

Curriculum Vitae

Full name: Andreas Martin Abel
Nationality: German
ORCID: 0000-0003-0420-4492

Employment and Education	2
Academic and Visiting Positions	2
Education	2
Degrees	3
Educational Fellowships	3
Industrial Involvement and Programming	3
Academic Merits and Service	4
Research Grants and Sabbaticals	4
Academic Leadership	4
Research administration and steering	4
Academic self-administration	4
Doctoral committee memberships	4
PC chairing	5
PC memberships	5
Additional reviewing	6
Event organization	7
Invitations	7
Invited Talks	7
Invitations to Working Meetings	7
Research	9
Main Line of Research.	9
Major Previous Research Topics.	10
Other Research Interests.	11
Publications	12
Peer-Reviewed Publications by Category	12
Bibliometrics	12
List of Publications	13
Teaching and Supervision	20
Courses in Summer Schools, Upon Invitation, Tutorials.	20
Teaching at Gothenburg University	20
Teaching at the University of Munich	20
Assistance in Teaching.	21
Supervision of Students in Seminars.	21
Supervised PhD Students	22
Supervised Diploma and Master Students	22
Supervised Bachelor Students and Student Projects	23
Supervised Industrial Master Theses.	23
Supervised Student-Proposed Bachelor Theses	24
Contact	25
References	25
Address	26

Academic and Visiting Positions

- Oct 2013 – Today *Universitetslektor (Senior Lecturer)*
Department of Computer Science and Engineering
Gothenburg University and Chalmers, Sweden
- Oct 2009 – March 2010 **Invited researcher**
PI.R2 project (πr^2) of INRIA Rocquencourt and
PPS (CNRS lab Programs, Proofs, and Systems)
Paris, France
- Oct 2008 – Sep 2013 *Akademischer Rat auf Zeit (A13)*
Assistant Professor as civil servant (*Beamter*)
Theoretical Computer Science (chair M. Hofmann)
Ludwig-Maximilians-University Munich
- Oct 2005 – Sep 2008 *Wissenschaftlicher Mitarbeiter (A13)*
Assistant Professor
Theoretical Computer Science (chair M. Hofmann)
Ludwig-Maximilians-University Munich
- Jan 2004 – Oct 2005 **Postdoctoral researcher**
Department of Computer Science
Chalmers University of Technology, Göteborg, Sweden
- Apr 2002 – Oct 2002 *Wissenschaftlicher Mitarbeiter (A13)*
Assistant
Theoretical Computer Science (chair M. Hofmann)
Ludwig-Maximilians-University Munich
- May 2000 – June 2001 **Visiting researcher**
hosted by Prof. F. Pfenning
Department of Computer Science
Carnegie-Mellon University, Pittsburgh, USA

Education

- 1999 – 2003 *Doktorand im Graduiertenkolleg Logik in der Informatik*
(Student in the PhD Program Logic in Computer Science)
Ludwig-Maximilians-University Munich
- 1994 – 1999 *Studium der Informatik* (Student in Computer Science)
Ludwig-Maximilians-University Munich

Degrees

- 2013 *Privatdozent (habilitation)*
Faculty of Mathematics, Computer Science, and Statistics, University of Munich
Thesis: *Normalization by Evaluation: Dependent Types and Impredicativity* [23]
Committee: Peter Dybjer, Martin Hofmann, Helmut Schwichtenberg
- 2006 *Doctor rerum naturalium (Ph.D.)*
Department of Computer Science, University of Munich
Thesis: *Type-Based Termination*
A Polymorphic Lambda-Calculus with Sized Higher-Order Types [56]
Supervisor: Martin Hofmann
Mark: *magna cum laude*
- 1999 *Informatik Diplom (M.Sc.)*
University of Munich
Thesis: *A Semantic Analysis of Structural Recursion* [76]
Mark: *sehr gut* (best mark)

Educational Fellowships

- 2004–2005 **Postdoctoral fellowship**
from the Swedish Foundation of Strategic Research (SFF)
in the CoVer Project, Chalmers, Gothenburg, Sweden
- 2000–2001 **Fellowship** (research scholar) from the Office of Technology in Education,
Carnegie-Mellon University, Pittsburgh, USA
- 1999–2003 **Stipend** from PhD Program Logic in Computer Science,
Deutsche Forschungsgemeinschaft (DFG) (German Research Foundation)

Industrial Involvement and Programming

- | | | |
|------------|---|---------|
| 2004–today | Open-source developer in the <i>Agda</i> team (me: 4300 patches) | Haskell |
| 2017–today | Maintainer of the <i>Backus Naur Form Compiler</i> (BNFC) | Haskell |
| 2011–2012 | External consultant for <i>Glueware Informatik GmbH</i> , Munich
(framework for business application components) | Scala |
| 2007–2013 | Developing the research prototype <i>MiniAgda</i> | Haskell |
| 2000–2001 | Development of the tutorial proof checker <i>Tutch</i> | SML |
| 1990–1998 | Software development for the publishing house
<i>Park Körner</i> , Munich | Delphi |

Further programming skills: Java, C, OCaml, Python, scripting

Research Grants and Sabbaticals

- 2018–2021 **Research project** *Syntax and Semantics of Univalent Type Theory* (VR 2017-04064)
4 100 000 SEK (450 000 EUR) granted by Vetenskapsrådet, Stockholm
PI: Thierry Coquand. I am a co-applicant.
- 2015–2018 **Research project** *Termination Certificates for Dependently-Typed Programs and Proofs via Refinement Types* (VR 2014-04864)
3 600 000 SEK (400 000 EUR) granted by Vetenskapsrådet, Stockholm
I am sole applicant and principal investigator.
- 2009–2010 **Invited researcher** (6 months) of INRIA, France
- 2008–2009 **Research cooperation** with LORIA, Nancy, France, on *Type-Based Termination*, supported with travel funds by the *Bayerisch-französisches Hochschulzentrum*
- 2005–2007 **Research project** *Typed Lambda Calculi and Applications*
1 800 000 SEK granted by Vetenskapsrådet, Stockholm
Co-application with Prof. Thierry Coquand and Prof. Peter Dybjer

Academic Leadership

Research administration and steering.

- Member of the steering committee of the *International Conference on Types for Proofs and Programs* (TYPES), since May 2018.
- EU Coordination Action CA15123 EUTYPES, Types for Programming and Verification: member of the core group and leader of working group WG3: Types for Programming (since 2016).
- International workshop series LFMTP: Member of the steering committee Nov 2009 – Oct 2014. Chair Jul 2010 – Oct 2014.
- EU Coordination Action FP6-2002-IST-C TYPES, Types for Proofs and Programs (510996): administrator of the Munich-LMU subsite (headed by Helmut Schwichtenberg) from Dec 2005 to Jun 2009.

Academic self-administration.

- In 2017, vice head of division Computer Science at the Department of Computer Science and Engineering, Chalmers and Gothenburg University.
- Coordinator of teaching of the chair of Theoretical Computer Science, LMU Munich, from 2005 to 2010.

Doctoral committee memberships.

