AFRL-AFOSR-VA-TR-2022-0037

Foundations of Type Theory for Computation and Mathematics

Andrej Bauer
UNIVERSITY OF LJUBLJANA
KONGRESNI TRG 12
LJUBLJANA, , 1000
SI

09/02/2021
Final Technical Report

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory
Air Force Office of Scientific Research
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
02-09-2021	Final	15 Aug 2017 - 14 Jan 2021

4. TITLE AND SUBTITLE
Foundations of Type Theory for Computation and Mathematics

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-17-1-0326

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Andrej Bauer

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF LJUBLJANA
KONGRESNI TRG 12
LJUBLJANA, 1000
SI

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTA2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-VA-TR-2022-0037

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Our research group worked on topics in foundations of type theory, homotopy type theory, and theory of programming languages. This abstract provides an overview of our research for the entire duration of the project. Please refer to the attached document for a more detailed description of our work, lists of external collaborators and publications, and summary of other scientific activities by our research group.
Foundations of type theory. We developed a comprehensive meta-theory of general type theories. We gave a precise definition of what type theories are and identified properties that a type theory must have in order to be well-behaved. These results provide a frame-work within which type theories may be systematically studied, developed, and designed. Large parts of the development were formalized in the Coq proof assistant.
We also developed a generic proof assistants, Andromeda 2, which implements (the finitary variant of) general type theories. The proof assistant allows the user to specify any standard type theory, and is equipped with a general user-extensible equality checker. We used the techniques from programming language design, namely algebraic effects and han-dlers, to allow the user to easily extend the proof assistant with custom proof development techniques.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			TRISTAN NGUYEN
U	U	U	UU	5	**19b. TELEPHONE NUMBER** *(Include area code)* 426-7796

Standard Form 298 (Rev.8/98)
Prescribed by ANSI Std. Z39.18

PROJECT FINAL REPORT:
"FOUNDATIONS OF TYPE THEORY FOR COMPUTATION AND MATHEMATICS"
(TYPECOMA)

Project No.: **FA9550-17-1-0326**
Reporting period: **August 15, 2017 – December 31, 2020**
Principal investigator: **prof. dr. Andrej Bauer**
AFOSR program officer: **Dr. Tristan Nguyen**

1. Abstract

[This abstract is identical to the one that was submitted in the online form.]

Our research group worked on topics in foundations of type theory, homotopy type theory, and theory of programming languages. This abstract provides an overview of our research for the entire duration of the project. Please refer to the attached document for a more detailed description of our work, lists of external collaborators and publications, and summary of other scientific activities by our research group.

Foundations of type theory. We developed a comprehensive meta-theory of general type theories. We gave a precise definition of what type theories are and identified properties that a type theory must have in order to be well-behaved. These results provide a framework within which type theories may be systematically studied, developed, and designed. Large parts of the development were formalized in the Coq proof assistant.

We also developed a generic proof assistants, Andromeda 2, which implements (the finitary variant of) general type theories. The proof assistant allows the user to specify any standard type theory, and is equipped with a general user-extensible equality checker. We used the techniques from programming language design, namely algebraic effects and handlers, to allow the user to easily extend the proof assistant with custom proof development techniques.

Algebraic effects and handlers. We further developed and applied the theory of algebraic effects and handlers, which is an approach to mathematical modeling of program behavior. It is particularly well-suited for analyzing how programs interact with their external environments through the so-called computational effects.

We devised an effect system (a type theory for computational effects) with explicit subtyping and used it to developed new optimization techniques that compile algebraic handlers into efficient low-level code. We verified the viability of our approach by implemented the techniques in the prototype language Eff.

We formulated a notion of local algebraic effects that makes it possible to reason about effectful code in a compositional fashion. Our results make it possible to analyze the behavior of each program component separately, using local algebraic theories, rather than in a monolithic fashion.

We developed a semantic framework for reasoning about programs exhibiting computational effects based on the notion of Dijkstra monads. These naturally combine computational monads (such as those for state, exceptions, nondeterminism, IO, etc) with specification monads (such as monads of predicate transformers, and pre- and postconditions). Parts of the work were implemented in F-star, a tool for formal specification and verification of software.

