Reinhard Wilhelm

Patrick Cousot

cims.nyu.edu/~pcousot

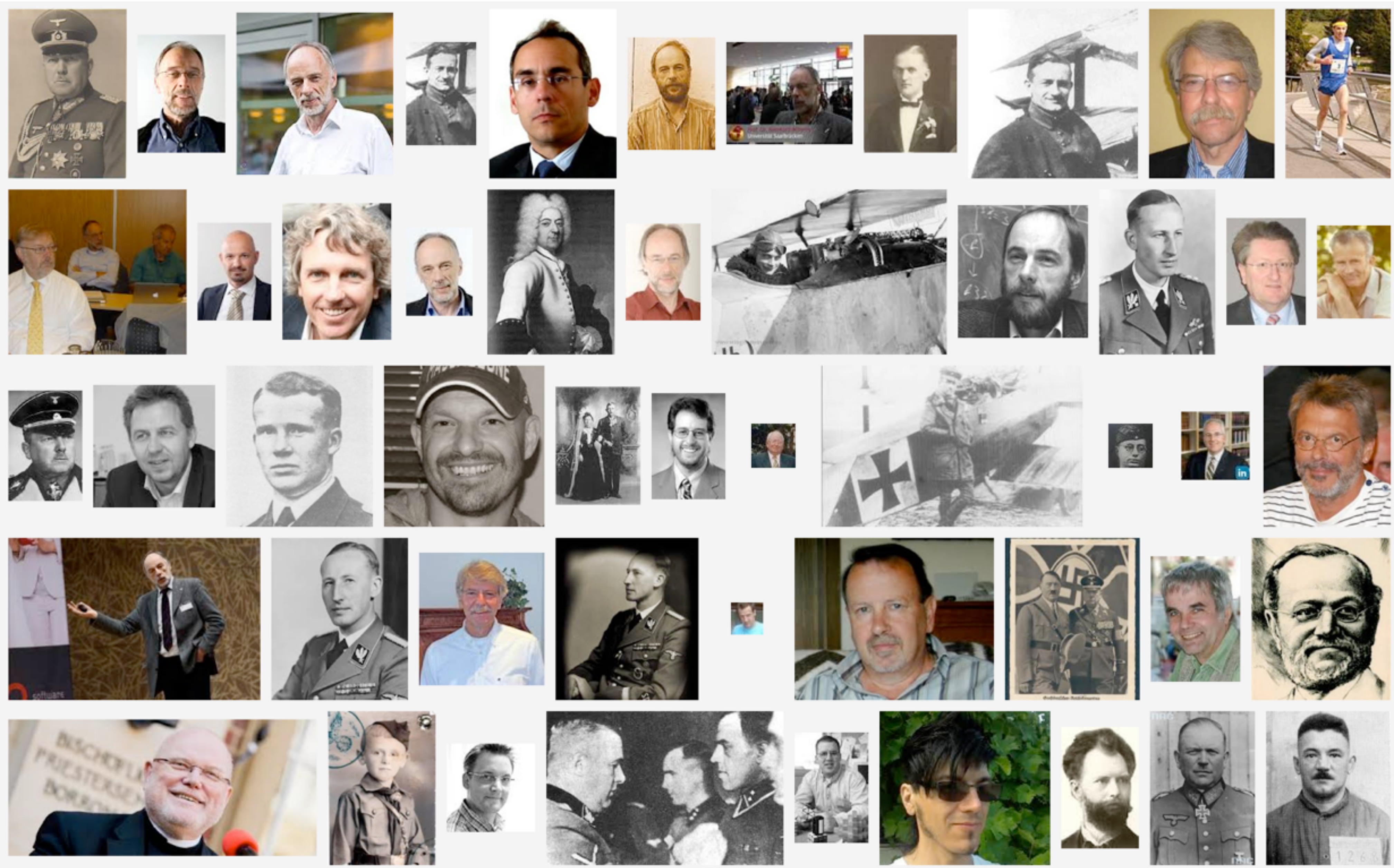
pcousot@cims.nyu.edu

Farewell Colloquium on the Occasion of Reinhard Wilhelm's 68th birthday

Saarbrüchen, November, 28th 2014

You said Reinhard Wilhelm?

But who is Reinhard Wilhelm?



But who is Reinhard Wilhelm?

But who is Reinhard Wilhelm?

1. You have undestood the limitations of ``Big data” and ``Advanced machine learning”

But who is Reinhard Wilhelm?

1. You have undestood the limitations of ``Big data” and ``Advanced machine learning”
2. This is THE Reinhard Wilhelm:

But who is Reinhard Wilhelm?

1. You have undestood the limitations of ``Big data'' and ``Advanced machine learning''
2. This is THE Reinhard Wilhelm:

sorry, this was 2 months ago on Wikipedia, thanks to the true Reinhard Wilhelm for updating his picture last month!

But who is Reinhard Wilhelm?

1. You have undestood the limitations of ``Big data” and ``Advanced machine learning”
2. This is THE Prof. em. Dr. Dr. h.c. Reinhard Wilhelm:

Reinhard Wilhelm

Reinhard Wilhelm, 2014

Born	5 June 1946 (age 68) Finnentrop, Germany
Fields	Computer Scientist
Institutions	Saarland University
Alma mater	University of Münster, Stanford University, Technical University Munich
Known for	compiler technology
Notable awards	Konrad Zuse Medal (2009) Merit Cross on Ribbon (2010) ACM Distinguished Service Award (2011)

There is only one, the proof is by Google

There is only one, the proof is by Google

Images for Prof. em. Dr. Dr. h.c. Reinhard Wilhelm   Report images

More images for Prof. em. Dr. Dr. h.c. Reinhard Wilhelm

And more …

• • •

Great Achievements

Great Achievements of Reinhard (I)

Great Achievements of Reinhard (II)

Book Format: Hardcover | Kindle Edition

Compiler Design: Syntactic and Semantic Analysis May 28, 2013
by Reinhard Wilhelm and Helmut Seidl

Hardcover
$55.99 ~~$69.99~~ Prime
Only 7 left in stock - order soon.

More Buying Choices
$35.00 used & new (54 offers)

Kindle Edition
from $19.56 to rent
$49.99 to buy
7-day free trial available

Compiler Design: Virtual Machines Jan 19, 2011
by Reinhard Wilhelm and Helmut Seidl

Hardcover
$47.96 ~~$69.99~~ Prime
Only 6 left in stock - order soon.

