Beautiful Formalizations in Isabelle/Naproche

Adrian De Lon$^{[0000-0002-2697-7253]}$, Peter Koepke$^{[0000-0002-2266-134X]}$, Anton Lorenzen, Adrian Marti, Marcel Schütz, and Erik Sturzenhecker

University of Bonn, Germany, https://www.math.uni-bonn.de/ag/logik/

Abstract. We present short example formalizations of basic theorems from number theory, set theory, and lattice theory which ship with the new $\mathbb{N}$aproche component in Isabelle 2021. The natural proof assistant $\mathbb{N}$aproche accepts input texts in the mathematical controlled natural language ForTheL. Some ForTheL texts that proof-check in $\mathbb{N}$aproche come close to ordinary mathematical writing. The formalization examples demonstrate the potential to write mathematics in a natural yet completely formal language and to delegate tedious organisatorial details and obvious proof steps to strong automated theorem proving so that mathematical ideas and the "beauty" of proofs become visible.

1 Introduction

In informal mathematical discourse one frequently encounters appraisals of theorems and proofs as "intuitive", "elegant", "interesting", "simple", or indeed "beautiful". Following Paul Erdős, perfect proofs by these criteria would be entered in God's BOOK of proofs [12]. Although mathematicians often agree about the beauty of particular proofs, mathematical beauty in principle appears as elusive as the concept of beauty in general. Discussions of beauty by eminent mathematicians exhibit a spectrum of ad hoc theories and personal opinions (see, e.g., [18], [22]). A popular view that fits the perspective of this paper is expressed in [13, p. 22], (but observe [6]):

> *Mathematicians have customarily regarded a proof as beautiful if it conformed to the classical ideals of brevity and simplicity.*

This explains that completely formal proofs as studied in formal mathematics are widely viewed as being the opposite of "beautiful". Reuben Hersh [5, p. 52], writes:

> *We prefer a beautiful proof with a serious gap over a boring hyper-correct one.*

Formal mathematicians themselves acknowledge difficulties with their proofs. Lawrence Paulson writes:

> *However, existing theorem provers are unsuitable for mathematics. Their formal proofs are unreadable.* [17]

A closer look, however, reveals that informal proofs usually contain a considerable amount of formality, and that current proof assistants are moving towards proof languages and proof presentations that are at least “readable” by human experts. The natural proof assistant $\mathbb{N}$aproche attempts to close the gap between informal and formal mathematics. Some texts which are proof-checked by $\mathbb{N}$aproche come close to ordinary mathematical writing. This is further emphasized by a new LaTeX dialect of the $\mathbb{N}$aproche input language ForTheL which allows immediate mathematical typesetting of input files. The $\mathbb{N}$aproche project aims at providing comfortable editing of natural mathematical texts with integrated automated proof checking.

$\mathbb{N}$aproche is included as a bundled component in the latest edition of the Isabelle prover platform. ForTheL texts in the classic `.ftl` format or the new `.ftl.tex` format can be edited in Isabelle/jEdit and are automatically checked by $\mathbb{N}$aproche. In this paper we present some formalization examples which are included in Isabelle 2021. These short texts demonstrate the potential for writing mathematics in a natural yet completely formal language and to delegate tedious detail to strong automated theorem proving. The examples present proofs that can be considered “beautiful”. Some of them follow proofs in THE BOOK [12].

The examples are contained in the folder `contrib/naproche-*/examples` within the Isabelle 2021 folder. They cover

- Cantor’s diagonal argument (`cantor.ftl.tex`, section 3);
- König’s Theorem from cardinal arithmetic (`koenig.ftl.tex`, section 4);
- the infinitude of primes according to Euclid (`euclid.ftl.tex`, section 5)
- ... and according to Fürstenberg (`fuerstenberg.ftl.tex`, section 6);
- the Knaster–Tarski fixpoint theorem (`tarski.ftl.tex`, 7).

Some of these formalizations go back to example texts that Andrei Paskevich included with his original SAD system ([15] and [16]). The files can be opened in Isabelle, and PDF-versions are provided for immediate reading. Note that we have made a few superficial typographic changes to the examples in this paper to increase legibility. There is always room for further improvements to the typesetting of texts.

In conclusion: we are certain that natural proof assistants will facilitate the eventual acceptance of formal mathematics in the wider mathematical community. Ideally, proofs should be beautiful *and* formally correct.