 Member of the following doctoral committees:

1. Johan Granström, *Reference and Computation in Intuitionistic Type Theory*, Uppsala University, Sweden, Jan 2009.

2. Cody Roux, *Type Based Termination: Semantics and Generalisations*, LORIA, Nancy, France, June 2011.
3. Jorge Sacchini, *On Type-Based Termination and Dependent Pattern Matching in the Calculus of Inductive Constructions*, École des Mines de Paris (MINES ParisTech), France, June 2011.
4. Ronan Saillard, *Type Checking in the $\lambda\Pi$ -Calculus Modulo: Theory and Practice*, École des Mines de Paris (MINES ParisTech), France, September 2015(rapporteur).
5. Guilhem Moulin, *Internalizing Parametricity*, Chalmers University of Technology, Sweden, May 2016 (substitute member).
6. Jesper Cockx, *Dependent pattern matching and proof-relevant unification*, KU Leuven, Belgium, 17 May 2017.
7. Rodolphe Lepigre, *Semantics and Implementation of an Extension of ML for Proving Programs*, Université de Savoie, France, 18 July 2017.
8. Sandro Stucki, *Higher-Order Subtyping with Type Intervals*, EPFL, Lausanne, Switzerland, 15 September 2017.
9. Matus Tejsicak, *Erasure in Dependently-Typed Programming*, St. Andrews University, Scotland, UK, 15 February 2018.
10. Henning Basold, *Mixed Inductive-Coinductive Reasoning*, Radboud Universiteit Nijmegen, The Netherlands, 19 April 2018.
11. Stephan Adelsberger, *A Formal Verification Framework for Object-Based Programming in Agda* Wirtschaftsuniversität Wien (Vienna University of Economics and Business), 20 July 2018.
12. Marco Vassena, *Verifying Information Flow Control Libraries*, Chalmers University of Technology, Gothenburg, Sweden, 15 February 2019.

PC chairing. Co-chair of program committee of international workshops LFMTP 2008 and PLPV 2013.

PC memberships. Member of the program committee of the following international conferences:

- APLAS 2016 and 2017 *Asian Symp. on Prog. Lang. and Systems*
- CPP 2015 *Certified Programs and Proofs*
- ESOP 2012 and 2016 *European Symp. on Programming*
- FLOPS 2016 *Int. Symp. on Functional and Logic Programming*
- FoSSaCS 2010, 2013, 2015, and 2018 *Found. of Software Sci. and Comput. Struct.*
- ICALP 2015 (Track B) *Int. Colloq. on Automata, Lang. and Programming*
- ITP 2012, 2018 and 2019 *Interactive Theorem Proving*
- Haskell Symposium 2013
- LiCS 2020 *Logics in Computer Science*
- MFCS 2011 *Mathematical Foundations of Computer Science*
- POPL 2015 (ext. rev. committee), 2019 *Principles of Programming Languages*
- POPL 2019 distinguished papers
- PPDP 2017 *Principles and Practice of Declarative Programming*

- RTA 2012 *Rewriting Techniques and Applications*
- TLCA 2013 *Typed Lambda Calculi and Applications*
- RTA-TLCA 2014, FSCD 2016 and 2017 *Formal Struct. for Comput. and Ded.*
- SOFSEM-FOCS 2017 *Int. Conf. on Current Trends in Theory and Practice of Computer Science – Foundations of Computer Science Track*

Member of the editorial committee of the TYPES 2011 and 2017 post-proceedings.

Member of the program committees of the international workshops

- CMCS 2014 *Coalgebraic Methods in Computer Science*
- DTP 2014 *Dependently Typed Programming*
- FICS 2013 *Fixed Points in Computer Science*
- HOR 2012 *Higher-Order Rewriting*
- LFMTTP 2007, 2014, 2015 *Log. Frameworks. and Meta Lang.: Theory and Pract.*
- MSFP 2010, 2014, 2018 *Mathematically Structured Functional Programming*
- PAR 2010 *Partiality and Recursion*
- PARIS 2018 *Programming And Reasoning on Infinite Structures*
- PEPM 2015 *Partial Evaluation and Program Manipulation*
- PLPV 2009 *Programming Languages meet Program Verification*
- TTT 2017 *Type Theory Based Tools*
- TYPES 2014, 2017 and 2018 *Types for Proofs and Programs*
- WGP 2013 *Workshop on Generic Programming*

Member of the program committees of the national workshops

- ATPS 2015 *Arbeitstagung Programmiersprachen*
- HaL 2016 *Haskell in Leipzig*

Additional reviewing. Review of more than 120 research papers for the conferences

- APLAS 2007, 2010, and 2013 *Asian Symp. on Prog. Lang. and Systems*
- CADE 2001, 2003, and 2008 *Conf. on Automated Deduction*
- CiE 2008, 2010, and 2018 *Computability in Europe*
- CPP 2018 *Certified Programs and Proofs*
- CSL 2001, 2002, 2008, 2009, 2010, 2011, and 2016 *Computer Science Logic*
- ESOP 2007, 2009, 2013, and 2017 *European Symp. on Programming*
- FoSSaCS 2008 *Foundations of Software Sci. and Comput. Structures*
- FLOPS 2010 *Int. Symp. on Functional and Logic Programming*
- FSTTCS 2000 *Foundations of Software Tech. and Theoret. Comput. Sci.*
- ICALP 2004 and 2008 *Int. Colloq. on Automata, Lang. and Programming*
- ICFP 2013 and 2015 *Int. Conf. on Functional Programming*
- LiCS 2008, 2011, 2013, 2014, 2017, and 2018 *Logics in Computer Science*
- LPAR 2005, 2006, and 2010 *Int. Conf. on Logic for Prog., AI and Reasoning*
- MFCS 2014 *Mathematical Foundations of Computer Science*
- MPC 2008 *Mathematics of Program Construction*
- POPL 2008, 2009, 2013, 2014, and 2018 *Principles of Programming Languages*
- PPDP 2007 and 2009 *Principles and Practice of Declarative Programming*
- RTA 2004, 2007, 2009, and 2011 *Rewriting Techniques and Applications*
- STACS 2007 *Symp. on Theoretical Aspects of Computer Science*
- TLCA 2003 and 2015 *Typed Lambda Calculi and Applications*

the post-workshop proceedings

- CL&C 2009 *Classical Logic and Computation*
- TFP 2005 and 2008 *Trends in Functional Programming*

- TYPES 2002, 2006, 2007, 2008, 2010, 2013, and 2014

Types for Proofs and Programs

and the journals:

- Fund. inform. *Fundamenta Informaticæ*
- HOSC *Higher-order and Symbolic Computation*
- IGPL *Logic Journal of the IGPL*
- IPL *Information Processing Letters*
- JAR *Journal of Automated Reasoning*
- JFP *Journal of Functional Programming*
- LMCS *Logical Methods in Computer Science*
- MSCS *Mathematical Structures in Computer Science*
- TCS *Theoretical Computer Science*

Event organization. Organization of the Agda Implementor’s Meeting (AIM):

- AIM XXI, Gothenburg, June 2015.
Co-organization with Bengt Nordström and Andrea Vezzosi.
- AIM XIX, Paris, May 2014.
- AIM XV, Fischbachau, Germany, February 2012.

Co-organization of the international workshops LFMTTP 2008, Pittsburgh, colocated with LICS 2008, and PLPV 2013, Rome, affiliated to POPL 2013.

Invitations

Invited Talks.

- 2016 **Keynote** at the international workshop *Coalgebraic Methods in Computer Science* (CMCS 2016), Eindhoven, The Netherlands
- 2014 **Invited presentation** at the international workshop on *Certification of High-Level and Low-Level Programs*, part of the Institut Henri Poincaré thematic trimester Semantics of proofs and certified mathematics, Paris, France.
- 2012 **Keynote** at the international workshop *Fixed-Points in Computer Science* (FICS 2012), Tallinn, Estonia

Invitations to Working Meetings.

