Naproche-ZF: Lessons learned from implementing a new natural-language-oriented theorem prover

Adrian De Lon

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

Naproche-ZF[1] is a new experimental natural-language-oriented theorem prover based on set theory that uses automated theorem provers to achieve a high degree of proof automation. As discussed in the initial project presentation at AITP 2022 [3], the system's main goal is to scale "natural" theorem proving to larger libraries compared to the existing ℕaproche [4, 7] theorem prover. Indeed, first experiments with Naproche-ZF show that the system can be over ten times faster than ℕaproche when checking formalizations of similar length.

We will present an overview of the new system and share our experiences with first-order and higher-order proof automation using automated theorem provers, slowly growing a natural language standard library of formalized mathematics, and lessons learned from a first student project in formalizing point-set topology.

Natural language processing. The formalization language of Naproche-ZF is a *controlled natural language* (CNL), a formally specified sublanguage of natural mathematical English embedded into LaTeX, which makes writing mathematical formulae and structuring documents straightforward. Only the contents of certain *formal LaTeX-environments* such as `definition`, `theorem`, and `proof` are checked; everything else is informal commentary.

The CNL is specified in the form of a context-free grammar within an embedded domain-specific language provided by the Earley [6] Haskell library. Naproche-ZF uses a scanner written with applicative regular expressions [2] to extract vocabulary information from formalizations and smart paradigms [5] so that users usually do not have manually specify grammatical variations of definienda.

Logical processing and semantics. After parsing, the surface CNL is translated to formulae of material unsorted set theory in higher-order logic with Henkin semantics [1]. Naproche-ZF is developed alongside a standard library in higher-order Tarski–Grothendieck set theory, which is an extension of the de facto foundation of modern mathematics, Zermelo–Fraenkel set theory with the axiom of choice. There is potential for interoperation with other systems based on set theory, such as Mizar, Megalodon, or Isabelle/ZF.

Every claim in a formalization is translated to a problem (which can include all preceding theorems and definitions, as well as all local assumptions and definitions as hypotheses) and exported to external automated theorem provers such as E [9] or Vampire [8]. In its current state this proof automation is locally first-order; it uses automatic elimination of certain higher-order constructs such as set comprehensions to translate proof obligations to TPTP problems in first-order form.

Growing the system. This summer semester we are undertaking a first student project in Naproche-ZF by formalizing point-set topology. This project will naturally lead to bug fixes, grammar refinements, and new features. We will report on the results of the formalization project and on initial impressions from users. We are also porting over some existing formalizations from ℕaproche, which will enable us to give a fairer comparison of the two systems.

[1] Available at https://adelon.net/naproche-zf

References

[1] Brown, C.E., Kaliszyk, C., Pąk, K.: Higher-order Tarski Grothendieck as a foundation for formal proof. In: Harrison, J., O'Leary, J., Tolmach, A. (eds.) ITP 2019 (2019)

[2] Cheplyaka, R.: regex-applicative (2020), https://hackage.haskell.org/package/regex-applicative

[3] De Lon, A., Koepke, P.: Scaling Naproche. In: 7th Conference on Artificial Intelligence and Theorem Proving (AITP 2022) (2022)

[4] De Lon, A., Koepke, P., Lorenzen, A., Marti, A., Schütz, M., Wenzel, M.: The Isabelle/Naproche natural language proof assistant. In: Platzer, A., Sutcliffe, G. (eds.) Automated Deduction – CADE 28. pp. 614–624. Springer International Publishing, Cham (2021), https://doi.org/10.1007/978-3-030-79876-5_36

[5] Détrez, G., Ranta, A.: Smart paradigms and the predictability and complexity of inflectional morphology. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 645–653 (2012)

[6] Fredriksson, O.: Earley (2019), https://hackage.haskell.org/package/Earley

[7] Naproche contributors: Naproche, https://naproche.github.io/

[8] Reger, G., Suda, M., Voronkov, A., Kovács, L., Bhayat, A., Gleiss, B., Hajdu, M., Hozzova, P., Rath, J., Kotelnikov, E., Rawson, M., Riener, M., Robillard, S., Schoisswohl, J.: Vampire 4.7-SMT system description. In: SMT-COMP 2022 (2022), https://smt-comp.github.io/2022/system-descriptions/Vampire.pdf

[9] Schulz, S., Cruanes, S., Vukmirović, P.: Faster, higher, stronger: E 2.3. In: Fontaine, P. (ed.) Proc. of the 27th CADE, Natal, Brasil. pp. 495–507. No. 11716 in LNAI, Springer (2019)