▸ TIN ADLEŠIĆ, VEDRAN ČAČIĆ, AND MARKO DOKO, *CoqNFU: formalizing New Foundations (with urelements) in Coq.*
Faculty of Teacher Education, University of Zagreb.
E-mail: tin.adlesic@ufzg.hr.
Department of Mathematics, Faculty of Science, University of Zagreb.
E-mail: veky@math.hr.
Heriot-Watt University, Edinburgh, UK.
E-mail: m.doko@hw.ac.uk.

Today, the uncontested winner of the search for the axiomatization of set theory is Zermelo–Fraenkel set theory (ZF), commonly augmented with the axiom of choice. As the years have gone by, ZF has established itself not only as the champion of axiomatic set theory, but also the "default" foundation for the entirety of mathematics. Alternative approaches to set theory, such as Quine's New Foundations (NF), have been left by the wayside.

However, the modern advances in computer-aided theorem proving have shaken up the way people look at the foundations of mathematics. Type theory arose as the main theory on top of which assisted theorem provers should be built. Today set theory (i.e., ZF) is used by people doing mathematics on paper (a.k.a. "mathematicians"), while type theory is used by those doing mathematics on computers (a.k.a. "computer scientists"). What is the reason for this split? The answer might be that the convoluted ways one needs to argue if something is a set in ZF make it difficult to offload such reasoning to a computer, while checking if something is well-typed is a perfect task for computers.

Can we bridge this divide? Perhaps by "redeeming" set theory in the eyes of computers? If we are willing to consider alternative approaches to axiomatization, the future looks very promising. To establish whether a class is a set, the theory of New Foundations (with urelements), NF(U), requires only a simple decidable syntactic check on the formula used for set comprehension! Such syntactic checks, while tedious and unintuitive for humans, are perfect for computers.

In this presentation, we will take a look at the ongoing effort of formalizing NFU in the Coq theorem prover, as a proof of concept of the feasibility of building computer-verified proofs based on set-theoretic principles.