- 2018 IFIP WG 1.3 *Foundations of System Specification*, Royal Holloway, London (*observer*)
- 2017 *Big Proof*: 6-week program at the Isaac Newton Institute for Mathematical Sciences in Cambridge
- 2017 IFIP WG 1.3 *Foundations of System Specification*, Binz, Germany (*observer*)
- 2016 Dagstuhl-Seminar 16421 *Universality of Proofs*
- 2016 Dagstuhl-Seminar 16131 *Language Based Verification Tools for Functional Programs*
- 2015 IFIP WG 2.1 *Algorithmic Languages and Calculi* in Göteborg (*local observer*)

- 2014 *Workshop on Certification of High-Level and Low-Level Programs*,
part of the thematic trimester *Semantics of proofs and certified mathematics* at
the Henri Poincaré Institute in Paris
- 2014 *Representing Streams II* at the Lorentz center in Leiden
- 2011 Shonan Seminar 007 *Dependently Typed Programming* in Japan
- 2004 Dagstuhl-Seminar 04381 *Dependently Typed Programming*

Main Line of Research

My research interests span programming languages, type systems, evaluation and compilation, termination, semantics, verification, proof languages, and logical foundations. The unifying vision is an efficient and theoretically founded language for integrated specification, programming and verification. A specific candidate for such a language is Dependent Type Theory, and I have been looking at the theory and technology of dependent types which are the foundation of the functional languages and proof assistants *Coq* (developed by INRIA, France) and *Agda* (mainly developed by Chalmers, Sweden).

Beginning 2004 I have contributed to the design and implementation of the dependently-typed functional language *Agda*. Since 2010 my investment has been substantial. As of today (May 2019), I have submitted over 4800 patches (out of 17700 total) and I am one of the main developers (with Ulf Norell, Nils Anders Danielsson, Jesper Cockx, and Andrea Vezzosi). Besides fixing of bugs and design flaws, I have contributed entire components of *Agda*, including the following:

1. A *termination checker* based on structural ordering and size-change termination.
2. An extension to type-based termination checker using *sized types*.
3. A modality for proof *irrelevance*.
4. Higher-order unification modulo extensionality for records.
5. A new approach to records and coinduction using *copattern* matching.
6. Rewrite rules (with Jesper Cockx [14]).

All these components implement published research which will be detailed in the following.

Termination. A termination checker is an integral part of a proof assistant based on dependent types. With Thorsten Altenkirch, I have designed and verified a termination checker for strongly typed functional languages based on the structural ordering [76, 70, 73] and integrated it into *Agda*.

For my PhD, I have studied type systems for termination checking [63, 67, 57, 43] culminating in *a polymorphic lambda-calculus with sized higher-order types* [56]. I am continuing to explore the technological aspects of sized types in *Agda* [13] and my prototypical implementation *MiniAgda* [34, 27] of dependent type theory.

Coinduction. Processes and potentially infinite streams are modeled in type theory by coinduction. I have applied the structural ordering and sized types to checking the productivity of functional programs with a potentially infinite output [74], justified by a semantics based on orthogonality [49] and saturated sets [48].

With Brigitte Pientka, David Thibodeau, and Anton Setzer I have developed *copatterns*, a new approach to corecursion in functional programming and Type Theory [26, 25, 22, 12]. *Copatterns* unify the concepts of productivity of infinite objects and termination of recursive functions.

The addition of copatterns to *Agda* has been subjected to several case studies [20, 19, 18, 10, 9], to attest that they enable elegant coinductive programming and reasoning.

With my PhD student Andrea Vezzosi I have looked at guarded types [21] as an alternative method to assert productivity of lazy functional programs.

Computational irrelevance. In dependently typed languages part of the code is static, i.e., it serves to certify program properties, but has no computational effect at runtime. I have been researching type systems that classify static code as irrelevant for program equivalence and allow to discard it during compilation [30, 28, 15].

Higher-order unification. Unification is the basis for type reconstruction in dependently typed languages. With Brigitte Pientka I have researched an extension of higher-order unification to dependent record types [33].

Major Previous Research Topics

Recursion Schemes and Generic Programming. Rich type systems such as the higher-order polymorphic lambda calculus F^ω allow to abstract over programming schemes in a type safe and termination ensuring manner. With Ralph Matthes and Tarmo Uustalu, I have been investigating recursion schemes for higher-order and nested data types expressible in F^ω [68, 66, 62] and its extension by retract types [65]. Even more recursion schemes are handled by type-based termination [56] and some of them arise as instances of generic programs. I have researched criteria on generic programs which ensure that all of their instances are terminating [58, 39].

Dependent Type Theory and Logical Frameworks. During my 22 month visit to Chalmers (2004-2005), research has concentrated on dependent type theory and Martin-Löf's logical framework. With Thierry Coquand, I have proven completeness of an efficient algorithm for untyped beta-eta equality in an extension of the logical framework by Sigma-types (dependent pairs) [60, 52]. This framework, together with further extension by data types and recursive definitions, has been implemented by Ulf Norell and me under the name *agdaLight* and connected to the first-order automated theorem prover Gandalf [59].

η -Equality and Normalization-by-Evaluation. Starting in August 2004 I have been collaborating with Klaus Aehlig, Thierry Coquand, and Peter Dybjer on normalization-by-evaluation for dependent types. We have explored the idea for a Martin-Löf's Type Theory with untyped [51] and judgmental equality [53]. Our results imply decidability and injectivity of the dependent function type constructor in type theory with typed equality and universes. Using the here developed techniques we have managed to verify a type-directed algorithm for checking $\beta\eta$ -equality [46] as it is used in Agda and Epigram. I have presented a version of NbE for System F [44] simplifying previous approaches. This has been extended to System F^ω [40] and the Calculus of Constructions [35]. With Miguel Pagano and Thierry Coquand I have also treated NbE for singleton types and proof irrelevance [41]. My habilitation thesis *Normalization by Evaluation: Dependent Types and Impredicativity* [23] summarizes the result obtained in this post-doctoral research period.

Other Research Interests

Subtyping. I have found a new and short proof of completeness of algorithmic subtyping for the higher-order polymorphic lambda-calculus [55, 42]. Together with my student Dulma Rodriguez I have extended it to System F-sub with bounded quantification [47]. I am also interested in subtyping for dependently typed languages, especially its role in type-based termination.

Educational Proof Systems. During my stay at CMU I implemented “Tutch”, a *tutorial proof checker*, which verifies natural deduction proofs written by students in a block-structured syntax [72]. The system is used for course 15213 at CMU, for course G51MCS at the University of Nottingham, and others. I have been using Tutch for teaching *Computer-Aided Formal Reasoning*. Ideas from Tutch have found their way into Prof. Schwichtenberg’s *MINLOG* system at the University of Munich, and variants of Tutch for linear and classical logic have been developed at CMU.

Proof Languages. In Tutch some rudimentary ideas how to write formal machine-checkable proofs have been implemented. I am further interested in the design of practical proof languages for the construction of proof documents with a flexible proof granularity, which are both readable by the human and verifiable by a semi-automated proof checker. Together with Thierry Coquand and Ulf Norell [59] I have investigated connecting AgdaLight, and implementation of constructive type theory, to a first-order theorem prover (see also below).

Higher-Order Abstract Syntax. As a case study for interactive theorem proving with higher-order abstract syntax (HOAS), I encoded the lambda-mu calculus in Twelf [71]. I also formalized a combinatorial weak normalization proof for the simply-typed lambda-calculus [64]. Continuing this line of research, I want to explore how to extend the type-based termination paradigm to the negative data types which appear in encodings of languages with binding in HOAS.

Bidirectional Type-Checking. For programming languages with impredicative polymorphism, type inference is undecidable. I am interested in heuristics which support type checking for a class of programs which should encompass practically relevant programs. I implemented a prototypical bidirectional type-checker “curry” for the higher-order polymorphic lambda-calculus. With Thorsten Altenkirch I have proven soundness and partial completeness of a bidirectional semi-algorithm for type checking the inconsistent type theory Type:Type [31].