We studied directed containers, and their comonadic semantics. Every (directed) container morphism can be canonically factored into two (directed) container morphisms, one which is identity on shapes and one that is identity on positions. Moreover, this construction generalizes to arbitrary comonads preserving certain Cartesian natural transformations, and that this factorization is the cofunctor variant of the standard full image factorization of ordinary functors.

We gave an account of asynchronous algebraic effects. These enable non-blocking operations whose continuations may continue without waiting for the result of an operation. We implemented a prototype language Æff which demonstrates how asynchronous algebraic effects can be used to implement various features of concurrent computations.

We used comodels of algebraic theories as a means to model program's external environments in a modular and user-definable way. We developed new programming concepts, called runners, which provide a general mechanism for resource management and encapsulation of computations in virtual environments. We also implemented a prototype language Coop, and a Haskell library to showcase the possibilities offered by runners.

Homotopy type theory. We developed topics in synthetic homotopy theory theory, which is the study of classical homotopy theory reformulated in the framework of homotopy type theory.

We gave a comprehensive treatment and characterization of lex modalities in homotopy type theory, thereby furthering our understanding of the structure of universes and modal operators in type theory. We also proved a modal descent theorem in homotopy type theory that reveals a connection between modalities in homotopy type theory and modal étale maps. We applied the results to real-cohesive type theory and explained how the modal étale maps relate to the formally étale maps from algebraic geometry.

We studied n-groups, a higher homotopical analogue of groups, and introduced the related notion of n-exactness for short exact sequences of pointed types

We initiated a study of finite discrete structures in homotopy type theory. We generalized the notion of a finite type to higher-finite types whose (truncations of) higher identity types are finite, and characterized how higher finiteness interacts with several type-theoretic constructions.

2. Performance report for the period of no-cost extension (August 15–December 31, 2020)

We provide below a summary of activities for the no-cost extension, granted for the period from August 15th to December 31st 2020. Please refer to annual performance reports submitted in 2018, 2019 and 2020 for detailed description of publications and project activities during those periods.

We have had one peer-reviewed publication [AP21], and one preprint [BP21] (submitted for publication).

2.1. **Personnel associated with the period of performance.**

- Dr. Andrej Bauer, principal investigator
- Dr. Matija Pretnar, senior researcher
- Dr. Egbert Rijke, postdoctoral researcher
- Dr. Katja Berčič, postdoctoral researcher
- Anja Petković, PhD student
- Žiga Lukšič, PhD student (graduated in November 2020)

2.2. **Mitigation of the impact of COVID-19 epidemic.** The COVID-19 epidemic has a continued impact on our research activities. With research travel not being possible, we focused on on-line activities that allow us to connect with the research community.

Our local research seminar takes place online. We continued to organize a series of talks *"Every proof assistant"* [Eve20] which attracted a sizable international audience, averaging around 70 participants per seminar. The seminars are devoted to presentation of proof assistants and other formalization tools.

Members of our research group participated in other online seminars and on-line versions of conferences.

2.3. **Doctoral students.** Žiga Lukšič successfully defended his doctoral dissertation in November 2020. After graduation he continued to work as a research and teaching assistant at the Department of Mathematics, University of Ljubljana. In the Spring 2021 he joined a software company in Slovenia.

Anja Petković continues the final year of her doctoral studies. She will graduate by the end of 2021.

2.4. **Leadership and community service.** Members of our research group participated as referees and program committee members in a number of international conferences and journals.

Anja Petković is a regular guest on the "Ugriznimo znanost" ("Let's bite science") popular science show on the Slovenian national TV. She helps popularize mathematics and science by presenting math puzzles suitable for the general population.

Matija Pretnar became a full member of the IFIP Working Group 2.1 Algorithmic Languages and Calculi, an international research group dedicated to exploration and evaluation of new ideas in the field of programming and design of new languages.

Andrej Bauer has been trusted with the following leadership roles:

(1) Andrej Bauer continues to serve as a member of the Steering Committee of the ACM SIGPLAN International Conference on Functional Programming (ICFP).
(2) Andrej Bauer continues to be a member of the Steering Committee of the international conference on Mathematical Foundations of Programming Semantics.
(3) Andrej Bauer continues to serve as the head of Theoretical computer science department at the Institute for mathematics, physics, and mechanics (Ljubljana, Slovenia).
(4) Andrej Bauer continues to serve as a member of the Habilitation committee of the University of Ljubljana, a university-wide body overseeing all academic promotions.