More Buying Choices
$37.63 used & new (42 offers)

Kindle Edition
from $16.79 to rent
$45.56 to buy
7-day free trial available

Compiler Design: Code Generation and Machine-Level Optimization Jun 24, 2015
by Sebastian Hack and Reinhard Wilhelm

Hardcover
$56.95 ~~$59.95~~ Prime
Available for Pre-order

Compiler Design Aug 12, 2012
by Helmut Seidl and Reinhard Wilhelm

Kindle Edition
$49.99
Auto-delivered wirelessly

Hardcover
$55.96 ~~$69.99~~ Prime
Get it by Friday, Nov 21

More Buying Choices
$48.58 used & new (49 offers)

Great Achievements of Reinhard (II)

Book Format: Hardcover | Kindle Edition

Compiler Design: Syntactic and Semantic Analysis May 28, 2013
by Reinhard Wilhelm and Helmut Seidl

Hardcover
$55.99 ~~$69.99~~ Prime
Only 7 left in stock - order soon.

More Buying Choices
$35.00 used & new (54 offers)

Kindle Edition
from $19.56 to rent
$49.99 to buy
7-day free trial available

Compiler Design: Virtual Machines Jan 19, 2011
by Reinhard Wilhelm and Helmut Seidl

Hardcover
$47.96 ~~$69.99~~ Prime
Only 6 left in stock - order soon.

More Buying Choices
$37.63 used & new (42 offers)

Kindle Edition
from $16.79 to rent
$45.56 to buy
7-day free trial available

Compiler Design: Code Generation and Machine-Level Optimization
by Sebastian Hack and Reinhard Wilhelm

Hardcover
$56.95 ~~$59.95~~ Prime
Available for Pre-order

Jun 24, 2015

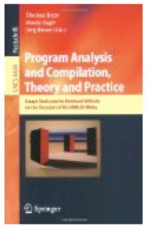
?

Compiler Design Aug 12, 2012
by Helmut Seidl and Reinhard Wilhelm

Kindle Edition
$49.99
Auto-delivered wirelessly

Hardcover
$55.96 ~~$69.99~~ Prime
Get it by **Friday, Nov 21**

More Buying Choices
$48.58 used & new (49 offers)



[Program Analysis and Compilation, Theory and Practice: Essays Dedicated to Reinhard Wilhelm on the Occasion of...] Mar 30, 2007
by Thomas Reps and Mooly Sagiv

Paperback
$82.57 ~~$109.00~~ Prime
Get it by **Monday, Nov 24**
More Buying Choices
$29.95 used & new (40 offers)

Trade-in eligible for an Amazon gift card
FREE Shipping

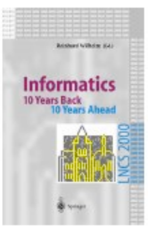
By Reinhard Wilhelm Compiler Design: Analysis and Transformation (1st) Aug 29, 2012
by Reinhard Wilhelm

Hardcover
$94.95 used & new (6 offers)

Grundlagen der Programmiersprachen (Leitfäden und Monographien der Informatik) (German Edition) Oct 1, 1986
by Kurt Mehlhorn and Reinhard Wilhelm

Paperback
$61.54 ~~$69.95~~ Prime
Get it by **Monday, Nov 24**
More Buying Choices
$33.37 used & new (14 offers)

FREE Shipping



Informatics: 10 Years Back. 10 Years Ahead (Lecture Notes in Computer Science) Mar 23, 2001
by Reinhard Wilhelm

Paperback
$60.01 ~~$109.00~~ Prime
Get it by **Monday, Nov 24**
More Buying Choices
$53.38 used & new (25 offers)

Trade-in eligible for an Amazon gift card
FREE Shipping

Informatik. Grundlagen, Anwendungen, Perspektiven. Apr 1, 1996
by Wilhelm Reinhard

Paperback
$0.01 used & new (2 offers)

Ç?bersetzerbau. Virtuelle Maschinen 2007
by REINHARD WILHELM

Paperback
$32.40 used & new (1 offer)

Les compilateurs: Théorie, construction, génération by Wilhelm (Dec 1, 1997)

Formats	Price	New	Used
Paperback			
Currently unavailable			

Compiler Construction: 10th International Conference, CC 2001 Held as Part of the Joint European Conferences on... Mar 21, 2001
by Reinhard Wilhelm

Paperback
$74.23 ~~$109.00~~ Prime
Get it by **Friday, Nov 21**
More Buying Choices
$32.63 used & new (35 offers)

FREE Shipping

Übersetzerbau: Band 2: Syntaktische und semantische Analyse (eXamen.press) (German Edition) Dec 31, 2012
by Reinhard Wilhelm and Helmut Seidl

Paperback
$26.82 ~~$29.95~~ Prime
Get it by **Monday, Nov 24**
More Buying Choices
$19.82 used & new (34 offers)

FREE Shipping on orders over $35

Kindle Edition
$14.41
Auto-delivered wirelessly

Übersetzerbau: Band 3: Analyse und Transformation (eXamen.press) (German Edition) Dec 14, 2009
by Helmut Seidl and Reinhard Wilhelm

Paperback
$27.45 ~~$29.99~~ Prime
Get it by **Friday, Nov 21**
More Buying Choices
$21.79 used & new (17 offers)

FREE Shipping on orders over $35

Übersetzerbau: Virtuelle Maschinen (eXamen.press) (German Edition) Jan 21, 2012
by Reinhard Wilhelm and Helmut Seidl

Paperback
$26.81 ~~$29.99~~ Prime
Usually ships in 1 to 3 weeks
More Buying Choices
$22.07 used & new (15 offers)

Übersetzerbau: Theorie, Konstruktion, Generierung (Springer-Lehrbuch) (German Edition) Oct 16, 1992
by Reinhard Wilhelm and Dieter Maurer

Hardcover
$51.36 ~~$59.00~~ Prime
Temporarily out of stock. Order now and we'll deliver when available.
More Buying Choices
$51.36 used & new (2 offers)

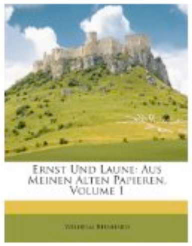
FREE Shipping

Not forgetting…Wilhelm Reinhard

Bekenntnisse Aus Leben Und Meinungen, Volume 1 (German Edition) Sep 5, 2011
by Wilhelm Reinhard

Paperback
$29.13 ~~$34.75~~ Prime

More Buying Choices
$19.95 used & new (10 offers)



Ernst Und Laune: Aus Meinen Alten Papieren, Volume 1 (German Edition) Apr 16, 2012
by Wilhelm Reinhard

Paperback FRI
$21.24 ~~$24.75~~ Prime

More Buying Choices
$15.57 used & new (9 offers)

Die Bundes-acte Über Ob, Wann Und Wie? Deutscher Landstände... (German Edition) Feb 16, 2012
by Wilhelm Reinhard

Paperback FREE Shipping on orders over $35
$18.68 ~~$20.75~~ Prime
Usually ships in 2 to 4 weeks

More Buying Choices
$12.80 used & new (12 offers)

Ernst Und Laune: Aus Meinen Alten Papieren, Volume 2 (German Edition) Apr 17, 2012
by Wilhelm Reinhard

Paperback FREE Shipping on orders over $35
$24.73 ~~$29.75~~ Prime
Usually ships in 2 to 4 weeks