Program Verification. At Chalmers I have been working in the project *CoVer: Combining Verification Methods for Software Development*. We have concentrated on verifying Haskell programs using systematical testing with QuickCheck, and proving properties in First-Order Logics and Type Theory. To facilitate the latter we have developed a monadic translation of Haskell programs into Type Theory, which bridges the gap between the partial programming language and the total logical framework [61].

Peer-Reviewed Publications by Category

In international journals (15):

PACMPL 3 (ICFP 19) [2] PACMPL 3 (ICFP 18) [5]
PACMPL 2 (POPL 18) [4] PACMPL 1 (ICFP 17) [7] JFP 17 [6]
JFP 16 [12] LMCS 12 [28] LMCS 11 [29] SCP 09 [39] JFP 09 [38]
MSCS 08 [42] LMCS 08 [43] Fund.Inf. 07 [52] TCS 05 [62] ITA 04 [63]
JFP 02 [70]

In workshop proceedings published in international journals (4):

MSFP 08 [31] MFPS 07 [51] LFM 04 [64] MERLIN 01 [71]

In proceedings of international conferences (24+3):

(ICFP 19 [2]) LICS 19 [1] (ICFP 18 [5]) (POPL 18 [4]) (ICFP 17 [7])
TLCA 15 [16] APLAS 14 [21] RTA-TLCA 14 [22] ICFP 13 [25]
POPL 13 [26] TLCA 11 [33] FoSSaCS 11 [30] FLOPS 10 [35]
CSL 09 [40] TLCA 09 [41] LPAR 08 [44] CSL 08 [47] MPC 08 [46]
FLOPS 08 [45] APLAS 07 [48] LiCS 07 [53] TLCA 07 [49] CSL 06 [57]
MPC 06 [58] CSR 06 [55] TLCA 05 [60] CSL 04 [65] TLCA 03 [67]
FoSSaCS 03 [66]

In proceedings of international workshops (14):

UNIF 14 [17] MSFP 14 [19] LFMTP 11 [32] PAR 10 [34] LFMTP 10 [36]
NBE 09 [37] HOR 07 [50] MSFP 06 [54] Haskell 05 [61] FroCoS 05 [59]
TYPES 02 [68] PTP 01 [72] TYPES 99a [74] TYPES 99b [73]

Extended abstracts in international workshops (11):

TYPES 18 [3] TYPES 17 [8] TYPES 16 [14, 11, 13, 10] TYPES 15 [15]
NWPT 14 [20] NWPT 13 [24] FICS 03 [69] WST 99 [75]

Extended abstracts in national workshops (1):

ATPS 14 [18]

Keynotes (2):

CMCS 2016 [9] FICS 2012 [27]

Bibliometrics

As of May 2019, Google Scholar assigns me an h-index of 20 and an i10-index of 43.

List of Publications

- [1] Brigitte Pientka, David Thibodeau, Andreas Abel, Francisco Ferreira, and Rebecca Zucchini. A type theory for defining logics and proofs. In *Logic in Computer Science (LICS'19)*, 2019. Accepted for publication.
- [2] Andrea Vezzosi, Anders Mortberg, and Andreas Abel. Cubical Agda: A dependently typed programming language with univalence and higher inductive types. In *ICFP*, 2019.
- [3] Andreas Abel. Resourceful dependent types. In José Espírito Santo and Luís Pinto, editors, *24th International Conference on Types for Proofs and Programs, TYPES 2018, Braga, Portugal, June 18-21, 2018, Book of Abstracts*. EasyChair, 2018.
- [4] Andreas Abel, Joakim Öhman, and Andrea Vezzosi. Decidability of conversion for type theory in type theory. *Proceedings of the ACM on Programming Languages*, 2(POPL):23:1–23:29, 2018.
- [5] Jesper Cockx and Andreas Abel. Eliminating dependent (co)pattern matching. *Proceedings of the ACM on Programming Languages*, 3(ICFP), 2018.
- [6] Andreas Abel, Stephan Adelsberger, and Anton Setzer. Interactive programming in Agda – objects and graphical user interfaces. *Journal of Functional Programming*, 27:e8, 2017.
- [7] Andreas Abel, Andrea Vezzosi, and Théo Winterhalter. Normalization by evaluation for sized dependent types. *Proceedings of the ACM on Programming Languages*, 1(ICFP):33:1–33:30, 2017.
- [8] Andreas Abel, Andrea Vezzosi, and Theo Winterhalter. Normalization by evaluation for sized dependent types. In Ambrus Kaposi and Tamás Kozsik, editors, *23rd International Conference on Types for Proofs and Programs, TYPES 2017, Budapest, Hungary, May 29 – June 1, 2017, Book of Abstracts*. EasyChair, 2017.
- [9] Andreas Abel. Compositional coinduction with sized types. In Ichiro Hasuo, editor, *Coalgebraic Methods in Computer Science, 13th IFIP WG 1.3 International Workshop, CMCS 2016, Colocated with ETAPS 2016, Eindhoven, The Netherlands, April 2-3, 2016, Revised Selected Papers*, volume 9608 of *Lecture Notes in Computer Science*, pages 5–10. Springer, 2016.
- [10] Andreas Abel and James Chapman. Normalization by evaluation in the delay monad. In Silvia Ghilezan and Jelena Ivetic, editors, *22nd International Conference on Types for Proofs and Programs, TYPES 2016, Novi Sad, Serbia, May 23-26, 2016, Book of Abstracts*. EasyChair, 2016.
- [11] Andreas Abel, Thierry Coquand, and Bassel Mannaa. On decidability of conversion in type theory. In Silvia Ghilezan and Jelena Ivetic, editors, *22nd International Conference on Types for Proofs and Programs, TYPES 2016, Novi Sad, Serbia, May 23-26, 2016, Book of Abstracts*. EasyChair, 2016.
- [12] Andreas Abel and Brigitte Pientka. Well-founded recursion with copatterns and sized types. *Journal of Functional Programming*, 26:61, 2016. ICFP 2013 special issue.

