Making Set Theory Great Again: The Naproche-SAD Project

by Steffen Frerix and Peter Koepke

University of Bonn
December 4, 2018

John Harrison, at AITP 2018, gave a programmatic talk "Let's make set theory great again!" in which he proposes to take standard set theory (and classical first-order logic) as a basis for automatic theorem proving, enriched conservatively and along current mathematical practise by a soft type system à la Freek Wiedijk and by syntactic sugar. This should lead to a standard hierarchy of number systems, a principled treatment of undefinedness, and to a "closer correspondence with informal texts".

SAD. A small system which embraces the Harrison approach is the *System for Automated Deduction* (SAD) by Andrei Paskevich et.al. SAD combines natural language input with first-order proof checking. Mathematical texts are expressed in the controlled mathematical language ForTheL, and checked for logical correctness by a "reasoner" together with a standard automated theorem prover like eprover.

Naproche-SAD. In 2017 we began our work with the SAD system, based on experiences with our earlier Naproche system. We have made the code more efficient and added set theoretical mechanisms. At AITP 2018, we reported on our progress. Meanwhile we are able to work with chapter-sized texts at the level of first-year undergraduate mathematics. We are working on a LaTeXinterface, and, together with Makarius Wenzel, on a jedit-PIDE for Naproche-SAD similar to Isabelle-jEdit.

Naturally Enriched First-Order Logic. ForTheL signature commands and definitions like

Signature. A *real number* is a notion. Let x, y, z denote real numbers.

Signature. $\mathbb{R}$ is the set of real numbers.

Definition 1. x *is* positive *iff* $x > 0$.

Signature. An *integer* is a real number. Let a, b denote integers. Let m, n denote positive integers.

serve to set up convenient first-order and set-theoretical environments. ForTheL allows many natural language constructs of ordinary mathematics. Proof methods like case splits, contradiction and induction are supported by automatically generating and checking their implicit proof obligations.

Soft Typing and Undefinedness. SAD allows soft dependent types via notions and adjectives, as in the above example. As in Wiedijk's article types are internally translated into obvious predicates. Type checking is turned into an *ontological check* at "runtime" during proving to ensure that all presuppositions are fulfilled. Usually these first-order obligations are considerably simpler than the main proof task. This approach also encompasses a correct treatment of undefinedness: the ontological check of the notorious fraction $\frac{1}{x}$ and hence the checking of the entire text fails if the system cannot prove $x \neq 0$ in the proof context of the term.

Set Theory. The well-known difficulties of set theory in automatic theorem proving stem from its vast infinite axiom system and the deep iterations of the $\in$-relation in set-theoretically defined notions like numbers. It is essential to keep the proof search away from arbitrary axioms and expansions of notions. In Naproche-SAD, instances of the problematic infinite axiom schemes have to be explicitly invoked, e.g., by the use of abstraction terms or function definitions. The reasoner of Naproche-SAD has a restrained strategy for definition expansions which benefits set theoretic definitions.

Hierarchical Number Systems can easily be employed in ForTheL as indicated in the signature example.

Example from a formalization of the first chapters of Walter Rudin's analysis textbook. One could obtain even stronger natural language resemblances by adding more argumentative phrases to ForTheL.

Theorem 2. (Naproche-SAD) *If* $x \in \mathbb{R}$ *and* $y \in \mathbb{R}$ *and* $x > 0$ *then there is a positive integer* n *such that*

$$n \cdot x > y.$$

Proof. Define $A = \{n \cdot x |\ n \text{ is a positive integer}\}$. Assume the contrary. Then y is an upper bound of A. Take a least upper bound α of A. $\alpha - x < \alpha$ and $\alpha - x$ is not an upper bound of A. Take an element z of A such that not $z \leq \alpha - x$. Take a positive integer m such that $z = m \cdot x$. Then $\alpha - x < m \cdot x$ (by 15b).

$$\alpha = (\alpha - x) + x < (m \cdot x) + x = (m+1) \cdot x.$$

$(m+1) \cdot x$ is an element of A. Contradiction. Indeed α is an upper bound of A. □

Theorem 3. (Rudin's original text) *(a) If* $x \in \mathbb{R}$, $y \in \mathbb{R}$, *and* $x > 0$, *then there is a positive integer* n *such that*

$$n\,x > y.$$

Proof. Let A be the set of all $n\ x$, where n runs through the positive integers. If (a) were false, then y would be an upper bound of A. But then A has a *least* upper bound in $\mathbb{R}$. Put $\alpha = \sup A$. Since $x > 0$, $\alpha - x < \alpha$, and $\alpha - x$ is not an upper bound of A. Hence $\alpha - x < m\,x$ for some positive integer m. But then $\alpha < (m+1)\,x \in A$, which is impossible, since α is an upper bound of A. □

Kelley Morse Class Theory. If one allows *class* quantifiers in the defining properties φ of abstraction terms $\{x|\ \varphi\}$ one is working in Kelley Morse class theory (KM) which is somewhat stronger than Zermelo-Fraenkel set theory. In KM, sets are those classes which are elements of some class. The abstraction term mechanism of Naproche-SAD corresponds to Kelley Morse terms. This has motivated our current formalization of the Appendix of in which the theory KM was introduced. Working with the Appendix has shown the necessity of splitting larger texts into chapters and using ideas of small theories and theory morphisms to control ontological manueuvers like turning the formation of Kuratowski ordered pairs into a basic function.