2 $\mathbb{N}$aproche, ForTheL, and LaTeX

The $\mathbb{N}$aproche proof assistant stems from two long-term efforts aiming towards naturalness: the Evidence Algorithm (EA) / System for Automated Deduction (SAD) projects at the universities of Kiev and Paris [15,16,20,21], and the Naproche project at Bonn [11,1,3,9]. In $\mathbb{N}$aproche, the ForTheL input language of SAD has been extended and embedded in LaTeX, allowing mathematical typesetting; the original proof-checking mechanisms have been made more efficient and varied.

The mathematical controlled language ForTheL has been developed over several decades in the Evidence Algorithm (EA) / System for Automated Deduction (SAD) project. It is carefully designed to approximate the weakly typed natural language of mathematics whilst being efficiently translatable to the language of first-order logic. In ForTheL, standard mathematical types are called *notions*, and these are internally represented as predicates with one distinguished variable, whilst the other variables are considered as parameters ("types as dependent predicates"). Compared to type systems of standard proof assistants, this yields a more flexible dependent type system where number systems can be cumulative ($\mathbb{N} \subseteq \mathbb{Q} \subseteq \mathbb{R}$), and notions can depend on parameters (subsets of $\mathbb{N}$, divisors of n).

First-order languages of notions, constants, relations, and functions can be introduced and extended by *signature* and *definition* commands. The formalization of Euclid's theorem to be discussed later, sets out like:

> **Signature.** A natural number is a small object.
> Let $\ldots m, n \ldots$ denote natural numbers.
> **Signature.** 0 is a natural number.
> ...
> **Signature.** $m + n$ is a natural number.

We have extended ℕaproche to support a `.ftl.tex` format, in addition to the original `.ftl` format. Files in `.ftl.tex` format can be processed by ℕaproche for logical checking and by LaTeX for typesetting.

The LaTeX tokenizer ignores everything except what is inside `forthel` environments of the form

```
\begin{forthel}
    % Insert what you want Naproche to process here
\end{forthel}
```

Inside a `forthel` environment, standard LaTeX syntax can be used for declaring text environments for theorems and definitions.

In ℕaproche, users can define their own operators and phrases through *patterns* of words and symbols. This mechanism has been adapted to allow LaTeX constructs in patterns. In the Euclid example we shall use the pattern `\Set{p}{1}{r}` for the finite set $\{p_1, \ldots, p_r\}$. By also defining `\Set` as a LaTeX macro we can arrange that the ForTheL pattern will be printed out in familiar set notation:

```
    \newcommand{\Set}[3]{\{#1_{#2}, \dots,#1_{#3}\}}
```

There are some primitive concepts in ℕaproche, such as the logical operators $\vee$, $\wedge$, $\exists$ that are directly recognized in the LaTeX source and expanded to corresponding internal tokens.

3 Example: Cantor's Theorem

In this section we prove Cantor's famous theorem, by which the powerset of a set has strictly greater cardinality than the given set. The proof rests on Cantor's beautiful diagonal argument which is also used in THE BOOK [12] to show that the set of real numbers is not countable.

Our formalization is so short, that we can include it in its entirety and use it to remark on further features of ℕaproche and ForTheL. More information can be found in a short tutorial introduction to ℕaproche in the file `TUTORIAL.ftl.tex` in the examples folder.

[synonym subset/-s] [synonym surject/-s]

Let M denote a set. Let f denote a function.

Axiom 1 *M is setsized.*

Axiom 2 *Let x be an element of M. Then x is setsized.*

Let the value of f at x stand for $f(x)$. Let f is defined on M stand for $\mathrm{Dom}(f) = M$. Let the domain of f stand for $\mathrm{Dom}(f)$.

Axiom 3 *The value of f at any element of the domain of f is a set.*

Definition 1 (Subset). *A subset of M is a set N such that every element of N is an element of M.*

Definition 2. *The powerset of M is the class of subsets of M.*

Axiom 4 *The powerset of M is a set.*

Definition 3. *f surjects onto M iff every element of M is equal to the value of f at some element of the domain of f.*

Theorem 1 (Cantor). *No function that is defined on M surjects onto the powerset of M.*

Proof. Proof by contradiction. Assume the contrary. Take a function f that is defined on M and surjects onto the powerset of M. Define

$$N = \{x \in M \mid x \notin f(x)\}.$$

Take an element z of M such that $f(z) = N$. Then

$$z \in N \leftrightarrow z \notin f(z) = N.$$

Contradiction. ■

Remarks:

1. This formalization, like the subsequent examples, is a self-contained natural language representation of a collection of first-order assumptions and consequences. Except for some built-in notions and axioms the whole logical scenario has to be set up explicitly. Future versions of ℕaproche will contain libraries of foundational theories which can be imported into formalizations.

