Managing $\mathbb{N}$aproche Formalizations with sTEX and FLAMS

Marcel Schütz

FAU Erlangen-Nürnberg

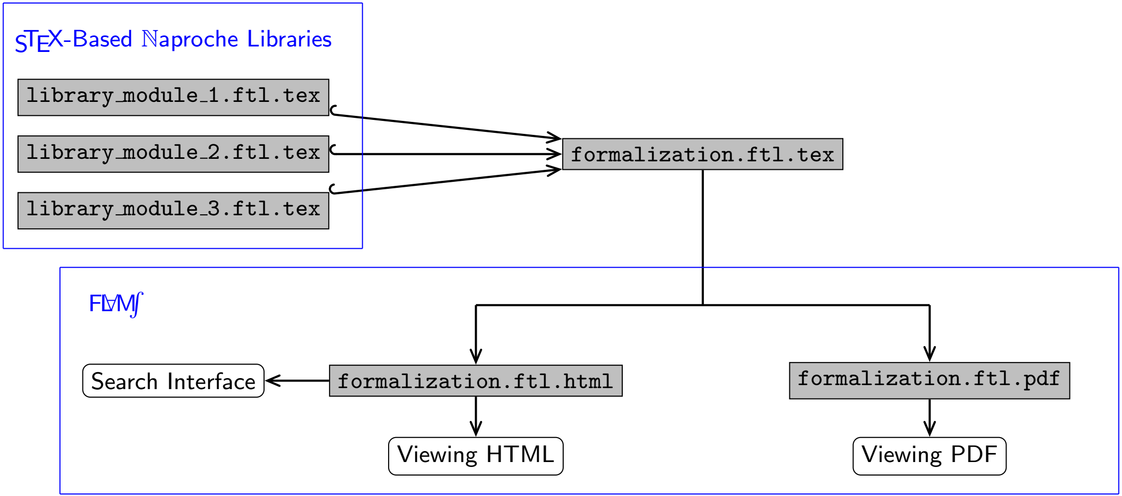
06 October 2025

NatFoM 2025, 6 October 2025
at CICM 2025, Brasilia, Brazil, 6–10 October 2025

Naproche

Overview

Overview


sTeX-Based Naproche Libraries
library_module_1.ftl.tex
library_module_2.ftl.tex
library_module_3.ftl.tex
formalization.ftl.tex
FLAMS
Search Interface
formalization.ftl.html
formalization.ftl.pdf
Viewing HTML
Viewing PDF

Motivation

Motivation

NatFoM 2023:

Introduced libraries of $\mathbb{N}$aproche formalizations

$\Rightarrow$ Focussed on the "logical level"

Remaining Challenges on the "Presentational Level":

For Formalization Authors:

- Importing TEX macro definitions from a library
- Referencing assertions stated in a library
- Searching symbols/notions/assertions across libraries

For Formalization Readers:

- Retrieving definitions/statements of symbols/notions/assertions imported from a library

Solution Approach: Integrating sTEX and FLAMS into the $\mathbb{N}$aproche ecosystem

sTeX

sTeX: A Module System for LaTeX Documents

https://github.com/slatex/stex

- sTeX provides a module system for LaTeX documents

  ⇒ Structure $\mathbb{N}$aproche libraries as sTeX modules

- TeX macros defined sTeX modules can be reused in other LaTeX documents

  ⇒ Define $\mathbb{N}$aproche symbols as sTeX macros

- Assertions defined in sTeX modules can be referenced in other LaTeX documents

  ⇒ Provide $\mathbb{N}$aproche assertions as components of sTeX modules

sTeX: ℕaproche Libraries as sTeX Modules

```
\begin{smodule}{congruency.ftl}

  \importmodule[libraries/arithmetics]{modulo.ftl}

  \symdef{Cong}[args=3]{#1 \equiv #2 (\text{mod} #3)}

  \symdecl*{congruency of sum}

  \begin{definition}[forthel, for=Cong]
    Let $n, m, k$ be natural numbers such that
    $k \NotEq \Zero$.
    $\Cong{n}{m}{k}$ iff $n \Mod k \Eq m \Mod k$.
  \end{definition}

  \begin{lemma}[forthel, name=congruency of sum]
    Let $n, k$ be natural numbers such that
    $k \NotEq \Zero$.
    Then $\Cong{n \Plus k}{n}{k}$.
  \end{lemma}

\end{smodule}
```

Definition. Let n, k be natural numbers such that $k \neq 0$. $n \equiv m \pmod{k}$ iff $n \bmod k = m \bmod k$.

Lemma. Let n, m, k be natural numbers such that $k \neq 0$. Then $n + k \equiv n \pmod{k}$.

sTeX: Reusing sTeX Macros & Referencing Assertions

Reusing `\Cong` **& Referencing** `congruency of sum`**:**

```
\usemodule[libraries/arithmetics]{congruency.ftl}

...
Then $\Cong{\One \Plus p}{\One}{p}$ (by \sn{congruency of sum}).
```

$\Rightarrow$ Interactive HTML Rendering:

Then $1 + p \equiv 1 \bmod p)$ (by congruency of sum).

Definition Let n, m, k be natural numbers such that $k \neq 0$. $n \equiv m \bmod k)$ iff $n \bmod k = m \bmod k$.

Then $1 + p \equiv 1 \bmod p)$ (by congruency of sum).

Assertion Let n, k be natural numbers such that $k \neq 0$. Then $n + k \equiv n \bmod k)$.

