Curriculum Vitae

Full name:	Andreas Martin Abel
Nationality:	German
Date of birth:	June 24, 1974
Place of birth:	Munich, Germany

Academic and Visiting Positions

Oct 2013 – Today	Senior Lecturer Department of Computer Science and Engineering Chalmers University of Technology and Gothenburg University, Gothenburg, Sweden
Oct 2009 – March 2010	Invited Researcher PI.R2 project of INRIA Rocquencourt and PPS (CNRS Lab Programs, Proofs, and Systems), Paris, France
Oct 2008 – Sep 2013	*Akademischer Rat auf Zeit (A13)* Assistant Professor with the status of a *Beamter*
Oct 2005 – Sep 2008	*Wissenschaftlicher Mitarbeiter (A13)* Assistant Professor at Prof. Hofmann's chair of Theoretical Computer Science, 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 at Prof. Hofmann's chair of Theoretical Computer Science, Ludwig-Maximilians-University Munich (substitute for Dr. Matthes)
May 2000 – June 2001	Visiting Teaching and Research Assistant for Prof. Pfenning, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, USA

Degrees

2013 *Privatdozent* (Dr. habil.), Faculty of Mathematics, Computer Science, and Statistics, University of Munich

2006 *Doctor rerum naturarum* (Ph.D.), Department of Computer Science, University of Munich
Mark: *magna cum laude*

1999 *Informatik Diplom* (M.Sc.), University of Munich
Mark: *sehr gut* (best mark)

Awards, Fellowships, and Grants

2015–2018	Research project *Termination Certificates for Dependently-Typed Programs and Proofs via Refinement Types* 3 600 000 SEK (400 000 EUR) granted by Vetenskapsrådet, Stockholm I am sole applicant and principal investigator.
2012	Invited Presentation at FICS Workshop, Tallinn, Estonia
2009	Invited Researcher (6 months) of INRIA, France
2008	EUR 6350 Travel funds for research cooperation with LORIA, Nancy, on *Type-Based Termination*, provided by the *Bayerisch-französisches Hochschulzentrum*
2005–2007	Research project *Typed Lambda Calculi and Applications* 600 000 SEK granted by Vetenskapsrådet, Stockholm Co-application with Prof. Thierry Coquand and Prof. Peter Dybjer
2004–2005	Postdoctoral Fellowship from the Swedish Foundation of Strategic Research (SFF) in the CoVer Project, Chalmers
2000–2001	Fellowship from the Office of Technology in Education, Carnegie-Mellon University
1999–2003	Stipend from PhD Program Logic in Computer Science, *Deutsche Forschungsgemeinschaft (DFG)* (German Research Foundation)

Industrial Involvement and Programming

2011–2012	External consultant for *Glueware Informatik GmbH*, Munich (framework for business application components)	Scala
2007–today	Developing the research prototype *MiniAgda*	Haskell
2004–today	Open-source developer in the *Agda* team	Haskell
2000–2001	Development of *Tutch*	SML
1990–1998	Software development for the publishing house *Park Körner*, Munich	Delphi

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

Professional Service

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 LFMTP 2008, Pittsburgh, colocated with LICS 2008, and PLPV 2013, Rome, affiliated to POPL 2013.

Research administration. Administrator of the Munich-LMU subsite (headed by Helmut Schwichtenberg) of the EU Coordination Action (FP6-2002-IST-C) TYPES, Types for Proofs and Programs (510996), from Dec 2005 to Jun 2009.

Steering. Member of the steering committee of the international workshop series LFMTP, Nov 2009 – Oct 2015. Chair Jul 2010 – Oct 2015.

Doctoral committee membership. Member of the doctoral committees of Johan Granström, *Reference and Computation in Intuitionistic Type Theory*, Uppsala University, Jan 2009; Cody Roux, *Type Based Termination: Semantics and Generalisations*, LORIA, Nancy, June 2011; and Jorge Sacchini, *On Type-Based Termination and Dependent Pattern Matching in the Calculus of Inductive Constructions*, École des Mines de Paris, June 2011.