2. The simple grammar of ℕaproche and ForTheL is directed towards the identification of first-order logical content. Writing grammatically correct English is possible (and encouraged) but not enforced by the system. [synonym subset/-s] is a parser command that identifies the token "subsets" with the token "subset". This allows to choose the correct grammatical number in statements. Note that ℕaproche does not have a predefined English vocabulary but works with arbitrary alphabetic tokens.

3. The notions of "set" and "function" are already coded into ℕaproche. Variables like M or f can be pretyped with those notions by, e.g., "Let M denote a set."

4. A rudimentary set- and class-theory is built into ℕaproche. Since classes can only contain "setsized" elements, we stipulate that every set is setsized by the axiom: "M is setsized." Also elements of sets are setsized by Axiom 2.

5. Naturalness requires to have alternative phrases available for the same logical entity, so that one may speak of the "value of f at x" instead of $f(x)$. Such alternatives are introduced by "Let ... stand for ..." commands.

6. Definitions 1 and 2 define new notions dependent on the pre-typed variable M for a set.

7. Axiom 4 is the well-known powerset axiom.

8. The short proof of Cantor's theorem uses the same language as undergraduate texts on basic set theory. A mathematical context is created by "Assume ...", "Take ...", or "Define ..." statements. At proof time ℕaproche checks that all terms and statements are type-correct: the term $f(x)$, e.g., spawns the obvious prover task derived from the assumptions in the definition of $f(x)$; namely that $x \in \mathrm{Dom}(f)$. This task is given to the background ATP eprover which is able to prove it within the local proof context.

9. Abstraction terms $\{\dots\}$ are already built into the syntactic mechanisms of ℕaproche.

10. ℕaproche supports familiar proof methods like proofs by cases, by induction, or, in this case, by contradiction. Internally, these methods influence the construction of proof tasks.

11. Mathematical typesetting is an important ingredient of the "beauty" of mathematical texts. ℕaproche mostly treats LaTeX commands as orthogonal to the logical content of a text and ignores them during parsing. This allows common layout features like prominently displaying the definition of N or the final equivalence by `\[ ... \]` commands.

4 Example: König's Theorem

The next example presents an important set-theoretical result about the arithmetic of cardinals which was proved by Julius König in 1905 [10] The global proof structure is again a Cantorean diagonal argument.

Mathematical notation greatly contributes to the brevity and aesthetics of mathematical texts. The "big operator" notation for multiple sums ($\sum$) or products ($\prod$) with their 2-dimensional arrangement of arguments represents typical mathematical symbolism.

These terms can be typeset by LaTeX macros which by the generous pattern mechanisms of ForTheL simultaneously stand for first-order functions. The sum macro is defined by:

```
\newcommand{\Sum}[2]{\sum_{i \in #2} \val{{#1}_{i}}{}}
```

`\Sum{_}{_}` is simultaneously used as a ForTheL pattern for an internal binary function. The instance `\Sum{\kappa}{D}` of the pattern typesets as $\sum_{i \in D} \kappa_i$.

The LaTeX interpretation of certain ForTheL patterns and the orthogonality of most LaTeX commands to the logical interpretation allow many typographical effects, according to taste and style.

Theorem. *Let κ, λ be sequences of cardinals on D. Assume that for every element i of D $\kappa_i < \lambda_i$. Then*

$$\sum_{i \in D} \kappa_i < \prod_{i \in D} \lambda_i.$$

Proof. Proof by contradiction. Assume the contrary. Then

$$\prod_{i \in D} \lambda_i \leqslant \sum_{i \in D} \kappa_i.$$

Take a function G such that $\dot{\bigcup}_{i \in D} \kappa_i$ is the domain of G and $\times_{i \in D} \lambda_i$ is the image of G. Indeed $\times_{i \in D} \lambda_i$ has an element.