FL∀M∫

FL∀M∫: The Flexiformal Annotation Management System

https://github.com/FlexiFormal/FLAMS

FL∀M∫ provides a webinterface to ...

- ... render LaTeX documents as PDF and interactive HTML.

  ⇒ Render ℕaproche formalizations in FL∀M∫

- ... view PDF and interactive HTML documents.

  ⇒ View ℕaproche formalizations in FL∀M∫

- ... search sTeX symbols.

  ⇒ Search ℕaproche symbols/notions/assertions in FL∀M∫

FTML: Rendering Formalizations

- All Archives ⓘ
  - FTML ⓘ
  - documentation ⓘ
  - libraries ⓘ
    - arithmetics ⓘ
    - everyday-ontology ⓘ
    - foundations ⓘ
    - lists ⓘ
      - concatenation.ftl.en.tex ⓘ
      - length-of-concatenation.ftl.en.tex ⓘ
      - length-of-reversion.ftl.en.tex ⓘ
      - length.ftl.en.tex ⓘ
      - lists.ftl.en.tex ⓘ
      - reversion.ftl.en.tex ⓘ
    - meta ⓘ
    - set-theory ⓘ
  - articles ⓘ
  - meta-inf ⓘ

53.3%; ca. 20s remaining

Running (2)

- ▸ **[libraries/lists]concatenation.ftl.en.tex (2/5) pdflatex**
- ▸ **[libraries/lists]length.ftl.en.tex (1/5) pdflatex_first**

Queued (1)

- ▸ **[libraries/lists]lists.ftl.en.tex (1/5) pdflatex_first**

Blocked (0)

Failed (0)

Finished (3)

- ▸ **[libraries/lists]length-of-reversion.ftl.en.tex (5/5) Done**
- ▸ **[libraries/lists]reversion.ftl.en.tex (5/5) Done**
- ▸ **[libraries/lists]length-of-concatenation.ftl.en.tex (5/5) Done**

FL∀M∫: Viewing Formalizations

Theorem 18 (Furstenberg). $\mathbb{P}$ is infinite.

Proof. $\bigcup P$ is a subset of $\mathbb{N}$. Let us show that for any $n \in \mathbb{N}$ we have $n \in \bigcup P$ iff n has a prime divisor. Let $n \in \mathbb{N}$. If n has a prime divisor then n belongs to $\bigcup P$.

Proof. Assume n has a p

have $N_{0,p} \in P$. He

If n belongs to $\bigcup P$

Definition 2. $\mathbb{P}$ is the class of all prime numbers.

Definition 15. $P = \{N_{0,p} \mid p \in \mathbb{P}\}$.

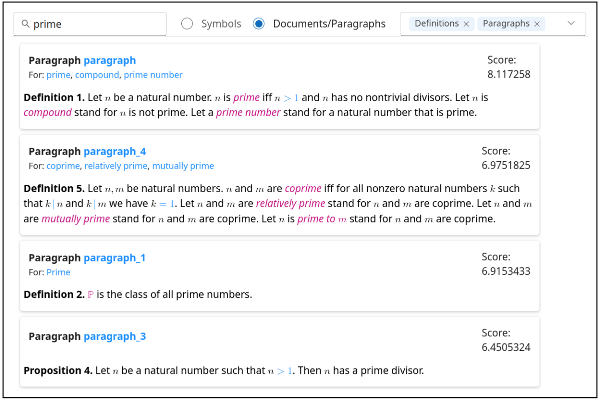
Proof. Assume that n belongs to $\bigcup P$. Take a prime number r such that $n \in N_{0,r}$. Hence $n \equiv 0 \pmod{r}$. Thus $n \bmod r = 0 \bmod r = 0$. Therefore r is a prime divisor of n. □

End. Hence For all $n \in \mathbb{N}$ we have $n \in \bigcup P^{\complement}$ iff n has no prime divisor. 1 has no prime divisor and any natural number having no prime divisor is equal to 1. Therefore $\bigcup P^{\complement} = \{1\}$. Indeed $\bigcup P^{\complement} \subset \{1\}$ and $\{1\} \subset \bigcup P^{\complement}$. P is infinite.

Proof by contradiction. Assume that P is finite. Then $\bigcup P$ is closed and $\bigcup P^{\complement}$ is open. Take a p such that $N_{1,p} \subset \bigcup P^{\complement}$. $1 + p$ is an element of $N_{1,p}$. Indeed $1 + p \equiv 1 \pmod{p}$ (by congruency of sum). $1 + p$ is not equal to 1. Hence $1 + p \notin \bigcup P^{\complement}$. Contradiction. □

■

FL∀M∫: Searching Symbols/Notions/Assertions



Conclusion

Conclusion: Managing ℕaproche Formalizations with sTeX and FLAMS

Challenges wrt. Naproche Libraries:

- Importing TeX macros and referencing assertions from a ℕaproche library

  ⇒ Structure ℕaproche libraries as sTeX modules

- Searching symbols/notions/assertions across libraries

  ⇒ Use the search interface of FLAMS

- Retrieving definitions/statements of symbols/statements

  ⇒ Use FLAMS to convert formalizations to HTML that shows definitions/statements on mouse hover

Links:

ℕaproche:	https://github.com/naproche/naproche/
sTeX:	https://github.com/slatex/stex
FLAMS:	https://github.com/FlexiFormal/FLAMS