3. Summary of publications for the reporting period

We provide below the summary of publications and other contributions for the entire duration of the project. These were also reported in seperate annual performance reports.

3.1. Archival publications for the period from August 15th 2017 to August 14th 2018.

- peer-reviewed publications: [Bau17], [FKLP17], [SKPS18],
- preprints: [BGH+18], [Bau18], [BS18], [PSFS17].

3.2. Archival publications for period from August 15th 2018 to August 14th 2019.

- peer-reviewed publications: [AU18], [MAA+19], [BS19], [SV18], [Voo18],
- preprints: [Voo19], [SV19]

3.3. Archival publications for period from August 15th 2019 to August 14th 2020.

- peer-reviewed publications: [AB20c, LP20, KPS+20, SvDR20]
- preprints: [CR20b, CR20a, BR19, AP19]
- other conference and workshop presentations: [BHP20a, BH20, BHP20b, BPS20]
- software: [AP20, AB20a, AB20b, Luk20a, Luk20b, Bau20, And20, Eff20]

3.4. Archival publications for period from August 15th 2020 to December 31st 2020.

- peer-reviewed publication: [AP21]
- preprint: [BP21].

References

[AB20a] Danel Ahman and Andrej Bauer. Coop. https://github.com/andrejbauer/coop, 2020.

[AB20b] Danel Ahman and Andrej Bauer. Haskell-coop. https://github.com/danelahman/haskell-coop, 2020.

[AB20c] Danel Ahman and Andrej Bauer. Runners in action. In *Programming Languages and Systems – 29th European Symposium on Programming, ESOP 2020*, volume 12075 of *Lecture Notes in Computer Science*, pages 29–55. Springer, 2020.

[And20] Andromeda 2 proof assistant. https://github.com/Andromedans/andromeda, 2020.

[AP19] Danel Ahman and Matija Pretnar. Asynchronous effects. arXiv:2003.02110, December 2019.

[AP20] Danel Ahman and Matija Pretnar. Æff. https://github.com/matijapretnar/aeff, 2020.

[AP21] Danel Ahman and Matija Pretnar. Asynchronous effects. *Proceedings of ACM on Programming Languages*, 5(POPL):1–28, 2021.

[AU18] Danel Ahman and Tarmo Uustalu. Decomposing comonad morphisms. In *8th Conference on Algebra and Coalgebra in Computer Science CALCO 2019, London*, June 2018.

[Bau17] Andrej Bauer. On fixed-point theorems in synthetic computability. *Tbilisi Mathematical Journal*, 10(3):167–181, 2017.

[Bau18] Andrej Bauer. What is algebraic about algebraic effects and handlers?, 2018. Preprint arXiv:1807.05923.

[Bau20] Andrej Bauer. Clerical – an imperative language for real-number computation. https://github.com/andrejbauer/clerical, 2020.

[BGH+18] Andrej Bauer, Gaëtan Gilbert, Philipp G. Haselwarter, Matija Pretnar, and Christopher A. Stone. Design and implementation of the andromeda proof assistant, 2018. Preprint arXiv:1804.06217.

[BH20] Andrej Bauer and Philipp G. Haselwarter. Finitary general type theories in computational form. In *26th International Conference on Types for Proofs and Programs, EUTYPES-TYPES*, Turin, Italy, March 2020.

[BHP20a] Andrej Bauer, Philipp G. Haselwarter, and Anja Petković. Equality checking for general type theories in Andromeda 2. In *Mathematical Software - ICMS 2020 - 7th International Conference, Braunschweig, Germany, July 13-16, 2020, Proceedings*, volume 12097 of *Lecture Notes in Computer Science*, pages 253–259. Springer, 2020.

[BHP20b] Andrej Bauer, Philipp G. Haselwarter, and Anja Petković. On equality checking for general type theories: Implementation in andromeda 2. In *26th International Conference on Types for Proofs and Programs, EUTYPES-TYPES*, Turin, Italy, March 2020.