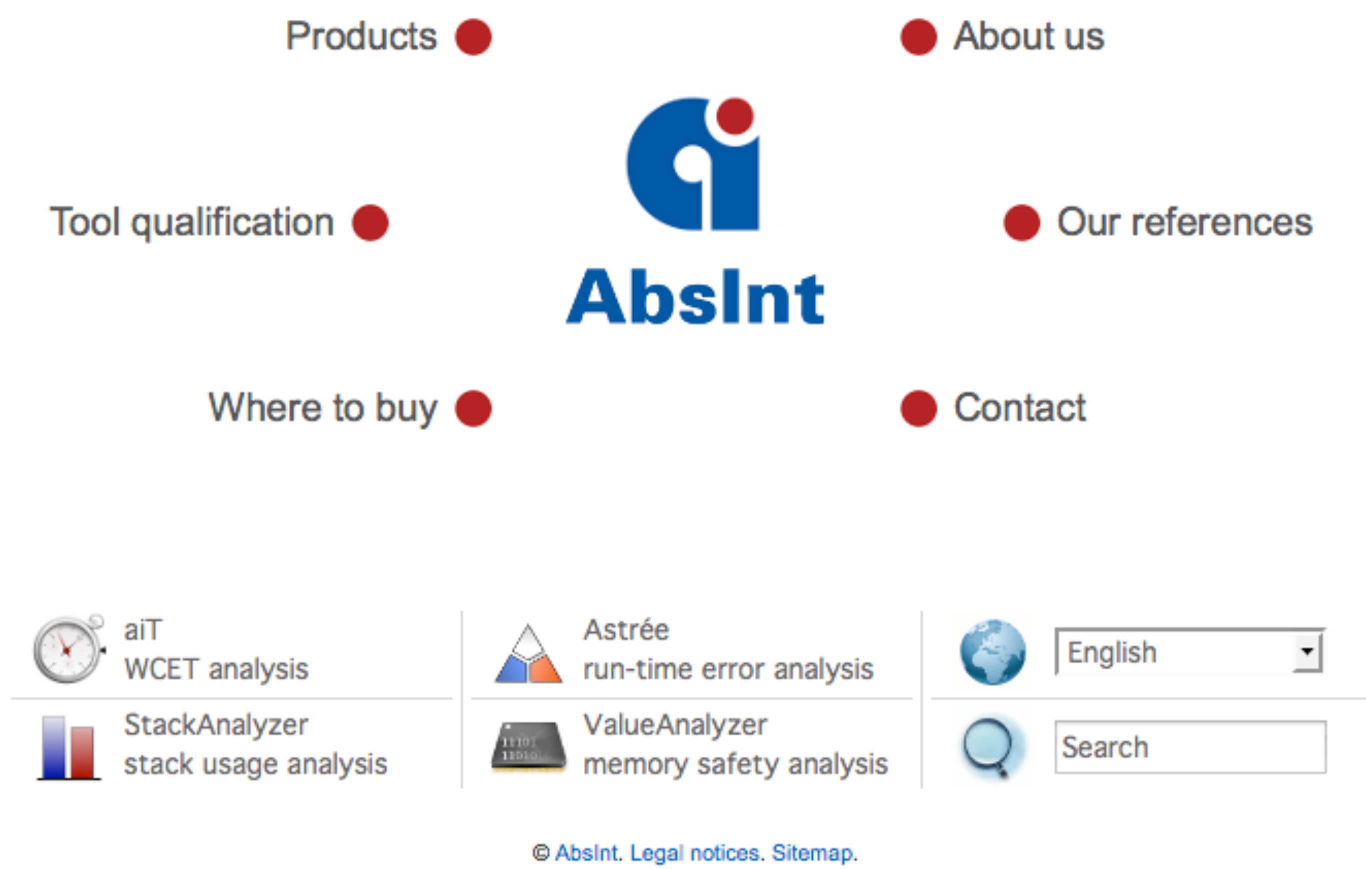
More Buying Choices
$21.22 used & new (8 offers)

Massage und physikalische Behandlungsmethoden (Heidelberger Taschenbücher) (German Edition) Jan 1, 1967
by Wilhelm Reinhard

Paperback FREE Shipping
$66.45 ~~$69.95~~ Prime
Temporarily out of stock. Order now and we'll deliver when available.

More Buying Choices
$53.50 used & new (16 offers)

Great Achievements of Reinhard (III)



Great Achievements of Reinhard (IV)

home | browse | search | about

dblp computer science bibliography

Reinhard Wilhelm

[–] **Author information**   -- show all --   by year   Trier 1

- *affiliation:* Saarland University, Compiler Design Lab
- *affiliation:* Schloss Dagstuhl - Leibniz Center for Informatics

[–] **2010 – today**

2014

[j41] Reinhard Wilhelm, Daniel Grund: **Computation takes time, but how much?** Commun. ACM 57(2): 94-103 (2014)

[j40] Philip Axer, Rolf Ernst, Heiko Falk, Alain Girault, Daniel Grund, Nan Guan, Bengt Jonsson, Peter Marwedel, Jan Reineke, Christine Rochange, Maurice Sebastian, Reinhard von Hanxleden, Reinhard Wilhelm, Wang Yi: **Building timing predictable embedded systems.** ACM Trans. Embedded Comput. Syst. 13(4): 82 (2014)

[c97] Jan Reineke, Reinhard Wilhelm: **Impact of resource sharing on performance and performance prediction.** DATE 2014: 1-2

[c96] Reinhard Wilhelm: **Software Quality Assurance by Static Program Analysis.** SWQD 2014: 1-11

2013

[b10] Reinhard Wilhelm, Helmut Seidl, Sebastian Hack: **Compiler Design - Syntactic and Semantic Analysis.** Springer 2013, ISBN 978-3-642-17539-8, pp. I-X, 1-225

[j39] Joseph Sifakis, Lothar Thiele, Reinhard Wilhelm: **Introduction to the special section on rigorous embedded systems design.** ACM Trans. Embedded Comput. Syst. 12(1s): 41 (2013)

[c95] Deepak Kapur, Robert Nieuwenhuis, Andrei Voronkov, Christoph Weidenbach, Reinhard Wilhelm: **Harald Ganzinger's Legacy:**

'10 '00 '90 '80 '70

Contributions to Logics and Programming. Programming Logics 2013: 1-18

[c94] Jörg Kreiker, Thomas W. Reps, Noam Rinetzky, Mooly Sagiv, Reinhard Wilhelm, Eran Yahav: **Interprocedural Shape Analysis for Effectively Cutpoint-Free Programs.** Programming Logics 2013: 414-445

[c93] Andreas Abel, Florian Benz, Johannes Doerfert, Barbara Dörr, Sebastian Hahn, Florian Haupenthal, Michael Jacobs, Amir H. Moin, Jan Reineke, Bernhard Schommer, Reinhard Wilhelm: **Impact of Resource Sharing on Performance and Performance Prediction: A Survey.** CONCUR 2013: 25-43