- [13] Andreas Abel and Theo Winterhalter. An extension of Martin-Löf Type Theory with sized types. In Silvia Ghilezan and Jelena Ivetic, editors, *22nd International Conference on Types for Proofs and Programs, TYPES 2016, Novi Sad, Serbia, May 23-26, 2016, Book of Abstracts*. EasyChair, 2016.
- [14] Jesper Cockx and Andreas Abel. Sprinkles of extensionality for your vanilla type theory. In Silvia Ghilezan and Jelena Ivetic, editors, *22nd International Conference on Types for Proofs and Programs, TYPES 2016, Novi Sad, Serbia, May 23-26, 2016, Book of Abstracts*. EasyChair, 2016.
- [15] Andreas Abel. The next 700 modal type assignment systems. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs, TYPES 2015, Tallinn, Estonia, May 18-21, 2015, Book of Abstracts*. Institute of Cybernetics at Tallinn University of Technology, 2015.
- [16] Brigitte Pientka and Andreas Abel. Wellfounded recursion over contextual objects. In Thorsten Altenkirch, editor, *Typed Lambda Calculi and Applications - 13th International Conference, TLCA 2015, Warsaw, Poland, July 1-3, 2015, Proceedings*, 2015.
- [17] Andrea Vezzosi and Andreas Abel. A categorical perspective on pattern unification. In Temur Kutsia and Christophe Ringeissen, editors, *The 28th International Workshop on Unification (UNIF 2014), FLoC workshop, hosted by RTA-TLCA and IJCAR, Vienna Summer of Logic (VSL 2014)*, July 2014.
- [18] Andreas Abel. Programming and reasoning with infinite structures using copatterns and sized types. In Klaus Schmid, Wolfgang Böhm, Robert Heinrich, Andrea Herrmann, Anne Hoffmann, Dieter Landes, Marco Konersmann, Thomas Ruhroth, Oliver Sander, Volker Stolz, Baltasar Trancón y Widemann, and Rüdiger Weißbach, editors, *Gemeinsamer Tagungsband der Workshops der Tagung Software Engineering 2014, 25.-26. Februar 2014 in Kiel, Deutschland.*, volume 1129 of *CEUR Workshop Proceedings*, pages 148–150. CEUR-WS.org, 2014.
- [19] Andreas Abel and James Chapman. Normalization by evaluation in the delay monad: A case study for coinduction via copatterns and sized types. In Paul Levy and Neel Krishnaswami, editors, *Proceedings 5th Workshop on Mathematically Structured Functional Programming, MSFP 2014, Grenoble, France, 12 April 2014*, volume 153 of *Electronic Proceedings in Theoretical Computer Science*, pages 51–67, 2014.
- [20] Andreas Abel and James Chapman. Normalization by evaluation in the delay monad: A case study for coinduction via copatterns and sized types. In *Nordic Workshop on Programming Theory, NWPT 2014, Halmstad, Sweden, 29-31 October 2014*, 2014.
- [21] Andreas Abel and Andrea Vezzosi. A formalized proof of strong normalization for guarded recursive types. In Jacques Garrigue, editor, *Programming Languages and Systems - 12th Asian Symposium, APLAS 2014, Singapore, November 17-19, 2014, Proceedings*, volume 8858 of *Lecture Notes in Computer Science*, pages 140–158. Springer, 2014.
- [22] Anton Setzer, Andreas Abel, Brigitte Pientka, and David Thibodeau. Unnesting of copatterns. In Gilles Dowek, editor, *Rewriting and Typed Lambda Calculi - Joint International Conference, RTA-TLCA 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8560 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2014.

- [23] Andreas Abel. *Normalization by Evaluation: Dependent Types and Impredicativity*. Habilitation thesis, Ludwig-Maximilians-University Munich, 2013. Habilitation thesis, Faculty for Mathematics, Informatics, and Statistics, Ludwig-Maximilians-University Munich.
- [24] Andreas Abel. Productive infinite objects via copatterns. In *Nordic Workshop on Programming Theory, NWPT 2013, Tallinn, Estonia, 20-22 November 2013*, 2013.
- [25] Andreas Abel and Brigitte Pientka. Wellfounded recursion with copatterns: A unified approach to termination and productivity. In Greg Morrisett and Tarmo Uustalu, editors, *Proceedings of the Eighteenth ACM SIGPLAN International Conference on Functional Programming, ICFP'13, Boston, MA, USA, September 25-27, 2013*, pages 185–196. ACM Press, 2013.
- [26] Andreas Abel, Brigitte Pientka, David Thibodeau, and Anton Setzer. Copatterns: Programming infinite structures by observations. In Roberto Giacobazzi and Radhia Cousot, editors, *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL'13, Rome, Italy, January 23 - 25, 2013*, pages 27–38. ACM Press, 2013.
- [27] Andreas Abel. Type-based termination, inflationary fixed-points, and mixed inductive-coinductive types. In Dale Miller and Zoltán Ésik, editors, *Proceedings of the 8th Workshop on Fixed Points in Computer Science (FICS 2012)*, volume 77 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–11, 2012. Invited talk.
- [28] Andreas Abel and Gabriel Scherer. On irrelevance and algorithmic equality in predicative type theory. *Logical Methods in Computer Science*, 8(1:29):1–36, 2012. TYPES'10 special issue.
- [29] Andreas Abel, Thierry Coquand, and Miguel Pagano. A modular type-checking algorithm for type theory with singleton types and proof irrelevance. *Logical Methods in Computer Science*, 7(2:4):1–57, May 2011.
- [30] Andreas Abel. Irrelevance in type theory with a heterogeneous equality judgement. In Martin Hofmann, editor, *Foundations of Software Science and Computational Structures, 14th International Conference, FoSSaCS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26 - April 3, 2011. Proceedings*, volume 6604 of *Lecture Notes in Computer Science*, pages 57–71. Springer, 2011.
- [31] Andreas Abel and Thorsten Altenkirch. A partial type checking algorithm for Type:Type. *Electronic Notes in Theoretical Computer Science*, 229(5):3–17, 2011. Proceedings of the Second Workshop on Mathematically Structured Functional Programming (MSFP 2008).
- [32] Andreas Abel and Nicolai Kraus. A lambda term representation inspired by linear ordered logic. In Herman Geuvers and Gopalan Nadathur, editors, *6th International Workshop on Logical Frameworks and Meta-languages: Theory and Practice (LFMTP 2011), Nijmegen, The Netherlands, August 26, 2011*, volume 71 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–13, 2011.
- [33] Andreas Abel and Brigitte Pientka. Higher-order dynamic pattern unification for dependent types and records. In C.-H. Luke Ong, editor, *Typed Lambda Calculi*

and Applications - 10th International Conference, TLCA 2011, Novi Sad, Serbia, June 1-3, 2011, Proceedings, volume 6690 of *Lecture Notes in Computer Science*, pages 10–26. Springer, 2011.

- [34] Andreas Abel. MiniAgda: Integrating sized and dependent types. In Ana Bove, Ekaterina Komendantskaya, and Milad Niqui, editors, *Workshop on Partiality And Recursion in Interactive Theorem Provers, PAR 2010, Satellite Workshop of ITP'10 at FLoC 2010*, volume 43 of *Electronic Proceedings in Theoretical Computer Science*, pages 14–28, 2010.
- [35] Andreas Abel. Towards normalization by evaluation for the $\beta\eta$ -Calculus of Constructions. In Matthias Blume, Naoki Kobayashi, and Germán Vidal, editors, *Functional and Logic Programming, 10th International Symposium, FLOPS 2010, Sendai, Japan, April 19-21, 2010. Proceedings*, volume 6009 of *Lecture Notes in Computer Science*, pages 224–239. Springer, 2010.
- [36] Andreas Abel and Brigitte Pientka. Explicit substitutions for contextual type theory. In Karl Crary and Marino Miculan, editors, *5th International Workshop on Logical Frameworks and Meta-languages: Theory and Practice (LFMTP 2010), Edinburgh, Scotland, UK, July 14, 2010*, volume 34 of *Electronic Proceedings in Theoretical Computer Science*, pages 5–20, 2010.
- [37] Andreas Abel. Extensional normalization in the logical framework with proof irrelevant equality. In Olivier Danvy, editor, *Workshop on Normalization by Evaluation, affiliated to LiCS 2009, Los Angeles, 15 August 2009, 2009*.
- [38] Andreas Abel. Implementing a normalizer using sized heterogeneous types. *Journal of Functional Programming*, 19(3–4):287–310, 2009.
- [39] Andreas Abel. Type-based termination of generic programs. *Science of Computer Programming*, 74(8):550–567, 2009. MPC'06 special issue.
- [40] Andreas Abel. Typed applicative structures and normalization by evaluation for System F ω . In Erich Grädel and Reinhard Kahle, editors, *Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the EACSL, Coimbra, Portugal, September 7-11, 2009. Proceedings*, volume 5771 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2009.
- [41] Andreas Abel, Thierry Coquand, and Miguel Pagano. A modular type-checking algorithm for type theory with singleton types and proof irrelevance. In Pierre-Louis Curien, editor, *Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009, Brasilia, Brazil, July 1-3, 2009, Proceedings*, volume 5608 of *Lecture Notes in Computer Science*, pages 5–19. Springer, 2009.
- [42] Andreas Abel. Polarized subtyping for sized types. *Mathematical Structures in Computer Science*, 18(5):797–822, 2008. Special issue on subtyping, edited by Healfdene Goguen and Adriana Compagnoni.
- [43] Andreas Abel. Semi-continuous sized types and termination. *Logical Methods in Computer Science*, 4(2:3):1–33, 2008. CSL'06 special issue.
- [44] Andreas Abel. Weak $\beta\eta$ -normalization and normalization by evaluation for System F. In Illiano Cervesato, Helmut Veith, and Andrei Voronkov, editors, *15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2008, 22-27 November 2008, Doha, Qatar, Proceedings*, volume 5330 of *Lecture Notes in Artificial Intelligence*, pages 497–511. Springer, 2008.