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

PC membership. Member of the program committee of the international conferences FoSSaCS 2010, MFCS 2011, ESOP 2012, RTA 2012, ITP 2012, FoSSaCS 2013, TLCA 2013, Haskell Symposium 2013, RTA-TLCA 2014, POPL 2015 (extended review committee), CPP 2015, FoSSaCS 2015, ICALP 2015 (Track B), ESOP 2016, and FLOPS 2016.

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

Member of the program committees of the international workshops LFMTP 2007, PLPV 2009, PAR 2010, MSFP 2010, HOR 2012, FICS 2013, WGP 2013, CMCS 2014, MSFP 2014, TYPES 2014, DTP 2014, LFMTP 2014, ATPS 2015, PEPM 2015, LFMTP 2015.

Additional reviewing. Review of more than 110 research papers for the conferences APLAS 2007, 2010, and 2013, CADE 2001, 2003, and 2008, CiE 2008 and 2010, CSL 2001, 2002, 2008, 2009, 2010, and 2011, ESOP 2007, 2009, and 2013, FoSSaCS 2008, FLOPS 2010, FSTTCS 2000, ICALP 2004 and 2008, ICFP 2013 and 2015, LiCS 2008, 2011, 2013, and 2014, LPAR 2005, 2006, and 2010, MFCS 2014, MPC 2008, POPL 2008, 2009, 2013, and 2014, PPDP 2007 and 2009, RTA 2004, 2007, 2009, and 2011, STACS 2007, TLCA 2003 and 2015, the post-workshop proceedings CL&C 2009, TFP 2005 and 2008, TYPES 2002, 2006, 2007, 2008, 2010, 2013, and 2014, and the journals Fundamenta Informaticae, HOSC, IGPL, IPL, JAR, JFP, LMCS, MSCS, and TCS.

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). I have advanced the field in the following areas:

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 [1, 2, 3] and integrated it into Agda 2.

For my PhD, I have studied type systems for termination checking [4, 5, 6, 7] culminating in *a polymorphic lambda-calculus with sized higher-order types* [8]. I am continuing to explore the technological aspects of sized types in my implementation MiniAgda [9, 10] of dependent type theory.

Coinduction. Processes and potentially infinite streams are modelled 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 [11], justified by a semantics based on orthogonality [12] and saturated sets [13].

With Brigitte Pientka, David Thibodeau, and Anton Setzer I have developed *copatterns*, a new approach to corecursion in functional programming and Type Theory [14, 15]. Copatterns unify the concepts of productivity of infinite objects and termination of recursive functions.

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 [16, 17, 18] and its extension by retract types [19]. Even more recursion schemes are handled by type-based termination [8] 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 [20, 21].

Dependent Type Theory and Logical Frameworks. During my 22 month visit to Chalmers, 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) [22, 23]. 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 [24].

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 [25] and judgmental equality [26]. 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 [27] as it is used in Agda and Epigram. I have presented a version of NbE for System F [28] simplifying previous approaches. This has been extended to System F^{ω} [29] and the Calculus of Constructions [30]. With Miguel Pagano and Thierry Coquand I have also treated NbE for singleton types and proof irrelevance [31].

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 compiliation [32, 33].

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 [34].

Technology and Practice of Dependent Types. My expertise in type-checking engines allow me to make significant contributions to the *Agda* code base; after Ulf Norell and Nils Anders Danielsson I am the 3rd top developer. As of today (April 2014), I have contributed close to 1200 patches (out of 5500 total) and fixed over 200 issues (out of 1000 total).

Other Research Interests

Subtyping. I have found a new and short proof of completeness of algorithmic subtyping for the higher-order polymorphic lambda-calculus [35, 36]. Together with my student Dulma Rodriguez I have extended it to System F-sub with bounded quantification [37]. 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 *tut*orial proof *ch*ecker, which verifies natural deduction proofs written by students in a block-structured syntax [38]. 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 [24] 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 [39]. I also formalized a combinatorial weak normalization proof for the simply-typed lambda-calculus [40]. 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 [41].

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 [42].