Define

$$\varDelta(i) = \{G((n, i))(i) \mid n \text{ is an element of } \kappa_i\} \text{ for } i \text{ in } D.$$

For every element f of $\times_{i \in D} \lambda_i$ for every element i of D $f(i)$ is an element of λ_i. For every element i of D λ_i is a set. For every element i of D for every element d of $\varDelta(i)$ we have $d \in \lambda_i$. For every element i of D $\varDelta(i)$ is a set.

(1) For every element i of D $|\varDelta(i)| < \lambda_i$.

Proof. Let i be an element of D. Define

$$F(n) = G((n, i))(i) \text{ for } n \text{ in } \kappa_i.$$

Then $F[\kappa(i)] = \varDelta(i)$. qed.

Define

$$f(i) = \text{ choose an element } v \text{ of } \lambda_i \backslash \varDelta(i) \text{ in } v \text{ for } i \text{ in } D.$$

Then f is an element of $\times_{i \in D} \lambda_i$. Take an element j of D and an element m of κ_j such that $G((m, j)) = f$. $G((m, j))(j)$ is an element of $\varDelta(j)$ and $f(j)$ is not an element of $\varDelta(j)$. Contradiction. ■

5 Example: Euclid's Theorem

We formalize the very first proof in THE BOOK [12], Euclid's theorem that there are infinitely many prime numbers. Before the proof the example sets up the axiomatic background: a language and axioms for natural numbers, arithmetic, divisibility and prime numbers, some set theory, and finite sets, sequences and products. Here we only present the concluding proof, juxtaposing the BOOK proof (left) and the $\mathbb{N}$aproche proof (right) in order to demonstrate their similarity:

	Signature. $\mathbb{P}$ *is the class of prime natural numbers.*
	Theorem (Euclid). $\mathbb{P}$ *is infinite.*
Euclid's Proof.	*Proof.*
For any finite set $\{p_1, \ldots, p_r\}$ of primes,	Assume that r is a natural number and p is a sequence of length r and $\{p_1, \ldots, p_r\}$ is a subclass of $\mathbb{P}$.
	(1) p_i is a nonzero natural number for every i such that $1 \leqslant i \leqslant r$.
consider the number $n = p_1 p_2 \cdots p_r + 1$.	Consider $n = p_1 \cdots p_r + 1$.
This n has a prime divisor p.	Take a prime divisor q of n.
But p is not one of the p_i:	Let us show that $q \neq p_i$ for all i such that $1 \leqslant i \leqslant r$.
otherwise	Proof by contradiction. Assume that $q = p_i$ for some natural number i such that $1 \leqslant i \leqslant r$.
p would be a divisor of n and of the product $p_1 p_2 \cdots p_r$,	q is a divisor of n and q is a divisor of $p_1 \cdots p_r$ (by factor property, 1).
and thus also of the difference $n - p_1 p_2 \cdots p_r = 1$,	Thus q divides 1.
which is impossible.	Contradiction. qed.
So a finite set $\{p_1, \ldots, p_r\}$ cannot be the collection of *all* prime numbers. ■	Hence $\{p_1, \ldots, p_r\}$ is not the class of prime natural numbers. ■

6 Example: Fürstenberg's Topological Proof

In 1955 Hillel Fürstenberg published another proof of the infinitude of primes using the language of topology. [4]. Paskevich provided a version of this proof as a ForTheL example in SAD [14] which we translated to ForTheL's LaTeX dialect in the course of the release of Isabelle 2021 [7]. Here is the concluding theorem and proof, taken directly from the example file in Isabelle 2021.

Theorem (Fuerstenberg). *Let* $S = \{r\mathbb{Z} + 0 \mid r \textit{ is a prime}\}$. S *is infinite.*

Proof. Proof by contradiction. S is a family of integer sets.

We have $\overline{\bigcup S} = \{1, -1\}$.
Proof. Let us show that for any integer n n belongs to $\bigcup S$ iff n has a prime divisor. Let n be an integer.

If n has a prime divisor then n belongs to $\bigcup S$.
Proof. Assume n has a prime divisor. Take a prime divisor p of n. $p\mathbb{Z} + 0$ is setsized. $p\mathbb{Z} + 0 \in S$. $n \in p\mathbb{Z} + 0$. Qed.