[BP21] Andrej Bauer and Anja Petković. An extensible equality checking algorithm for dependent type theories. arXiv:2103.07397, March 2021.

[BPS20] Andrej Bauer, Sewon Park, and Alex K. Simpson. Clerical: an imperative language for verified real-number computation. In *Seventeenth International Conference on Computability and Complexity in Analysis (CCA)*, Online conference, September 2020.

[BR19] Ulrik Buchholtz and Egbert Rijke. The long exact sequence of homotopy n-groups. arXiv:1912.08696, December 2019.

[BS18] Andrej Bauer and Andrew Swan. Every metric space is separable in function realizability, 2018. Preprint arXiv:1804.00427.

[BS19] Andrej Bauer and Andrew Swan. Every metric space is separable in function realizability. *Logical Methods in Computer Science*, 15(2), 2019.

[CR20a] Felix Cherubini and Egbert Rijke. Modal descent. arXiv:2003.09713, March 2020.

[CR20b] J. Daniel Christensen and Egbert Rijke. Characterizations of modalities and lex modalities. arXiv:2008.03538, August 2020.

[Eff20] Eff programming language. https://github.com/matijapretnar/eff, 2020.

[Eve20] Every proof assistant (online seminar series). http://math.andrej.com/category/every-proof-assistant/, April–June 2020.

[FKLP17] Yannick Forster, Ohad Kammar, Sam Lindley, and Matija Pretnar. On the expressive power of user-defined effects: effect handlers, monadic reflection, delimited control. *PACMPL*, 1(ICFP):13:1–13:29, 2017.

[KPS+20] Georgios Karachalias, Matija Pretnar, Amr Hany Saleh, Stien Vanderhallen, and Tom Schrijvers. Explicit effect subtyping. *Journal of Functional Programming*, 30:e15, 2020.

[LP20] Žiga Lukšič and Matija Pretnar. Local algebraic effect theories. *Journal of Functional Programming*, 30:e13, 2020.

[Luk20a] Žiga Lukšič. Formalization of eeff. https://github.com/zigaLuksic/eeff-formalization, 2020.

[Luk20b] Žiga Lukšič. Implementation of eeff. https://github.com/zigaLuksic/eff/tree/EEFF, 2020.

[MAA+19] Kenji Maillard, Danel Ahman, Robert Atkey, Guido Martínez, Cătălin Hriţcu, Exequiel Rivas, and Éric Tanter. Dijkstra monads for all. *Journal Proceedings of the ACM on Programming Languages*, 3(ICFP):104:1–104:29, July 2019.

[PSFS17] Matija Pretnar, Amr Hany Saleh, Axel Faes, and Tom Schrijvers. Efficient compilation of algebraic effects and handlers. Technical Report CW 708, KU Leuven Department of Computer Science, 2017.

[SKPS18] Amr Hany Saleh, Georgios Karachalias, Matija Pretnar, and Tom Schrijvers. Explicit effect subtyping. In Amal Ahmed, editor, *Programming Languages and Systems - 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10801 of *Lecture Notes in Computer Science*, pages 327–354. Springer, 2018.

[SV18] Alex Simpson and Niels F. W. Voorneveld. Behavioural equivalence via modalities for algebraic effects. In *Programming Languages and Systems - 27th European Symposium on Programming, ESOP 2018 Thessaloniki, Greece, April 14-20, 2018*, pages 300–326, 2018.

[SV19] Alex Simpson and Niels F. W. Voorneveld. Behavioural equivalence via modalities for algebraic effects. *CoRR*, abs/1904.08843, 2019.

[SvDR20] Kristina Sojakova, Floris van Doorn, and Egbert Rijke. Sequential colimits in homotopy type theory. In *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany*, pages 845–858. ACM, July 2020.

[Voo18] Niels F. W. Voorneveld. Non-deterministic effects in a realizability model. *Electronic Notes Theoretical Computer Science*, 336:299–314, 2018.

[Voo19] Niels F. W. Voorneveld. Quantitative logics for equivalence of effectful programs. *CoRR*, abs/1904.11771, 2019.