- [45] Andreas Abel, Thierry Coquand, and Peter Dybjer. On the algebraic foundation of proof assistants for intuitionistic type theory. In Jacques Garrigue and Manuel V. Hermenegildo, editors, *Functional and Logic Programming, 9th International Symposium, FLOPS 2008, Ise, Japan, April 14-16, 2008. Proceedings*, volume 4989 of *Lecture Notes in Computer Science*, pages 3–13. Springer, 2008.
- [46] Andreas Abel, Thierry Coquand, and Peter Dybjer. Verifying a semantic $\beta\eta$ -conversion test for Martin-Löf type theory. In Philippe Audebaud and Christine Paulin-Mohring, editors, *Mathematics of Program Construction, 9th International Conference, MPC 2008, Marseille, France, July 15-18, 2008. Proceedings*, volume 5133 of *Lecture Notes in Computer Science*, pages 29–56. Springer, 2008.
- [47] Andreas Abel and Dulma Rodriguez. Syntactic metatheory of higher-order subtyping. In Michael Kaminski and Simone Martini, editors, *Computer Science Logic, 22nd International Workshop, CSL 2008, 17th Annual Conference of the EACSL, Bertinoro, Italy, September 16-19, 2008. Proceedings*, volume 5213 of *Lecture Notes in Computer Science*, pages 446–460. Springer, 2008.
- [48] Andreas Abel. Mixed inductive/coinductive types and strong normalization. In Zhong Shao, editor, *Programming Languages and Systems, 5th Asian Symposium, APLAS 2007, Singapore, November 29-December 1, 2007. Proceedings*, volume 4807 of *Lecture Notes in Computer Science*, pages 286–301. Springer, 2007.
- [49] Andreas Abel. Strong normalization and equi-(co)inductive types. In Simona Ronchi Della Rocca, editor, *Typed Lambda Calculi and Applications, 8th International Conference, TLCA 2007, Paris, France, June 26-28, 2007. Proceedings*, volume 4583 of *Lecture Notes in Computer Science*, pages 8–22. Springer, 2007.
- [50] Andreas Abel. Syntactical normalization for intersection types with term rewriting rules. In *Fourth International Workshop on Higher-Order Rewriting, HOR'07, Paris, France, 25 June 2007, 2007*.
- [51] Andreas Abel, Klaus Aehlig, and Peter Dybjer. Normalization by evaluation for Martin-Löf type theory with one universe. In Marcelo Fiore, editor, *Proceedings of the 23rd Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIII), New Orleans, LA, USA, 11-14 April 2007*, volume 173 of *Electronic Notes in Theoretical Computer Science*, pages 17–39. Elsevier, 2007.
- [52] Andreas Abel and Thierry Coquand. Untyped algorithmic equality for Martin-Löf's logical framework with surjective pairs. *Fundamenta Informaticae*, 77(4):345–395, 2007. TLCA'05 special issue.
- [53] Andreas Abel, Thierry Coquand, and Peter Dybjer. Normalization by evaluation for Martin-Löf Type Theory with typed equality judgements. In *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings*, pages 3–12. IEEE Computer Society Press, 2007.
- [54] Andreas Abel. Implementing a normalizer using sized heterogeneous types. In Conor McBride and Tarmo Uustalu, editors, *Workshop on Mathematically Structured Functional Programming, MSFP 2006, Kuressaare, Estonia, July 2, 2006*, electronic Workshop in Computing (eWiC). The British Computer Society (BCS), 2006.

- [55] Andreas Abel. Polarized subtyping for sized types. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *Computer Science - Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8-12, 2006, Proceedings*, volume 3967 of *Lecture Notes in Computer Science*, pages 381–392. Springer, 2006.
- [56] Andreas Abel. *A Polymorphic Lambda-Calculus with Sized Higher-Order Types*. PhD thesis, Ludwig-Maximilians-Universität München, 2006.
- [57] Andreas Abel. Semi-continuous sized types and termination. In Zoltán Ésik, editor, *Computer Science Logic, 20th International Workshop, CSL 2006, 15th Annual Conference of the EACSL, Szeged, Hungary, September 21-24, 2006, Proceedings*, volume 4207 of *Lecture Notes in Computer Science*, pages 72–88. Springer, 2006.
- [58] Andreas Abel. Towards generic programming with sized types. In Tarmo Uustalu, editor, *Mathematics of Program Construction: 8th International Conference, MPC 2006, Kuressaare, Estonia, July 3-5, 2006. Proceedings*, volume 4014 of *Lecture Notes in Computer Science*, pages 10–28. Springer, 2006.
- [59] Andreas Abel, Thierry Coquand, and Ulf Norell. Connecting a logical framework to a first-order logic prover. In Bernhard Gramlich, editor, *5th International Workshop on Frontiers of Combining Systems, FroCoS'05, Vienna, Austria, September 19-21, 2005*, volume 3717 of *Lecture Notes in Artificial Intelligence*, pages 285–301. Springer, September 2005.
- [60] Andreas Abel and Thierry Coquand. Untyped algorithmic equality for Martin-Löf’s logical framework with surjective pairs. In Paweł Urzyczyn, editor, *Typed Lambda Calculi and Applications (TLCA 2005), Nara, Japan*, volume 3461 of *Lecture Notes in Computer Science*, pages 23–38. Springer, April 2005.
- [61] Andreas Abel, Marcin Benke, Ana Bove, John Hughes, and Ulf Norell. Verifying Haskell programs using constructive type theory. In *ACM SIGPLAN 2005 Haskell Workshop, Haskell’05, Tallinn, Estonia, September 30, 2005*, pages 62–73. ACM Press, 2005.
- [62] Andreas Abel, Ralph Matthes, and Tarmo Uustalu. Iteration schemes for higher-order and nested datatypes. *Theoretical Computer Science*, 333(1–2):3–66, 2005.
- [63] Andreas Abel. Termination checking with types. *RAIRO – Theoretical Informatics and Applications*, 38(4):277–319, 2004. Special Issue: Fixed Points in Computer Science (FICS’03).
- [64] Andreas Abel. Weak normalization for the simply-typed lambda-calculus in Twelf. In *Logical Frameworks and Metalanguages (LFM 2004), IJCAR Satellite Workshop, Cork, Ireland, 2004*.
- [65] Andreas Abel and Ralph Matthes. Fixed points of type constructors and primitive recursion. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *Computer Science Logic, 18th International Workshop, CSL 2004, 13th Annual Conference of the EACSL, Karpacz, Poland, September 20-24, 2004, Proceedings*, volume 3210 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2004.
- [66] Andreas Abel, Ralph Matthes, and Tarmo Uustalu. Generalized iteration and coiteration for higher-order nested datatypes. In A. Gordon, editor, *Foundations*

