Scaling a natural proof assistant

Adrian De Lon (University of Bonn)

2023-09-06

Size of Naproche/SAD formalizations over time (approximate)

2007 (Paskevich): Ramsey’s theorem (540 SLOC), smaller examples

2020 (Sturzenhecker): representation theory (4200 SLOC)

2020 (Penquitt): set theory up to Silver’s theorem (13800 deduped SLOC)

2019–2022: linear algebra, geometry, additional examples (<2000 SLOC each)

2022: first 5 chapters of Baby Rudin (>5400 SLOC)

2023: selected theorems from Wiedijk’s 100 (2460 SLOC)

```
## The last Lemma before Silver's Theorem
Lemma Silver2. Let kappa, x, kap, F be objects. Let (kappa,x,kap) be
a coftriple. Let Silver below kappa. Let A be a zffunction. Let A be
silvercompatible with kap relative to kappa and x. Let F be almost
disjoint relative to A. Then Card(F) /subset Plus[kappa].
Proof. # Getting Started
    (kappa,x) is a cofpair.
    x /subset kappa /cap /Card.
```

——— *900 lines later...* ———

```
        kappa /in /Card.
        Then /NN /subset kappa.
        kappa /in Plus[kappa].
        kappa /subset Plus[kappa].
        Then /NN /subset Plus[kappa].
    QED.
    Then Plus[kappa] * Plus[kappa] = Plus[kappa].
    Then Card(Plus[kappa] /times Plus[kappa]) = Plus[kappa].
    Then Card(G) /subset Plus[kappa].
    Then Card(F) /subset Plus[kappa].
QED.
```

Issue: check times

No concurrency, somewhat fragile caching

→ *Thread pool for ATP, individual caching*

Basic import mechanisms (similar to an include directive) can lead to checking theorems multiple times

→ *Replaced by module graph*

Naproche sometimes hopelessly tries to prove things again with unfolded definitions (also leads to silent performance degradation)

→ *Don't do it* ™

No premise selection

→ *Simple MePo-like filter, initial stages of GNN-based premise selection*

Some overhead from class/set/atom distinction (Kelley–Morse + atoms)

→ *Use (higher-order) ZFC instead*

Parser is exponential in some areas

→ *Use CFG + Earley parser (worst-case is cubic time)*

Issue: surface area for user errors

Naproche formalizations often use "low-level" first-order formalism: users freely extend the signature and add axioms.

Some types of functions require axiomatization.

The grammar is too permissive/ambiguous in parts.

Example of a proof in the new version

Theorem (Burali-Forti antimony) There exists no set Ω such that for all α we have $\alpha \in \Omega$ iff α is an ordinal.
Proof. Suppose not. Consider Ω such that for all α we have $\alpha \in \Omega$ iff α is an ordinal. For all x, y such that $x \in y \in \Omega$ we have $x \in \Omega$. So Ω is $\in$-transitive. Thus Ω is an ordinal. Hence $\Omega \in \Omega$. Contradiction. □

```
\begin{theorem}[Burali-Forti antimony]\label{burali_forti}
    There exists no set $\Omega$ such that for all $\alpha$
    we have $\alpha\in \Omega$ iff $\alpha$ is an ordinal.
\end{theorem}
\begin{proof}
    Suppose not.
    Consider $\Omega$ such that for all $\alpha$ we have
        $\alpha\in \Omega$ iff $\alpha$ is an ordinal.
    For all $x, y$ such that $x\in y\in\Omega$ we have $x\in\Omega$.
    So $\Omega$ is \in-transitive. Thus $\Omega$ is an ordinal.
    Hence $\Omega\in\Omega$.
    Contradiction.
\end{proof}
```

Towards higher-order set theory

Natural language mathematics regularly uses higher-order constructs: set comprehensions, induction principles, adverbs used like “locally P”

Naproche has some higher order constructs, but they are special cases and always eliminated.

Induction principle is “magic” and limited to a single symbol.

Idea: full higher-order syntax, “locally first-order” proof automation by default, but also integrate higher-order ATPs (like Zipperposition and Lash).

Premise selection

Different intended use compared to Sledgehammer (always running implicitly, less interactive, no alternative automation tools).

Even just one “bad” irrelevant hypothesis can drastically increase checking times.

You can manually selected premises (reusing standard LaTeX referencing mechanisms) when proofs are too slow.

Manually selected premises are extracted as training data.

I’m reusing Miroslav Olšák’s GNN-based premise selection which worked well for Mizar.

Still needs much more data and experiments, but first impressions were promising.

Included standard library

Currently over 4400 SLOC, covering basics of sets, relations, functions, orders, topology, ordinals, partitions, filters, etc.

Some material from undergraduate textbooks, some ported over from Naproche, Isabelle/ZF, Lean, etc.

Takes about 20 s to check on a reasonably modern laptop.

262 steps with explicitly specified premises (out of over 1000 ATP steps).

Ongoing work

Scaling up premise selection

Experimenting with using higher-order ATPs in the included library

More formalizations, especially set theory (textbook length formalizations)

Thank you!