2012

[b9] Helmut Seidl, Reinhard Wilhelm, Sebastian Hack: **Compiler Design - Analysis and Transformation.** Springer 2012, ISBN 978-3-642-17547-3, pp. I-XII, 1-177

[j38] Ernst-Rüdiger Olderog, Reinhard Wilhelm: **Turing und die Verifikation.** Informatik Spektrum 35(4): 271-279 (2012)

[c92] Reinhard Wilhelm, Philipp Lucas, Oleg Parshin, Lili Tan, Björn Wachter: **Improving the Precision of WCET Analysis by Input Constraints and Model-Derived Flow Constraints.** Advances in Real-Time Systems 2012: 123-143

[c91] Reinhard Wilhelm, Jan Reineke: **Embedded systems: Many cores - Many problems.** SIES 2012: 176-180

[e13] Reinhard Wilhelm, Heiko Falk, Wang Yi (Eds.): **SIGPLAN/SIGBED Conference on Languages, Compilers and Tools for Embedded Systems 2012, LCTES '12, Beijing, China - June 12 - 13, 2012.** ACM 2012, ISBN 978-1-4503-1212-7

2011

[j37] Jörg Kreiker, Andrzej Tarlecki, Moshe Y. Vardi, Reinhard Wilhelm: **Modeling, Analysis, and Verification - The Formal Methods Manifesto 2010 (Dagstuhl Perspectives Workshop 10482).** Dagstuhl Manifestos 1(1): 21-40 (2011)

[c90] Daniel Grund, Jan Reineke, Reinhard Wilhelm: **A Template for Predictability Definitions with Supporting Evidence.** PPES 2011: 22-31

[e12] Philipp Lucas, Lothar Thiele, Benoit Triquet, Theo Ungerer, Reinhard Wilhelm (Eds.): **Bringing Theory to Practice: Predictability and Performance in Embedded Systems, DATE Workshop PPES 2011, March 18, 2011, Grenoble, France.**

OASICS 18, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany 2011, ISBN 978-3-939897-28-6

'10 '00 '90 '80 '70

2010

[b8] Reinhard Wilhelm, Helmut Seidl: **Compiler Design - Virtual Machines.** Springer 2010, ISBN 978-3-642-14908-5, pp. I-XIII, 1-187

[c89] Reinhard Wilhelm, Matteo Maffei: **Ubiquitous Verification of Ubiquitous Systems.** SEUS 2010: 47-58

[c88] Reinhard Wilhelm, Sebastian Altmeyer, Claire Burguière, Daniel Grund, Jörg Herter, Jan Reineke, Björn Wachter, Stephan Wilhelm: **Static Timing Analysis for Hard Real-Time Systems.** VMCAI 2010: 3-22

[–] **2000 – 2009**

2009

[j36] Christoph M. Kirsch, Reinhard Wilhelm: **EMSOFT 2007 guest editors' introduction.** Design Autom. for Emb. Sys. 13(1-2): 3-4 (2009)

[j35] Roswitha Bardohl, Christian Lindig, Angelika Müller-von Brochowski, Reinhard Wilhelm: **Editorial.** Computer Science - R&D 23(1): 3-5 (2009)

[j34] Reinhard Wilhelm, Daniel Grund, Jan Reineke, Marc Schlickling, Markus Pister, Christian Ferdinand: **Memory Hierarchies, Pipelines, and Buses for Future Architectures in Time-Critical Embedded Systems.** IEEE Trans. on CAD of Integrated Circuits and Systems 28(7): 966-978 (2009)

[c87] Thomas Gawlitza, Jérôme Leroux, Jan Reineke, Helmut Seidl, Grégoire Sutre, Reinhard Wilhelm: **Polynomial Precise Interval Analysis Revisited.** Efficient Algorithms 2009: 422-437

[c86] Thomas Heinz, Reinhard Wilhelm: **Towards device emulation code generation.** LCTES 2009: 109-118

[c85] Reinhard Wilhelm: **The PROMPT design principles for predictable multi-core architectures.** SCOPES 2009: 31-32

[c84] Niklas Holsti, Guillem Bernat, Christian Ferdinand, Peter P. Puschner, Reinhard Wilhelm: **Teaching WCET Analysis in Academia and Industry: A Panel Discussion.** WCET 2009

[p3] Reinhard Wilhelm: **Determining Bounds on Execution Times.** Embedded Systems Design and Verification 2009: 9

2008

'10 '00 '90 '80 '70

[j33] Reinhard Wilhelm, Jakob Engblom, Andreas Ermedahl, Niklas Holsti, Stephan Thesing, David B. Whalley, Guillem Bernat, Christian Ferdinand, Reinhold Heckmann, Tulika Mitra, Frank Mueller, Isabelle Puaut, Peter P. Puschner, Jan Staschulat, Per Stenström: **The worst-case execution-time problem - overview of methods and survey of tools.** ACM Trans. Embedded Comput. Syst. 7(3) (2008)

[c83] Reinhard Wilhelm, Björn Wachter: **Abstract Interpretation with Applications to Timing Validation.** CAV 2008: 22-36

[c82] E. Frank, Reinhard Wilhelm, Rolf Ernst, Alberto L. Sangiovanni-Vincentelli, Marco Di Natale: **Methods, Tools and Standards for the Analysis, Evaluation and Design of Modern Automotive Architectures.** DATE 2008: 659-663

[c81] Daniel Kästner, Reinhard Wilhelm, Reinhold Heckmann, Marc Schlickling, Markus Pister, Marek Jersak, Kai Richter, Christian Ferdinand: **Timing Validation of Automotive Software.** ISoLA 2008: 93-107

[c80] Reinhard Wilhelm: **Architectural Influence on the Timing Predictability.** MMB 2008: 11-12

[c79] Sebastian Altmeyer, Christian Humbert, Björn Lisper, Reinhard Wilhelm: **Parametric Timing Analysis for Complex Architectures.** RTCSA 2008: 367-376

[c78] Sascha A. Parduhn, Raimund Seidel, Reinhard Wilhelm: **Algorithm visualization using concrete and abstract shape graphs.** SOFTVIS 2008: 33-36

2007

[j32] Bernd Becker, Andreas Podelski, Werner Damm, Martin Fränzle, Ernst-Rüdiger Olderog, Reinhard Wilhelm: **SFB/TR 14 AVACS - Automatic Verification and Analysis of Complex Systems (Der Sonderforschungsbereich/Transregio 14 AVACS - Automatische Verifikation und Analyse komplexer Systeme).** it - Information Technology 49(2): 118-126 (2007)

[j31] Jan Reineke, Daniel Grund, Christoph Berg, Reinhard Wilhelm: **Timing predictability of cache replacement policies.** Real-Time Systems 37(2): 99-122 (2007)