- of *Software Science and Computation Structures (FoSSaCS 2003)*, volume 2620 of *Lecture Notes in Computer Science*, pages 54–69, Warsaw, Poland, April 2003. Springer.
- [67] Andreas Abel. Termination and guardedness checking with continuous types. In Martin Hofmann, editor, *Typed Lambda Calculi and Applications, 6th International Conference, TLCA 2003, Valencia, Spain, June 10-12, 2003, Proceedings*, volume 2701 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2003.
 - [68] Andreas Abel and Ralph Matthes. (Co-)iteration for higher-order nested datatypes. In H. Geuvers and F. Wiedijk, editors, *Types for Proofs and Programs, International Workshop, TYPES 2002*, volume 2646 of *Lecture Notes in Computer Science*, pages 1–20, Berg en Dal, The Netherlands, 2003. Springer.
 - [69] Andreas Abel and Ralph Matthes. Primitive recursion for rank-2 inductive types. In *Fixed Points in Computer Science, FICS'03, Satellite Workshop to ETAPS 03, Warsaw, Poland, 2003*.
 - [70] Andreas Abel and Thorsten Altenkirch. A predicative analysis of structural recursion. *Journal of Functional Programming*, 12(1):1–41, January 2002.
 - [71] Andreas Abel. A third-order representation of the $\lambda\mu$ -calculus. In R.L. Crole S.J. Ambler and A. Momigliano, editors, *Electronic Notes in Theoretical Computer Science*, volume 58. Elsevier, 2001.
 - [72] Andreas Abel, Bor-Yuh Evan Chang, and Frank Pfenning. Human-readable machine-verifiable proofs for teaching constructive logic. Technical report, Università degli Studi Siena, Dipartimento di Ingegneria dell'Informazione, 2001. In *Proceedings of the Workshop on Proof Transformation and Presentation and Proof Complexities (PTP'01)*.
 - [73] Andreas Abel. Specification and verification of a formal system for structurally recursive functions. In Thierry Coquand, Peter Dybjer, Bengt Nordström, and Jan Smith, editors, *Types for Proofs and Programs, International Workshop TYPES'99, Lökeberg, Sweden, June 12-16, 1999, Selected Papers*, volume 1956 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2000.
 - [74] Andreas Abel and Thorsten Altenkirch. A predicative strong normalisation proof for a λ -calculus with interleaving inductive types. In T. Coquand, P. Dybjer, B. Nordström, and J. Smith, editors, *Types for Proof and Programs, International Workshop, TYPES '99*, volume 1956 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2000.
 - [75] Andreas Abel and Thorsten Altenkirch. A semantical analysis of structural recursion. In *4th International Workshop on Termination (WST'99), Dagstuhl, Germany, May 1999*.
 - [76] Andreas Abel. A semantic analysis of structural recursion. Master's thesis, Ludwig-Maximilians-University Munich, 1999.

Courses in Summer Schools, Upon Invitation, Tutorials

- 2019 46th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2019), tutorial *Correct-By-Construction Programming in Agda*, Cascais, Portugal, January 2019.
- 2018 *Introduction to Dependent Types and Agda*, Eighth Summer School on Formal Techniques, Menlo College, Atherton, CA, USA, May 2018.
- 2017 *Type Theory*, EAFIT University Medellín, Columbia, March 2017.
- 2016 *Agda*, Autumn school Proof and Computation, Fischbachau, Germany, October 2016.
- 2016 *Type Theory, A Constructive Foundation for Logics and Computer Science*, European Summer School on Logic, Language, and Information (ESSLLI 2016).

Teaching at Gothenburg University

At the Department of Computer Science and Engineering of the Chalmers University of Technology and the Gothenburg University, I have been teaching the following classes. Terms last 8 weeks, with 4 academic hours of lecture each week.

- *Programming Language Technology* (DAT151/DIT231, 7.5 ETCS). Master level course on parsing, type-checking, interpretation, and compilation. Given in 2015, 2016, 2017, 2018.
- *Logic, Algorithms, and Datastructures* (DIT725, 7.5 ETCS). Introductory course on the Bachelor level in program *Software Engineering and Management*. Given in 2014.

Since December 2017 I am running the PhD course *Types and Programming Languages* aka *Initial Types Club* targeted also at Master students. The purpose is to cover topics beyond the Master curriculum to bridge the gap to research, and to encourage informal interaction between research candidates (Master students) on the one hand, and researchers (PhD students, postdocs and lecturers) on the other hand.

Teaching at the University of Munich

At the Department of Computer Science of the Ludwig-Maximilians-University Munich, I have given the following courses entirely by myself. Terms last 14 weeks, annotations like (4+2) shall mean 4 hours of lecture and 2 hours of tutorial section per week.

- *Computer-Aided Formal Reasoning* (3+2), Winter 2010/11.
- *Lambda-Calculus* (3+1), held in the Winter terms of 2005/06 and 2007/08.
- *Theory and Implementation of Object-Oriented Programming Languages* (2+2), Summer 2002.
- *Python*, one week compact course, Summer 2008.

These courses I have shared with colleagues:

- *Advanced Functional Programming* (3+2), with Dr. Steffen Jost, Summer 2012.
- *Functional Programming in SML* (3+2), with Prof. Martin Hofmann in Summer 2011.
- *Type Systems* (3+2), with Dr. Ulrich Schöpp, Summer 2011.
- *Semantics of Programming Languages* (4+2), with Dr. Ulrich Schöpp, Summer 2010.
- *Advanced Functional Programming* (2+2), with Dr. Hans-Wolfgang Loidl, Summer 2009.
- *Type Systems* (4+2), with Prof. Martin Hofmann, Summer 2007.
- *Compiler Construction Lab* (2+2), with Dr. Ulrich Schöpp, Winter 2011/12 and Summer 2013, and with Dr. Hans-Wolfgang Loidl, Winter 2007/08.
- *Programming Language Theory Seminar* (2), with Dr. Lennart Beringer and Dr. Hans-Wolfgang Loidl, Summer 2007.

Assistance in Teaching

For other courses I have organized the tutorial sections, designed assignments and exams, and held tutorial sections, including the following first and second year courses:

- *Logics and Discrete Structures* (3+2), held by Prof. Martin Hofmann in Summer 2011.
- *Formal Languages and Complexity* (3+2), held by Prof. Martin Hofmann in Summer 2011.
- *Introduction to Programming with JAVA* (3+2), held by Prof. Martin Hofmann in Winter 2011/12.
- *Datastructures and Algorithms* (3+2), held by Prof. Martin Hofmann in Summer 2009.
- *Programming and Modelling with SML* (3+2), held by Prof. François Bry in Summer 2008.
- *Efficient Algorithms* (4+2), held by Prof. Martin Hofmann in Summer 2006 and 2002.
- *Constructive Logic*, held by Prof. Frank Pfenning at Carnegie Mellon University in Fall 2000.

Supervision of Students in Seminars

For the following seminars (Diploma, Bachelor, and Master level) I supervised students for their talk preparation.

2013 Type-based Program Analysis.

2012 Coding and Information.

2011 Efficient Functional Data Structures.
2011 Scientific Work and Teaching.
2010 Distributed Algorithms.
2009 Program Analysis.
2007 Programming Language Theory.

Supervised PhD Students

Currently I am supervising the following PhD students in the Department of Computer Science and Engineering, Gothenburg University.