Peer-Reviewed Publications

In international journals (10):

LMCS 12 [33] LMCS 11 [43] SCP 09 [21] JFP 09 [44] MSCS 08 [36] LMCS 08 [7] Fund.Inf. 07 [23] TCS 05 [18] ITA 04 [4] JFP 02 [2]

In workshop proceedings published in international journals (4):

MSFP 14 [45] MSFP 08 [41] LFM 04 [40] MERLIN 01 [39]

In proceedings of international conferences (25):

TLCA 15 [46] APLAS 14 [47] RTA-TLCA 14 [48] ICFP 13 [15] POPL 13 [14] TLCA 11 [34] FoSSaCS 11 [32] FLOPS 10 [30] CSL 09 [29] TLCA 09 [31] LPAR 08 [28] CSL 08 [37] MPC 08 [27] FLOPS 08 [49] APLAS 07 [13] LiCS 07 [26] TLCA 07 [12] MFPS 07 [25] CSL 06 [6] MPC 06 [20] CSR 06 [35] TLCA 05 [22] CSL 04 [19] TLCA 03 [5] FoSSaCS 03 [17]

In proceedings of international workshops (13):

UNIF 14 [50] LFMTP 11 [51] PAR 10 [9] LFMTP 10 [52] NBE 09 [53] HOR 07 [54] MSFP 06 [55] Haskell 05 [42] FroCoS 05 [24] TYPES 02 [16] PTP 01 [38] TYPES 99a [11] TYPES 99b [3]

Invited presentations:

FICS 2012 [10]

Bibliography

[1] Andreas Abel. A semantic analysis of structural recursion. Master's thesis, Ludwig-Maximilians-University Munich, 1999.

[2] Andreas Abel and Thorsten Altenkirch. A predicative analysis of structural recursion. *Journal of Functional Programming*, 12(1):1–41, January 2002.

[3] 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 Proof and Programs, International Workshop, TYPES '99*, volume 1956 of *Lecture Notes in Computer Science*, pages 1–20. Springer-Verlag, 2000.

[4] 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).

[5] Andreas Abel. Termination and guardedness checking with continuous types. In M. Hofmann, editor, *Typed Lambda Calculi and Applications (TLCA 2003), Valencia, Spain*, volume 2701 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, June 2003.

[6] 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-Verlag, 2006.

[7] Andreas Abel. Semi-continuous sized types and termination. *Logical Methods in Computer Science*, 4(2:3):1–33, 2008. CSL'06 special issue.

[8] Andreas Abel. *A Polymorphic Lambda-Calculus with Sized Higher-Order Types*. PhD thesis, Ludwig-Maximilians-Universität München, 2006.

[9] 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.

[10] Andreas Abel. Type-based termination, inflationary fixed-points, and mixed inductive-coinductive types. *Electronic Proceedings in Theoretical Computer Science*, 77:1–11, 2012. Proceedings of FICS 2012.

[11] 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-Verlag, 2000.

[12] 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-Verlag, 2007.

[13] 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-Verlag, 2007.

[14] 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.

[15] 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.

[16] 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-Verlag.

[17] 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-Verlag.

[18] Andreas Abel, Ralph Matthes, and Tarmo Uustalu. Iteration schemes for higher-order and nested datatypes. *Theoretical Computer Science*, 333(1–2):3–66, 2005.

[19] 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-Verlag, 2004.

[20] 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-Verlag, 2006.

[21] Andreas Abel. Type-based termination of generic programs. *Science of Computer Programming*, 74(8):550–567, 2009. MPC'06 special issue.

[22] 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-Verlag, April 2005.

[23] 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.

[24] 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-Verlag, September 2005.

[25] 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.

[26] 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.

[27] 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-Verlag, 2008.

[28] 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-Verlag, 2008.

[29] Andreas Abel. Typed applicative structures and normalization by evaluation for System Fω. 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-Verlag, 2009.

[30] 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-Verlag, 2010.

[31] Andreas Abel, Thierry Coquand, and Miguel Pagano. A modular type-checking algorithm for type theory with singleton types and proof irrelevance. In P.-L. 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-Verlag, 2009.

[32] 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-Verlag, 2011.

[33] 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.

[34] 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-Verlag, 2011.

[35] 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-Verlag, 2006.