[j30] Greta Yorsh, Thomas W. Reps, Mooly Sagiv, Reinhard Wilhelm: **Logical characterizations of heap abstractions.** ACM Trans. Comput. Log. 8(1) (2007)

'10 '00 '90 '80 '70

[c77] Christoph M. Kirsch, Reinhard Wilhelm: **Grand challenges in embedded software.** EMSOFT 2007: 2-6

[c76] Nicolas Fritz, Philipp Lucas, Reinhard Wilhelm: **Exploiting SIMD Parallelism with the CGiSCompiler Framework.** LCPC 2007: 246-260

[c75] Jörg Bauer, Reinhard Wilhelm: **Static Analysis of Dynamic Communication Systems by Partner Abstraction.** SAS 2007: 249-264

[p2] Thomas W. Reps, Mooly Sagiv, Reinhard Wilhelm: **Shape Analysis and Applications.** The Compiler Design Handbook, 2nd ed. 2007: 12

[e11] Christoph M. Kirsch, Reinhard Wilhelm (Eds.): **Proceedings of the 7th ACM & IEEE International conference on Embedded software, EMSOFT 2007, September 30 - October 3, 2007, Salzburg, Austria.** ACM 2007, ISBN 978-1-59593-825-1

[e10] Reinhard Wilhelm (Ed.): **5th Intl. Workshop on Worst-Case Execution Time (WCET) Analysis, July 5, 2005, Palma de Mallorca, Spain.** OASICS 1, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany 2007

2006

[j29] Eran Yahav, Thomas W. Reps, Shmuel Sagiv, Reinhard Wilhelm: **Verifying Temporal Heap Properties Specified via Evolution Logic.** Logic Journal of the IGPL 14(5): 755-783 (2006)

[c74] Christian Ferdinand, Reinhold Heckmann, Hans-Jörg Wolff, Christian Renz, Oleg Parshin, Reinhard Wilhelm: **Towards Model-Driven Development of Hard Real-Time Systems.** ASWSD 2006: 145-160

[c73] Philipp Lucas, Nicolas Fritz, Reinhard Wilhelm: **The CGiS Compiler-A Tool Demonstration.** CC 2006: 105-108

[c72] Serge Autexier, Stephan Merz, Leendert W. N. van der Torre, Reinhard Wilhelm, Pierre Wolper: **Preface -- Workshop Trustworthy Software 2006.** Trustworthy Software 2006

[c71] Jörg Bauer, Reinhard Wilhelm: **Abstract Interpretation of Graph Transformation.** Simulation and Verification of Dynamic Systems 2006

[c70] Jörg Bauer, Reinhard Wilhelm: **Analysis of Dynamic Communicating Systems by Hierarchical Abstraction.** Softwar

'10 '00 '90 '80 '70

Verification: Infinite-State Model Checking and Static Program Analysis 2006

[c69] Jochen Eisinger, Ilia Polian, Bernd Becker, Alexander Metzner, Stephan Thesing, Reinhard Wilhelm: **Automatic Identification of Timing Anomalies for Cycle-Accurate Worst-Case Execution Time Analysis.** DDECS 2006: 15-20

[c68] Philipp Lucas, Nicolas Fritz, Reinhard Wilhelm: **The Development of the Data-Parallel GPU Programming Language CGiS.** International Conference on Computational Science (4) 2006: 200-203

[c67] Werner Damm, Alexander Metzner, Friedrich Eisenbrand, Gennady Shmonin, Reinhard Wilhelm, Sebastian Winkel: **Mapping Task-Graphs on Distributed ECU Networks: Efficient Algorithms for Feasibility and Optimality.** RTCSA 2006: 87-90

[c66] Jan Reineke, Björn Wachter, Stephan Thesing, Reinhard Wilhelm, Ilia Polian, Jochen Eisinger, Bernd Becker: **A Definition and Classification of Timing Anomalies.** WCET 2006

[e9] Serge Autexier, Stephan Merz, Leendert W. N. van der Torre, Reinhard Wilhelm, Pierre Wolper (Eds.): **Workshop "Trustworthy Software" 2006, May 18-19, 2006, Saarland University, Saarbrücken, Germany.** OASICS 3, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany 2006

2005

[j28] Paul Caspi, Alberto L. Sangiovanni-Vincentelli, Luís Almeida, Albert Benveniste, Bruno Bouyssounouse, Giorgio C. Buttazzo, Ivica Crnkovic, Werner Damm, Jakob Engblom, Gerhard Fohler, Marisol García-Valls, Hermann Kopetz, Yassine Lakhnech, François Laroussinie, Luciano Lavagno, Giuseppe Lipari, Florence Maraninchi, Philipp Peti, Juan Antonio de la Puente, Norman Scaife, Joseph Sifakis, Robert de Simone, Martin Törngren, Paulo Veríssimo, Andy J. Wellings, Reinhard Wilhelm, Tim A. C. Willemse, Wang Yi: **Guidelines for a graduate curriculum on embedded software and systems.** ACM Trans. Embedded Comput. Syst. 4(3): 587-611 (2005)

[c65] Noam Rinetzky, Jörg Bauer, Thomas W. Reps, Shmuel Sagiv, Reinhard Wilhelm: **A semantics for procedure local heaps and its abstractions.** POPL 2005: 296-309

[c64] Dierk Johannes, Raimund Seidel, Reinhard Wilhelm: **Algorithm animation using shape analysis: visualising abstract**

executions. SOFTVIS 2005: 17-26

[c63] Nurit Dor, John Field, Denis Gopan, Tal Lev-Ami, Alexey Loginov, Roman Manevich, G. Ramalingam, Thomas W. Reps, Noam Rinetzky, Mooly Sagiv, Reinhard Wilhelm, Eran Yahav, Greta Yorsh: **Automatic Verification of Strongly Dynamic Software Systems.** VSTTE 2005: 82-92

[c62] Reinhard Wilhelm: **2005 WCET Preface - Message from the Workshop Chair.** WCET 2005

2004

[j27] Lothar Thiele, Reinhard Wilhelm: **Design for Timing Predictability.** Real-Time Systems 28(2-3): 157-177 (2004)

[c61] Christian Ferdinand, Reinhold Heckmann, Reinhard Wilhelm: **Analyzing the Worst-Case Execution Time by Abstract Interpretation of Executable Code.** ASWSD 2004: 1-14

[c60] Abdur Rakib, Oleg Parshin, Stephan Thesing, Reinhard Wilhelm: **Component-Wise Instruction-Cache Behavior Prediction.** ATVA 2004: 211-229

[c59] Thomas W. Reps, Shmuel Sagiv, Reinhard Wilhelm: **Static Program Analysis via 3-Valued Logic.** CAV 2004: 15-30

[c58] Christoph Berg, Jakob Engblom, Reinhard Wilhelm: **Requirements for and Design of a Processor with Predictable Timing.** Design of Systems with Predictable Behaviour 2004