- Andrea Vezzosi (2014–2019), *On Induction, Coinduction and Equality in Martin-Löf and Homotopy Type Theory*.
Examination committee: Rasmus Ejlers Møgelberg (ITU Copenhagen, opponent), Robert Atkey (Strathclyde University, Glasgow), Hugo Herbelin (INRIA Paris), Andrew Pitts (Cambridge).
Joint publications: [17] [21] [7] [4] [2].
- Victor Lopez Juan (2015–) (co-supervisor) on *Implementing a Dependent Type-Checker with Implicit Arguments*.

In Munich, I have been co-supervising PhD student Christoph-Simon Senjak on *An Implementation of Deflate in Coq*. (graduated 2018) from 2012 to 2013 in the DFG PhD Program PUMA (program and model analysis), and Dulma Rodriguez on *Amortised Resource Analysis for Object-Oriented Programs* who graduated in 2012.

Supervised Diploma and Master Students

Ongoing supervisions (2019):

1. Elias Forsberg, *A Formally Verified Parser Generator in Agda*
2. Alexander Fuhs, *Verified Compilation of an Intrinsically-Typed while Language in Agda*
3. Siavash Hamedani, *Agda's Scope Checker implemented in Agda*
4. Jannis Limperg, *Modeling Agda's Sized Types in Agda*
5. Frederik Folkmar Ramcke, *LF Definability in Agda*

Completed supervisions:

- 2018 Pierre Krafft, *Formalizing a Lambda Calculus for Typed Actors*
- 2016 Joakim Öhman, *A Logical Relation for Dependent Type Theory Formalized in Agda*. Master's thesis, published [4].
- 2016 Glen Mevel, *System U with η -equality*. ENS Cachan research internship.
- 2015 Theo Winterhalter, *Dependent Type Theory with Sized Types*. ENS Cachan research internship. Publication [7].
- 2015 Marcus Eskil Johansson and Jesper Llody, *Eliminating the problems of hidden-lambda insertion*. Master's thesis.

- 2012 Matthias Benkard, *Type Checking without Types*. Investigation of a functional language with pattern matching and a refinement relation between terms that plays the role of typing and subtyping. Won a ICFP 2012 Student research competition travel award and a short presentation at the main conference.
- 2011/12 Frederic Kettelhoit, *A Prelude for Agda*. Master's thesis.
- 2011 Gabriel Scherer, *Universe Subtyping in Martin-Löf Type Theory*. Research internship on the Master's level. Publication [28].
- 2007 Karl Mehlretter, *Termination Checking for a Dependently Typed Language*. Diploma thesis.
- 2007 Dulma Rodriguez, *Algorithmic Subtyping for Higher-Order Bounded Quantification Revisited*. Diploma thesis, published [47].

Supervised Bachelor Students and Student Projects

- 2018 Johannes Backlund, Ludvig Ekman, Magnus Harryson, Johan Ronnås, Robin Åstedt and Jonatan Öberg, *A Smartphone Game Modeled after Logic and Proofs*
- 2015 Gregor Ulm, *Compiling Agda to System F_ω* .
- 2013 Felix Reihl, *Solving Size Constraints Using Graph Representation*. Bachelor's thesis.
- 2012 David Thibodeau, *A Core Calculus for Covering Copatterns*. Undergraduate research internship. Publication [26].
- 2010/11 Nicolai Kraus, *A Term Representation Based on Ordered Logic*. Bachelor's thesis, published [32].
- 2009/10 Julien Oster, *Red-Black Trees in Agda*. Advanced programming lab.
- 2006 Dulma Rodriguez, *Verification of Iteration and Coiteration Schemes for Higher-Order and Nested Datatypes in Coq*, Fortgeschrittenenpraktikum (Advanced Programming Lab) at Ludwig-Maximilians-University, München.
- 2003 Jan Peter Gutzmann, *Implementation eines Typprüfers für System F_ω* , Fortgeschrittenenpraktikum (Advanced Programming Lab) at Ludwig-Maximilians-University, München.
- 2001 Bor-Yuh Evan Chang, *Human-Readable Machine-Verifiable Proofs for Teaching Constructive Logic*, Junior Research Project at Carnegie Mellon University, Pittsburgh. Publication [72].

Supervised Industrial Master Theses

- 2016 Alaa Alnuweiri and Jemima Masamu, *Generic Log and Performance Data from Customer Installations*
- 2016 Philip Dahlstedt and Gustav Öhman, *Dynamic and fault tolerant distributed computations using the actor model*
- 2016 Anton Lindgren, *Low-latency floating subsets through combined indexes*

Supervised Student-Proposed Bachelor Theses

- 2014 Henrik Alburg, Filip Brynfors, Björn Persson Mattsson, Florian Minges, Jakob Svensson, *Making and Acting on Predictions in StarCraft Brood War.*
- 2014 Julia Gustafsson, Antonious Kioksoglou, Anton Kloek, Victor Lindhé, Barnabas Sapan, *Safe and Social: Location-based Social Networking Focusing on Security.*
- 2014 Viktor Anderling, Olle Andreasson, Christoffer Olsson, Sean Pavlov, Christian Svensson, Johannes Wikner, *Generation of music through genetic algorithms.*
- 2014 Jack Petterson, Leif Schelin, Niklas Wärvik, Joakim Öhman, *A general peer-to-peer based distributed computation network.*

References

For letters of recommendation, please contact the following researchers:

Prof. Thorsten Altenkirch, PhD
School of Computer Science
University of Nottingham, Jubilee Campus
Wollaton Road
Nottingham, NG8 1BB, UK
phone: +44 115 84 66516
email: txa@cs.nott.ac.uk

Prof. Dr. Thierry Coquand
Dept. of Comput. Sci. and Eng.
Gothenburg University
Rännvägen 6b
41296 Göteborg, Sweden
phone: +46 31 772 1030
email: coquand@chalmers.se

Prof. Peter Dybjer, PhD
Dept. of Comput. Sci. and Eng.
Chalmers University of Technology
Rännvägen 6b
41296 Göteborg, Sweden
phone: +46 31 772 1035
email: peterd@chalmers.se

Hugo Herbelin, Directeur de Recherche
Laboratoire PPS,
Université Paris Diderot - Paris 7,
Case 7014,
75205 Paris Cedex 13, France
phone: +33 1 57 27 90 87
email: Hugo.Herbelin@inria.fr

Dr. Ralph Matthes
IRIT - Université Paul Sabatier
Équipe ACADIE
118 route de Narbonne
F-31062 Toulouse Cedex 9, France
phone: +33 56155 4342
email: matthes@irit.fr

Prof. Frank Pfenning, PhD
Department of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891, U.S.A.
phone: +1 412 268-6343
email: fp@cs.cmu.edu

Prof. Brigitte Pientka, PhD
McConnell Eng. Bldg., Room 107N
School of Computer Science, McGill University
3480 University Street
Montreal, Quebec, H3A 0E9, Canada
phone: +1 514 398 2583
email: bpientka@cs.mcgill.ca

Prof. Andrew Pitts, PhD
University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
phone: +44 1223 334629
email: Andrew.Pitts@cl.cam.ac.uk

Prof. Dr. Schwichtenberg
Institut für Mathematik
Ludwig-Maximilians-Universität München
Theresienstr. 39
80333 München, Germany
phone: +49 89 2180 4413
email: schwicht@math.lmu.de

Address

office: Department of Computer Science and Engineering
Gothenburg University
Rännvägen 6b
41296 Göteborg
SWEDEN

phone: +46-31-772-1731

email: andreas.abel@gu.se

web: <http://www.cse.chalmers.se/~abela>

home: Kolonigatan 7
41321 Göteborg
SWEDEN

phone: +46-729-744-783