If n belongs to $\bigcup S$ then n has a prime divisor.
Proof. Assume n belongs to $\bigcup S$. Take a prime r such that $n \in r\mathbb{Z} + 0$. Then r is a prime divisor of n. Qed. End. Qed.

Assume that S is finite. Then $\bigcup S$ is closed and $\overline{\bigcup S}$ is open.

Take p such that $p\mathbb{Z} + 1 \subseteq \overline{\bigcup S}$.

$p\mathbb{Z} + 1$ has an element x such that neither $x = 1$ nor $x = -1$.
Proof. $1 + p$ and $1 - p$ are integers. $1 + p$ and $1 - p$ belong to $p\mathbb{Z} + 1$. Indeed $1 + p = 1 \pmod{p}$ and $1 - p = 1 \pmod{p}$. $1 + p \neq 1 \wedge 1 - p \neq 1$. $1 + p \neq -1 \vee 1 - p \neq -1$. Qed.

We have a contradiction. ■

In 2020 Manuel Eberl published an Isar version of Fürstenberg's proof in the Archive of Formal Proofs [2]. In this section we will discuss the formalization in ForTheL's LaTeX dialect and compare it with Eberl's Isar version.

Let us start with the statement.

Theorem. *There are infinitely many primes.*

Despite its apparent simplicity, it is not as easy as it seems to formalize it. Even the *natural* formal language ForTheL cannot capture it. The problem is the quantification "there are infinitely many". We reformulate the statement in terms of the cardinality of the set of primes as in the Isar formalization:

Theorem. *infinite* $\{p :: \textit{nat. prime } p\}$

Here we have a unary predicate *infinite* with an argument $\{p :: \textit{nat. prime } p\}$. In ForTheL however we cannot pass class terms as parameters to predicates, hence we cannot adopt the Isar statement literally to ForTheL. So what we have to state instead is the following:

Theorem. *Let* $S = \{r \mid r \textit{ is a prime}\}$. S *is infinite.*

In fact Paskevich's formalization of Fürstenberg's proof does not provide a full axiomatization of integers or even a general notion of infinity, and rather proves the infinitude of the set $\{p\mathbb{Z} \mid p \text{ is a prime}\}$. The ForTheL theorem thus reads:

Theorem. *Let* $S = \{r\mathbb{Z} + 0 \mid r \textit{ is a prime}\}$. S *is infinite.*

Note that we cannot write $r\mathbb{Z}$ instead of $r\mathbb{Z}+0$. For our formalization introduces the pattern $q\mathbb{Z}+a$ for arbitrary integers a, q (where q is supposed to be non-zero). If we would additionally define the pattern $q\mathbb{Z}$ as $q\mathbb{Z}+0$ then $\mathbb{N}$aproche could not figure out the meaning of $q\mathbb{Z}+0$. It could either refer to the pattern $(x\mathbb{Z}+y)[q/x, 0/y]$ or to the pattern $(x+y)[q\mathbb{Z}/x, 0/y]$, where $[t/x, t'/y]$ denotes substitution of x by t and of y by t'. The further development of $\mathbb{N}$aproche will have mechanisms to disambiguate such overloadings.

Let us continue our comparison of the Isar version of Fürstenberg's proof with the ForTheL version. The Isar proof begins with the following statements:

assume *fin*: *finite* $\{p :: \mathit{nat.\ prime}\ p\}$
define A **where** $A = (\bigcup p \in \{p :: \mathit{nat.\ prime}\ p\}.\mathit{arith\text{-}prog\text{-}fb}\ 0\ p)$
have *closed* A
...
hence *open* $(-A)$

Here *arith-prof-fb* $0\ p$ denotes the set $p\mathbb{Z}+0$ and $-A$ denotes the complement of A in $\mathbb{Z}$. In ForTheL we can directly write $p\mathbb{Z}+0$ which allows for a better intuitive understanding of the proof text:

Let $S = \{r\mathbb{Z}+0 \mid r \text{ is a prime}\}$.
...
Assume that S is finite. Then $\bigcup S$ is closed and $\overline{\bigcup S}$ is open.