[c57] Lothar Thiele, Reinhard Wilhelm: **Abstracts Collection.** Design of Systems with Predictable Behaviour 2004

[c56] Lothar Thiele, Reinhard Wilhelm: **Design for Time-Predictability.** Design of Systems with Predictable Behaviour 2004

[c55] Reinhard Wilhelm: **Timing Analysis and Timing Predictability.** FMCO 2004: 317-323

[c54] Reinhard Wilhelm: **Formal Analysis of Processor Timing Models.** SPIN 2004: 1-4

[c53] Reinhard Wilhelm: **Why AI + ILP Is Good for WCET, but MC Is Not, Nor ILP Alone.** VMCAI 2004: 309-322

[e8] Lothar Thiele, Reinhard Wilhelm (Eds.): **Perspectives Workshop: Design of Systems with Predictable Behaviour, 16.-19. November 2003.** Dagstuhl Seminar Proceedings 03471, IBFI, Schloss Dagstuhl, Germany 2004

2003

[j26] Reinhold Heckmann, Marc Langenbach, Stephan Thesing, Reinhard Wilhelm: **The influence of processor architecture on the design and the results of WCET tools.** Proceedings of the IEEE 91(7): 1038-1054 (2003)

[c52] Stephan Thesing, Jean Souyris, Reinhold Heckmann, Famantanantsoa Randimbivololona, Marc Langenbach, Reinhard Wilhelm, Christian Ferdinand: **An Abstract Interpretation-Based Timing Validation of Hard Real-Time Avionics Software.** DSN 2003: 625-632

[c51] Eran Yahav, Thomas W. Reps, Shmuel Sagiv, Reinhard Wilhelm: **Verifying Temporal Heap Properties Specified via Evolution Logic.** ESOP 2003: 204-222

[c50] Reinhard Wilhelm: **Run-Time Guarantees for Real-Time Systems.** FORMATS 2003: 166-167

[c49] Christian Ferdinand, Daniel Kästner, Florian Martin, Marc Langenbach, Martin Sicks, Stephan Wilhelm, Reinhold Heckmann, Nicolas Fritz, Stephan Thesing, Frank Fontaine, Henrik Theiling, Michael Schmidt, Alexander A. Evstiougov-Babaev, Reinhard Wilhelm: **Validierung des Zeitverhaltens von kritischer Echtzeit-Software.** GI Jahrestagung (1) 2003: 335-339

[c48] Christian Ferdinand, Reinhold Heckmann, Henrik Theiling, Reinhard Wilhelm: **Convenient User Annotations for a WCET Tool.** WCET 2003: 17-20

[c47] Reinhard Wilhelm, Jakob Engblom, Stephan Thesing, David B. Whalley: **Industrial Requirements for WCET Tools - Answers to the ARTIST Questionnaire.** WCET 2003: 39-43

[i1] Greta Yorsh, Thomas W. Reps, Shmuel Sagiv, Reinhard Wilhelm: **Logical Characterizations of Heap Abstractions.** CoRR cs.LO/0312014 (2003)

2002

[j25] Shmuel Sagiv, Thomas W. Reps, Reinhard Wilhelm: **Parametric shape analysis via 3-valued logic.** ACM Trans. Program. Lang. Syst. 24(3): 217-298 (2002)

[p1] Reinhard Wilhelm, Thomas W. Reps, Shmuel Sagiv: **Shape Analysis and Applications.** The Compiler Design Handbook 2002: 175-218

2001

[c46] Reinhard Wilhelm, Tomasz Müldner, Raimund Seidel: **Algorithm Explanation: Visualizing Abstract States and Invariants.**

'10
'00
'90
'80
'70

Software Visualization 2001: 381-394

[c45] Christian Ferdinand, Reinhold Heckmann, Marc Langenbach, Florian Martin, Michael Schmidt, Henrik Theiling, Stephan Thesing, Reinhard Wilhelm: **Reliable and Precise WCET Determination for a Real-Life Processor.** EMSOFT 2001: 469-485

[e7] Reinhard Wilhelm (Ed.): **Compiler Construction, 10th International Conference, CC 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2-6, 2001, Proceedings.** Lecture Notes in Computer Science 2027, Springer 2001, ISBN 3-540-41861-X

[e6] Reinhard Wilhelm (Ed.): **Informatics - 10 Years Back. 10 Years Ahead.** Lecture Notes in Computer Science 2000, Springer 2001, ISBN 3-540-41635-8

2000

[j24] Henrik Theiling, Christian Ferdinand, Reinhard Wilhelm: **Fast and Precise WCET Prediction by Separated Cache and Path Analyses.** Real-Time Systems 18(2/3): 157-179 (2000)

[j23] Beatrix Braune, Reinhard Wilhelm: **Focusing in Algorithm Explanation.** IEEE Trans. Vis. Comput. Graph. 6(1): 1-7 (2000)

[c44] Reinhard Wilhelm, Shmuel Sagiv, Thomas W. Reps: **Shape Analysis.** CC 2000: 1-17

[c43] Tal Lev-Ami, Thomas W. Reps, Shmuel Sagiv, Reinhard Wilhelm: **Putting static analysis to work for verification: A case study.** ISSTA 2000: 26-38

[–] 1990 – 1999

1999

[j22] Reinhard Wilhelm: **Timing Analysis and Validation for Real-Time Systems - Guest Editor's Introduction.** Real-Time Systems 17(2-3): 127-129 (1999)

[j21] Christian Ferdinand, Reinhard Wilhelm: **Efficient and Precise Cache Behavior Prediction for Real-Time Systems.** Real-Time Systems 17(2-3): 131-181 (1999)

[j20] Christian Ferdinand, Florian Martin, Reinhard Wilhelm, Martin Alt: **Cache Behavior Prediction by Abstract Interpretation.** Sci. Comput. Program. 35(2): 163-189 (1999)

'10
'00
'90
'80
'70

[c42] Christian Ferdinand, Daniel Kästner, Marc Langenbach, Florian Martin, Michael Schmidt, Jörn Schneider, Henrik Theiling, Stephan Thesing, Reinhard Wilhelm: **Run-Time Guarantees for Real-Time Systems - The USES Approach.** GI Jahrestagung 1999: 410-419

[c41] Shmuel Sagiv, Thomas W. Reps, Reinhard Wilhelm: **Parametric Shape Analysis via 3-Valued Logic.** POPL 1999: 105-118

[c40] Beatrix Braune, Stephan Diehl, Andreas Kerren, Reinhard Wilhelm: **Animation of the Generation and Computation of Finite Automata for Learning Software.** WIA 1999: 39-47

[e5] Y. Annie Liu, Reinhard Wilhelm (Eds.): **Proceedings of the ACM SIGPLAN 1999 Workshop on Languages, Compilers, and Tools for Embedded Systems (LCTES'99), Atlanta, Georgia, USA, May 5, 1999.** ACM 1999, ISBN 1-58113-136-4

