Journal of Automated Reasoning (2019) 63:319–321
https://doi.org/10.1007/s10817-018-09500-0

Selected Extended Papers of ITP 2017

Preface

Mauricio Ayala-Rincón[1] · **César Muñoz**[2]

Received: 9 November 2018 / Accepted: 26 November 2018 / Published online: 4 December 2018

This special issue contains extended versions of selected contributions of the eighth *International Conference on Interactive Theorem Proving* (ITP 2017). The conference was held in Brasília, Brazil, on September 2017 and its proceedings appeared as volume 10499 of the series Lecture Notes in Computer Science (LNCS), published by Springer Nature. The ITP conference series is concerned with all topics related to interactive theorem proving, ranging from theoretical foundations to implementation aspects and applications in program verification, security, and formalization of mathematics. The papers were carefully reviewed by specialists, including members of the ITP 2017 PCs and additional experts. The reviewers guaranteed both significant additional contributions with respect to the LNCS proceedings and assured the quality standards of the Journal of Automated Reasoning, for which the guest editors are grateful. In the following, a short introduction is given to each of the nine papers included in this volume.

In *A Formalization of Convex Polyhedra Based on the Simplex Method*, Xavier Allamigeon and Ricardo D. Katz present a formalization in Coq of the theory of convex polyhedra using the MathComp library. The authors use a complete formalization of the simplex method, together with the proof of its correctness and termination. This formalization provides an effective way to determine the feasibility of a polyhedral region, and in the case of optimization, the minimum value or a proof of the unboundedness of the objective function. The formal development includes important results such as the duality theorem, Farkas' lemma, and Minkovski separation theorem.

The paper by Alexander Bentkamp, Jasmin Christian Blanchette, and Dietrich Klakow, *A Formal Proof of the Expressiveness of Deep Learning*, presents a formalization in Isabelle/HOL of a recent mathematical result, formulated by Cohen et al. This result concerns the higher expressiveness of deep learning over shallow learning for one specific architecture called convolutional arithmetic circuits (CAC). Cohen et al. result states that CAC shallower networks must be exponentially larger than deeper networks expressing the same function. A

✉ Mauricio Ayala-Rincón
ayala@unb.br

César Muñoz
cesar.a.munoz@nasa.gov

1 Universidade de Brasília, Brasilia, DF, Brazil

2 NASA, Hampton, VA, USA

special case of this result compares the deepest possible network with the shallowest one. The formalization required the development of a library of tensors and contributions to theories of matrices and polynomials specifically on Lebesgue measure.

In *CompCertS: A Memory-Aware Verified C Compiler using Pointer as Integer Semantics*, Frédéric Besson, Sandrine Blazy, and Pierre Wilke present an extension of CompCert with symbolic values. CompCert is a C compiler verified in Coq, but its correctness certificate only applies to programs that do not result in undefined behavior. This means that the compiler can produce miscompilation for programs that violate the rules of the memory model. The extension, named CompCertS, enhances CompCert by avoiding miscompilation of programs that perform pointer arithmetic. The extension deals with bounded memory consumption in such a manner that if the source program does not run out of memory, then the target program is guaranteed not to run out of memory too.

Yannick Forster and Gert Smolka, in *Call-by-Value Lambda Calculus as a Model of Computation in Coq*, present the programming language L, a subsystem of the λ-calculus with a weak head and call-by-value reduction strategy. This calculus stands as a minimal functional programming language as well as as a Turing complete model of computation. For this language, several classical results of computability theory, such as Scott's theorem, Rice's theorem, Post's theorem, among others, were formalized in Coq.

In *Categoricity Results and Large Model Constructions for Second-Order ZF in Dependent Type Theory*, Dominik Kirst and Gert Smolka provide a Coq proof of Zermelo's embedding theorem for models, having as consequence that ZF is categorical in all cardinalities. It refines Zermelo's results by providing an extension that by fixing the number of Gorthendieck universes obtains this embedding internally. The development assumes excluded middle, and specifies and formalizes the properties of models of second-order ZF set theory in the type theory of Coq. As a consequence, the formalization does not use (transfinite) ordinals directly.

Andreas Lochbihler, in *Effect Polymorphism in Higher-Order Logic*, provides a "proof pearl" formalization of monads in Isabelle/HOL. In general, monads cannot be formalized as type operators in HOL since the general monad laws use such operators with different types. As a partial solution to this problem, the formalization restricts the laws for an arbitrary monad to a single type instance. This formalization gives up type polymorphism but keeps what is called effect polymorphism. Several extensions of the monad laws are considered including the treatment of exceptions, state, probabilistic choice, and non-determinism. Concrete implementations are provided and proven to satisfy the specifications.

The paper *A Verified Generational Garbage Collector for CakeML*, by Adam Sandberg Ericsson, Magnus O. Myreen, and Johannes Åman Pohjola, describes the verification of a generational copying garbage collector algorithm for the CakeML compiler. The formalization of the garbage collection problem is stated using HOL4 definitions, and an analysis of the correctness of a generational garbage collector is confirmed using the HOL4 theorem prover. The verification is structured into a proof of safety for the (abstract) collector algorithm and a proof that the implementation matches that algorithm. The key idea for handling partial collections is to model them as full collections on a modified heap in which pointers to old generations are treated as non-movable constants. The problem of tracking references from old to new generations is handled by allocating all mutable data separately and using the entire mutable data set as part of the root set.

In *Verifying a Concurrent Garbage Collector with a Rely-Guarantee Methodology*, Yannick Zakowski, David Cachera, Delphine Demange, Gustavo Petri, David Pichardie, Suresh Jagannathan, and Jan Vitek formally verify in Coq a concurrent garbage collector. The contributions include developing a suitable compiler purpose-built language called RtIR meant to be used as an intermediate representation in a future compiler stack, balancing low-level

features for executability and high-level features to ease proofs, devising a rely-guarantee program logic that allows for an incremental proof and decouples single-thread verification from reasoning about thread interference, and proving correctness of a lock-free concurrent garbage collector implemented in this RtIR. In addition to the rely-guarantee program logic for RtIR, the proof technology involves a technique that allows invariants to be progressively defined.

Finally, Bohua Zhan, in *Formalization of the Fundamental Group in Untyped Set Theory Using Auto2*, proposes a formal framework for formalizing mathematics in untyped set theory by using the prover auto2. The auto2 prover is packaged as a tactic in Isabelle. The proposed framework allows the entire development chain to be formalized in Isabelle/FOL with a high automation level. Besides the formalization of untyped set theory, the article discusses some tool improvements and strategies that are suitable for the addressed domain of untyped set theory. A main contribution of the article is to show that auto2 scales well to relatively large scale theories while being able to manage the complexity of using untyped set theory for the purpose of formalizing mathematics.

Last but not least, we wish to thank the Editor-in-Chief, Tobias Nipkow, and the editorial office of Springer Nature for their collaboration in producing this special issue, the reviewers for their thoughtful reviews, and the authors for contributing their papers.