Up to now both proof texts are quite similar (if we “‘identify” a prime p with the set $p\mathbb{Z}+0$), except that ForTheL uses natural language constructs like subject-predicate-object sentences. The central part of Fürstenberg's proof is to show that

$$\bigcup\{p\mathbb{Z}+0 \mid p \text{ is prime}\} = \mathbb{Z}\backslash\{1,-1\}$$

(as in the Isar version) or, equivalently, that

$$\bigcup\{r\mathbb{Z}+0 \mid r \text{ is a prime}\} = \{n \in \mathbb{Z} \mid n \text{ has a prime divisor}\}$$

(as in the ForTheL version). Let us first have a look at how the statement $\mathbb{Z}\backslash\{1,-1\} \subseteq \bigcup\{p\mathbb{Z}+0 \mid p \text{ is prime}\}$ is proven in the Isar text.

fix $x :: \mathit{int}$ **assume** $x : x \in -\{1,-1\}$
...
show $x \in (\bigcup p \in \{p :: \mathit{nat.\ prime}\ p\}.\ \mathit{arith\text{-}prog\ 0\ p})$
...
obtain p **where** p : *prime p p dvd x*
using *prime-divisor-exists*[*of x*] **and** $\langle |x| \neq 1\rangle$ **by** *auto*
hence $x \in$ *arith-prog 0* (*nat p*) **using** *prime-gt-0-int*[*of p*]
by *(auto simp: arith-prog-def cong-0-iff)*
thus *?thesis* **using** *p*
by *(auto simp: A-def intro!: exI* [*of - nat p*]*)*

On the other hand the proof of the statement $\{n \in \mathbb{Z} \mid n \text{ has a prime divisor}\} \subseteq \bigcup\{p\mathbb{Z} + 0 \mid p \text{ is prime}\}$ looks like the following in ForTheL.

> Let n be an integer.
> ...
> If n has a prime divisor then n belongs to $\bigcup S$.
> Proof. Assume n has a prime divisor. Take a prime divisor p of n. $n \in p\mathbb{Z} + 0$. Qed.

Note that in both versions we silently assumed that $x \neq 0$ and $n \neq 0$, respectively.

In principle, both proofs are similar. But whereas Isabelle uses proof tactics to search for proofs, $\mathbb{N}$aproche relies on an external ATP. Users of Isabelle can steer proof search efficiently by commands like

```
by (auto simp: A-def intro!: exI [of - nat p])
```

On the other hand one would not want to see such technicalities in a natural or even "beautiful" proof à la $\mathbb{N}$aproche. As a future project we shall investigate whether $\mathbb{N}$aproche can reach a similar prover efficiency by using sledgehammer methods to steer external ATPs.

Finally, let us compare the statement discussed above to its original formulation in THE BOOK [12]:

> Since any number $n \neq 1, -1$ has a prime divisor p, and hence is contained in $N_{0,p}$, we conclude
> $$\mathbb{Z} \backslash \{1, -1\} = \bigcup_{p \in \mathbb{P}} N_{0,p}.$$

Here $\mathbb{P}$ denote the set of prime numbers and $N_{0,p}$ the set $p\mathbb{Z} + 0$. Obviously this is a very elegant formulation compared to Isar and ForTheL, paying the price of a quite complicated sentence structure as a combination of three statements with internal dependencies:

> Since φ, and hence ψ, we conclude χ.

Moreover, there are hidden variables, e.g. n occurs in ψ without being explicitly mentioned, and implicit variable bindings, e.g. p is not free in ψ as it might seem if we consider ψ being independent from φ. Parsing such sentences is beyond the possibilities of the current $\mathbb{N}$aproche, and it will have to be discussed if one would even want this level of grammatical complication in an efficient controlled natural language for mathematics.

7 Example: Knaster–Tarski

The Knaster–Tarski theorem is a result from lattice theory about fixed points of monotone functions. Bronisław Knaster and Alfred Tarski established it in 1928 for the special case of power set lattices [8]. This more general result was stated by Tarski in 1955 [19].

The theorem states that the set of fixpoints of a monotone function on a complete lattice is also a complete lattice. In particular, we can take the supremum or the infimum of the empty set in order to get the biggest or the smallest fixpoint. We see that the Knaster–Tarski theorem is considerably stronger than a mere statement about the existence of fixpoints.

The full formalization starts by defining a complete lattice, a monotone function and a fixpoint. The formalized proof relies on automation to achieve a natural brevity.