1998

[j19] Shmuel Sagiv, Nissim Francez, Michael Rodeh, Reinhard Wilhelm: **A Logic-Based Approach to Program Flow Analysis.** Acta Inf. 35(6): 457-504 (1998)

[j18] Shmuel Sagiv, Thomas W. Reps, Reinhard Wilhelm: **Solving Shape-Analysis Problems in Languages with Destructive Updating.** ACM Trans. Program. Lang. Syst. 20(1): 1-50 (1998)

[c39] Florian Martin, Martin Alt, Reinhard Wilhelm, Christian Ferdinand: **Analysis of Loops.** CC 1998: 80-94

[c38] Christian Ferdinand, Reinhard Wilhelm: **On Predicting Data Cache Behavior for Real-Time Systems.** LCTES 1998: 16-30

1997

[b7] Reinhard Wilhelm, Dieter Maurer: **Übersetzerbau - Theorie, Konstruktion, Generierung, 2. Auflage.** Springer 1997, ISBN 3-540-61692-6

[j17] Reinhold Heckmann, Reinhard Wilhelm: **A Functional Description of TEX's Formula Layout.** J. Funct. Program. 7(5): 451-485 (1997)

[j16] Reinhard Wilhelm: **Program Analysis: A Toolmaker's Perspective.** SIGPLAN Notices 32(1): 120-121 (1997)

1996

[b6] Reinhard Wilhelm, Reinhold Heckmann: **Grundlagen der Dokumentenverarbeitung.** Addison-Wesley 1996, ISBN 978-3-89319-877-1, pp. I-XIV, 1-274

[j15] Reinhard Wilhelm: **Program Analysis - A Toolmaker's Perspective.** ACM Comput. Surv. 28(4es): 177 (1996)

'10 '00 '90 '80 '70

[c37] Reinhard Wilhelm, Martin Alt, Florian Martin, Martin Raber: **Parallel Implementation of Functional Languages.** LOMAPS 1996: 279-295

[c36] Shmuel Sagiv, Thomas W. Reps, Reinhard Wilhelm: **Solving Shape-Analysis Problems in Languages with Destructive Updating.** POPL 1996: 16-31

[c35] Martin Alt, Christian Ferdinand, Florian Martin, Reinhard Wilhelm: **Cache Behavior Prediction by Abstract Interpretation.** SAS 1996: 52-66

[e4] Reinhard Wilhelm (Ed.): **Informatik: Grundlagen - Amwendungen - Perspektiven [Forum "Perspektiven der Informatik", Dagstuhl, November 1993].** Verlag C. H. Beck 1996, ISBN 3-406-40338-7

1995

[b5] Reinhard Wilhelm, Dieter Maurer: **Compiler design.** Addison-Wesley 1995, ISBN 978-0-201-42290-0, pp. I-XV, 1-606

[j14] Yosi Ben-Asher, Gudula Rünger, Assaf Schuster, Reinhard Wilhelm: **2DT-FP: A parallel functional programming language on two-dimensional data.** International Journal of Parallel Programming 23(5): 389-422 (1995)

[c34] Georg Sander, Martin Alt, Christian Ferdinand, Reinhard Wilhelm: **CLaX - A Visualized Compiler.** Graph Drawing 1995: 459-462

[c33] Thomas Rauber, Gudula Rünger, Reinhard Wilhelm: **An application specific parallel programming paradigm.** HPCN Europe 1995: 735-740

1994

[b4] Reinhard Wilhelm, Dieter Maurer: **Les compilateurs - théorie, construction, génération.** Masson 1994, ISBN 978-2-225-84615-1, pp. I-XII, 1-612

[j13] Christian Ferdinand, Helmut Seidl, Reinhard Wilhelm: **Tree Automata for Code Selection.** Acta Inf. 31(8): 741-760 (1994)

[j12] Peter G. Bouillon, Georg Sander, Reinhard Wilhelm: **Lokale Optimierung ausnahmebehafteter Programme durch Spuroptimierung.** Inform., Forsch. Entwickl. 9(2): 72-81 (1994)

[c32] Yosi Ben-Asher, Gudula Rünger, Reinhard Wilhelm, Assaf Schuster: **Implementing 2DT on a Multiprocessor.** CC 1994: 113-127

1993

dblp: Reinhard Wilhelm http://www.informatik.uni-trier.de/~ley/pers/hd/w/Wilhelm:R...

'10 '00 '90 '80 '70

[c31] Volker Claus, Reinhard Wilhelm: **Einleitung.** Perspektiven der Informatik 1993: 9-12

[c30] Reinhard Wilhelm: **Generische und generative Methoden.** Perspektiven der Informatik 1993: 84-85

[c29] Reinhard Wilhelm, Hans Hagen: **Programmiersprachen.** Perspektiven der Informatik 1993: 86-90

[c28] Reinhard Wilhelm, Otto Spaniol: **Parallele und verteilte Systeme.** Perspektiven der Informatik 1993: 90-94

[c27] Yosi Ben-Asher, Gudula Rünger, Assaf Schuster, Reinhard Wilhelm: **2DT-FP: An FP Based Programming Language for Efficient Parallel Programming of Multiprocessor Networks.** PARLE 1993: 42-55

[c26] Martin Alt, Georg Sander, Reinhard Wilhelm: **Generation of Synchronization Code for Parallel Compilers.** PLILP 1993: 420-421

[c25] Martin Alt, Christian Fecht, Christian Ferdinand, Reinhard Wilhelm: **Transformation Development: TrafoLa-H Subsystem.** PROSPECTRA Book 1993: 539-576

1992

[b3] Reinhard Wilhelm, Dieter Maurer: **Übersetzerbau - Theorie, Konstruktion, Generierung.** Springer 1992, ISBN 3-540-55704-0

1991

[j11] Jürgen Börstler, Ulrich Möncke, Reinhard Wilhelm: **Table Compression for Tree Automata.** ACM Trans. Program. Lang. Syst. 13(3): 295-314 (1991)

[c24] Christian Ferdinand, Helmut Seidl, Reinhard Wilhelm: **Tree Automata for Code Selection.** Code Generation 1991: 30-50

[c23] Ulrich Möncke, Reinhard Wilhelm: **Grammar Flow Analysis.** Attribute Grammars, Applications and Systems 1991: 151-186

[c22] Peter Lipps, Ulrich Möncke, Reinhard Wilhelm: **An Overview of the OPTRAN System.** Attribute Grammars, Applications and Systems 1991: 505-506

[c21] Reinhard Wilhelm: **Attributive Reevaluation in OPTRAN.** Attribute Grammars, Applications and Systems 1991: 507

1990

[c20] Shmuel Sagiv, Nissim Francez, Michael Rodeh, Reinhard Wilhelm: **A Logic-Based Approach to Data Flow Analysis Problem.** PLILP