[36] Andreas Abel. Polarized subtyping for sized types. *Mathematical Structures in Computer Science*, 18:797–822, 2008. Special issue on subtyping, edited by Healfdene Goguen and Adriana Compagnoni.

[37] 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-Verlag, 2008.

[38] 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).

[39] Andreas Abel. A third-order representation of the $\lambda\mu$-calculus. In S.J. Ambler, R.L. Crole, and A. Momigliano, editors, *Electronic Notes in Theoretical Computer Science*, volume 58. Elsevier Science Publishers, 2001.

[40] 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.

[41] 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).

[42] 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.

[43] 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.

[44] Andreas Abel. Implementing a normalizer using sized heterogeneous types. *Journal of Functional Programming*, 19(3–4):287–310, 2009.

[45] Andreas Abel and James Chapman. Normalization by evaluation in the delay monad: A case study for coinduction via copatterns and sized types. In *International Workshop on Mathematically Structured Functional Programming, colocated with ETAPS 2014, 5-13 April 2014, Grenoble, France*, April 2014.

[46] 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.

[47] 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-Verlag, 2014.

[48] Anton Setzer, Andreas Abel, Brigitte Pientka, and David Thibodeau. Unnesting of copatterns. In *Joint 25th International Conference on Rewriting Techniques and Applications and 12th International Conference on Typed Lambda Calculi and Applications (RTA-TLCA 2014), July 14-17, 2014, Vienna, Austria*, 2014. Accepted for presentation.

[49] 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-Verlag, 2008.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

Teaching

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.

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.

Supervised PhD Students

Currently I am supervising PhD student Andrea Vezzosi in the Department of Computer Science and Engineering, Gothenburg University. Joint publications: [50] [47].

In Munich, I have been co-supervising PhD student Christoph-Simon Senjak (2012-2013) in the DFG PhD Program PUMA (Program- and Modelanalysis), and Dulma Rodriguez who graduated in 2012.

Supervised Diploma and Master Students

2015 Theo Winterhalter, *Dependent Type Theory with Sized Types*. ENS Cachan research internship (supervision ongoing).

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.

2007 Karl Mehltretter, *Termination Checking for a Dependently Typed Language*. Diploma thesis.

2007 Dulma Rodriguez, *Algorithmic Subtyping for Higher-Order Bounded Quantification Revisited*. Diploma thesis, published [37].

Supervised Bachelor Students and Student Projects

2015 Gregor Ulm, *Compiling Agda to System F_ω* (supervision ongoing).

2013 Felix Reihl, *Solving Size Constraints Using Graph Representation*. Bachelor's thesis.

2012 David Thibodeau, *A Core Calculus for Covering Copatterns*. Undergraduate research internship.

2010/11 Nicolai Kraus, *A Term Representation Based on Ordered Logic*. Bachelor's thesis, published [51].

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.

References

For letters of recommendation, please contact the following researchers:

Thorsten Altenkirch, PhD, Reader
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**
Department of Computer Science
Chalmers University of Technology
Rännvägen 6
41296 Göteborg, Sweden
phone: +46 31 772 1030
email: coquand@chalmers.se

Prof. **Peter Dybjer**, PhD
Department of Computer Science
Chalmers University of Technology
Rännvägen 6
41296 Göteborg, Sweden
phone: +46 31 772 1035
email: peterd@chalmers.se

Prof. **Martin Hofmann**, PhD
Institut für Informatik
Ludwig-Maximilians-Universität München
Oettingenstr. 67
80538 München, Germany
phone: +49 89 2180 9317
email: martin.hofmann@ifi.lmu.de

Hugo Herbelin, Directeur de Recherche
INRIA
23 avenue d'Italie
CS 81321
75214 Paris Cedex 13, France
phone: +33 1 39 63 59 28
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 Pienkta**, 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

Address

office: Department of Computer Science
Chalmers University of Technology
Rännvägen 6
41296 Göteborg
SWEDEN

phone: +46-31-772-1731

email: andreas.abel@gu.se

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

home: Långedragsvägen 29
42671 Göteborg
SWEDEN

phone: +49-176-400-333-23