Theorem (Knaster–Tarski). *Let U be a complete lattice and f be a monotone function on U. Let S be the class of fixed points of f. Then S is a complete lattice.*

Proof. Let T be a subset of S.
Let us show that T has a supremum in S.
Define

$$P = \{x \in U \mid f(x) \leqslant x \text{ and } x \text{ is an upper bound of } T \text{ in } U\}.$$

Take an infimum p of P in U. $f(p)$ is a lower bound of P in U and an upper bound of T in U. Hence p is a fixed point of f and a supremum of T in S.
End.
Let us show that T has an infimum in S.
Define

$$Q = \{x \in U \mid f(x) \leqslant x \text{ and } x \text{ is an lower bound of } T \text{ in } U\}.$$

Take a supremum q of Q in U. $f(q)$ is an upper bound of Q in U and a lower bound of T in U. Hence q is a fixed point of f and an infimum of T in S.
End. ■

8 Outlook

The ℕaproche project will continue to expand the methods presented in this paper. We shall enlarge our grammar to capture more natural language phrases. Recurrent notions and notations will be predefined in library files. Tuning the background ATP for the demands of ℕaproche checking will allow to make further proof steps implicit and make it easier to follow existing natural texts.

This approach will have to prove its value by further, more comprehensive formalizations and by interlinked libraries of natural formalizations.

References

1. Marcos Cramer. *Proof-checking mathematical texts in controlled natural language.* PhD thesis, University of Bonn, 2013.

2. Manuel Eberl. Furstenberg's topology and his proof of the infinitude of primes. *Archive of Formal Proofs*, March 2020. https://isa-afp.org/entries/Furstenberg_Topology.html, Formal proof development.
3. Steffen Frerix and Peter Koepke. Automatic proof-checking of ordinary mathematical texts. *Proceedings of the Workshop Formal Mathematics for Mathematicians*, 2018.
4. Hillel Fürstenberg. On the infinitude of primes. *American Mathematical Monthly*, 62(5):353, 1955.
5. R. Hersh. *What is Mathematics, Really?* Oxford University Press, 1997.
6. Matthew Inglis and Andrew Aberdein. Beauty is not simplicity: An analysis of mathematicians' proof appraisals. *Philosophia Mathematica*, 23:87–109, 02 2014.
7. Isabelle contributors. The Isabelle2021 release, February 2021.
8. Bronisław Knaster and Alfred Tarski. Un théorème sur les fonctions d'ensembles. *Annales de la Société Polonaise de Mathématique*, 6:133–134, 1928.
9. Peter Koepke. Textbook mathematics in the Naproche-SAD system. *Joint Proceedings of the FMM and LML Workshops*, 2019.
10. Julius König. Zum Kontinuumsproblem. *Mathematische Annalen*, 60:177–180, 1905.
11. Daniel Kühlwein, Marcos Cramer, Peter Koepke, and Bernhard Schröder. The Naproche system, 2009.
12. Günter M. Ziegler Martin Aigner. *Proofs from THE BOOK*. Springer-Verlag, 4th edition, 2009.
13. James W McAllister. Mathematical beauty and the evolution of the standards of mathematical proof. In Micheleed Emmer, editor, *The Visual Mind II*, pages 15–34. Cambridge, Mass.: MIT Press, 2005.
14. Andrei Paskevich. Fürstenbeg's proof in SAD.
15. Andrei Paskevich. *Méthodes de formalisation des connaissances et des raisonnements mathématiques: aspects appliqués et théoriques*. PhD thesis, Université Paris 12, 2007.
16. Andrei Paskevich. The syntax and semantics of the ForTheL language, 2007.
17. L. C. Paulson. Alexandria: Large-scale formal proof for the working mathematician, 2018.
18. Gian-Carlo Rota. The phenomenology of mathematical beauty. *Synthese*, 111(2):171–182, 1997.
19. Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955.
20. Konstantin Verchinine, Alexander Lyaletski, and Andrei Paskevich. System for automated deduction (SAD): a tool for proof verification. *Automated Deduction–CADE-21*, pages 398–403, 2007.
21. Konstantin Verchinine, Alexander Lyaletski, Andrei Paskevich, and Anatoly Anisimov. On correctness of mathematical texts from a logical and practical point of view. In *International Conference on Intelligent Computer Mathematics*, pages 583–598. Springer, 2008.
22. David Wells. Are these the most beautiful? *The Mathematical Intelligencer*, 12:37–41, 09 1990.