(c_0_52,negated_conjecture,($false), inference(cn,
[status(thm)],[inference(rw,[status(thm)],[inference(spm,[status(thm)],
[c_0_50, c_0_51]), c_0_46])]), ['proof']).
[eprover] # SZS output end CNFRefutation
```

Refutation Proof

$\ldots$

$R(\mathrm{sk}_{15}, \mathrm{sk}_{14})$

$X_1 \in \mathbb{P} \vee \neg \mathrm{prime}(X_1) \vee \neg \mathrm{natural}(X_1)$

$\mathrm{prime}(\mathrm{sk}_{29}(X_2, X_1)) \vee \neg R(X_1, X_2)$

$\mathrm{natural}(\mathrm{sk}_{29}(X_2, X_1)) \vee \neg R(X_1, X_2)$

$\mathrm{sk}_{29}(\mathrm{sk}_{14}, \mathrm{sk}_{15}) \notin \mathbb{P}$

$\mathrm{sk}_{29}(X_1, X_2) \in \mathbb{P} \vee \neg R(X_2, X_1)$

$\bot$

Statistics of Checking Euclid

- Text is checked in $\leqslant$1 minute
- 42 proofs found by eprover
- These proofs have between 2 and 106 clauses
- Project: combine such proofs to (superposition) derivations of complete texts

- Proof-checked ForTheL text $\Longrightarrow$ derivation
- Is that the derivation that the original proof in Aigner-Ziegler or Euclid “indicated”?
- ...

The Future of Formal Mathematics

- Interactive and automated theorem provers already allow the construction of justifications without any gaps for complex mathematics
- Mathematical practice will shift towards formal mathematics
- A decisive factor for the acceptance and momentum of this shift will be the ease and naturality of the interaction with the software
- Natural input languages are possible and will be provided for several interactive theorem provers
- Natural formal mathematics will require substantial further research and development

Thank You!