'10 '00 '90 '80 '70

1990: 277-292

[c19] Reinhard Wilhelm: **Tree Tranformations, Functional Languages, and Attribute Grammars.** WAGA 1990: 116-129

[–] 1980 – 1989

1989

[b2] Jacques Loeckx, Kurt Mehlhorn, Reinhard Wilhelm: **Foundations of Programming Languages.** John Wiley 1989, ISBN 0-471-92139-4

[j10] Dieter Maurer, Reinhard Wilhelm: **MaMa - eine abstrakte Maschine zur Implementierung funktionaler Programmiersprachen.** Inform., Forsch. Entwickl. 4(2): 67-88 (1989)

[j9] Winfried Thome, Reinhard Wilhelm: **Simulating Circular Attribute Grammars Through Attribute Reevaluation.** Inf. Process. Lett. 33(2): 79-81 (1989)

[c18] Reinhard Wilhelm: **Übersetzer für imperative, funktionale und logische Programmiersprachen: Ein Vergleich (eingeladener Vortrag).** Software-Entwicklung 1989: 156-165

1988

[j8] Peter Lipps, Ulrich Möncke, Matthias Olk, Reinhard Wilhelm: **Attribute (Re)evaluation in OPTRAN.** Acta Inf. 26(3): 213-239 (1988)

[c17] Martin Raber, Thomas Remmel, Erwin Hoffmann, Dieter Maurer, Fritz Müller, Hans-Georg Oberhauser, Reinhard Wilhelm: **Complied Graph Reduction on a Processor Network.** ARCS 1988: 198-212

[c16] Peter Lipps, Ulrich Möncke, Reinhard Wilhelm: **OPTRAN - A Language/System for the Specification of Program Transformations: System Overview and Experiences.** CC 1988: 52-65

[c15] Beatrix Weisgerber, Reinhard Wilhelm: **Two Tree Pattern Matchers for Code Selection.** CC 1988: 215-229

[c14] M. Baston, H.-J. Bach, A. Lucks-Baus, Fritz Müller, Reinhard Wilhelm: **Implementierung der funktionalen Programmiersprache HOPE mit Hilfe von Kombinatoren.** Innovative Informations-Infrastrukturen 1988: 114-131

'10 '00 '90 '80 '70

[c13] Gerhard Becker, Brigitte Kuhn, Dieter Maurer, Reinhard Wilhelm: **$S_{i}A T_{E}X$ - eine interaktive Arbeitsumgeubng für $T_{E}X$.** Innovative Informations-Infrastrukturen 1988: 162-169

1987

[j7] F. Warren Burton, Dieter Maurer, Hans-Georg Oberhauser, Reinhard Wilhelm: **A Space-Efficient Optimization of Call-by-Need.** IEEE Trans. Software Eng. 13(6): 636-642 (1987)

1986

[b1] Jacques Loeckx, Kurt Mehlhorn, Reinhard Wilhelm: **Grundlagen der Programmiersprachen.** Teubner 1986, ISBN 3-519-02254-0

[e3] Bernard Robinet, Reinhard Wilhelm (Eds.): **ESOP 86, European Symposium on Programming, Saarbrücken, Federal Republic of Germany, March 17-19, 1986, Proceedings.** Lecture Notes in Computer Science 213, Springer 1986, ISBN 3-540-16442-1

1984

[c12] Reinhard Wilhelm: **Inverse Currying Transformation on Attribute Grammars.** POPL 1984: 140-147

[c11] Ulrich Möncke, Beatrix Weisgerber, Reinhard Wilhelm: **How to Implement a System for Manipulation of Attributed Trees.** Fachtagung über Programmiersprachen 1984: 112-127

1983

[j6] Reinhard Wilhelm: **Symbolische Programmausführung - Das aktuelle Schlagwort.** Informatik Spektrum 6(3): 170 (1983)

1982

[j5] Jan Messerschmidt, Reinhard Wilhelm: **Constructors for Composed Objects.** Comput. Lang. 7(2): 53-59 (1982)

[c10] Reinhard Wilhelm: **Imperative, prädikative und funktionale Programmierung (Kurzfassung).** GI Jahrestagung 1982: 188-193

[c9] Reinhard Wilhelm: **LL- and LR-Attributed Grammars.** Fachtagung über Programmiersprachen 1982: 151-164

[c8] Harald Ganzinger, Robert Giegerich, Ulrich Möncke, Reinhard Wilhelm: **A Truly Generative Semantics-Directed Compiler Generator.** SIGPLAN Symposium on Compiler Construction 1982: 172-184

1981

[j4] Reinhard Wilhelm: **A Modified Tree-to-Tree Correction Problem.** Inf. Process. Lett. 12(3): 127-132 (1981)

[c7] Robert Giegerich, Ulrich Möncke, Reinhard Wilhelm: **Invariance of Approximate Semantics with Respect to Program Transformations.** GI Jahrestagung 1981: 1-10

1980

[c6] Ingrid Glasner, Ulrich Möncke, Reinhard Wilhelm: **OPTRAN, a Language for the Specification of Program Transformations.** Fachtagung über Programmiersprachen 1980: 125-142

[e2] Reinhard Wilhelm (Ed.): **CAD-Fachgespräch, GI - 10. Jahrestagung, Saarbrücken, 30. September - 2. Oktober 1980, Proceedings.** Informatik-Fachberichte 34, Springer 1980, ISBN 3-540-10389-9

[e1] Reinhard Wilhelm (Ed.): **GI - 10. Jahrestagung, Saarbrücken, 30. September - 2. Oktober 1980, Proceedings.** Informatik-Fachberichte 33, Springer 1980, ISBN 3-540-10388-0

[–] 1970 – 1979

1979

[j3] Reinhard Wilhelm: **Computation and Use of Data Flow Information in Optimizing Compilers.** Acta Inf. 12: 209-225 (1979)

[j2] Reinhard Wilhelm: **Attributierte Grammatiken.** Informatik Spektrum 2(3): 123-130 (1979)

1978

[j1] Robert Giegerich, Reinhard Wilhelm: **Counter-One-Pass Features in One-Pass Compilation: A Formalization Using Attribute Grammars.** Inf. Process. Lett. 7(6): 279-284 (1978)

1977

[c5] Robert Giegerich, Reinhard Wilhelm: **Implementierbarkeit attributierter Grammatiken.** GI Jahrestagung 1977: 17-36

[c4] Harald Ganzinger, Knut Ripken, Reinhard Wilhelm: **Automatic Generation of Optimizing Multipass Compilers.** IFIP Congress 1977: 535-540

1976

[c3] Reinhard Wilhelm, Knut Ripken, Joachim Ciesinger, Harald Ganzinger, Walter Lahner, R. Nollmann: **Design Evaluation of the Compiler Generating System MUGI.** ICSE 1976: 571-576

1975

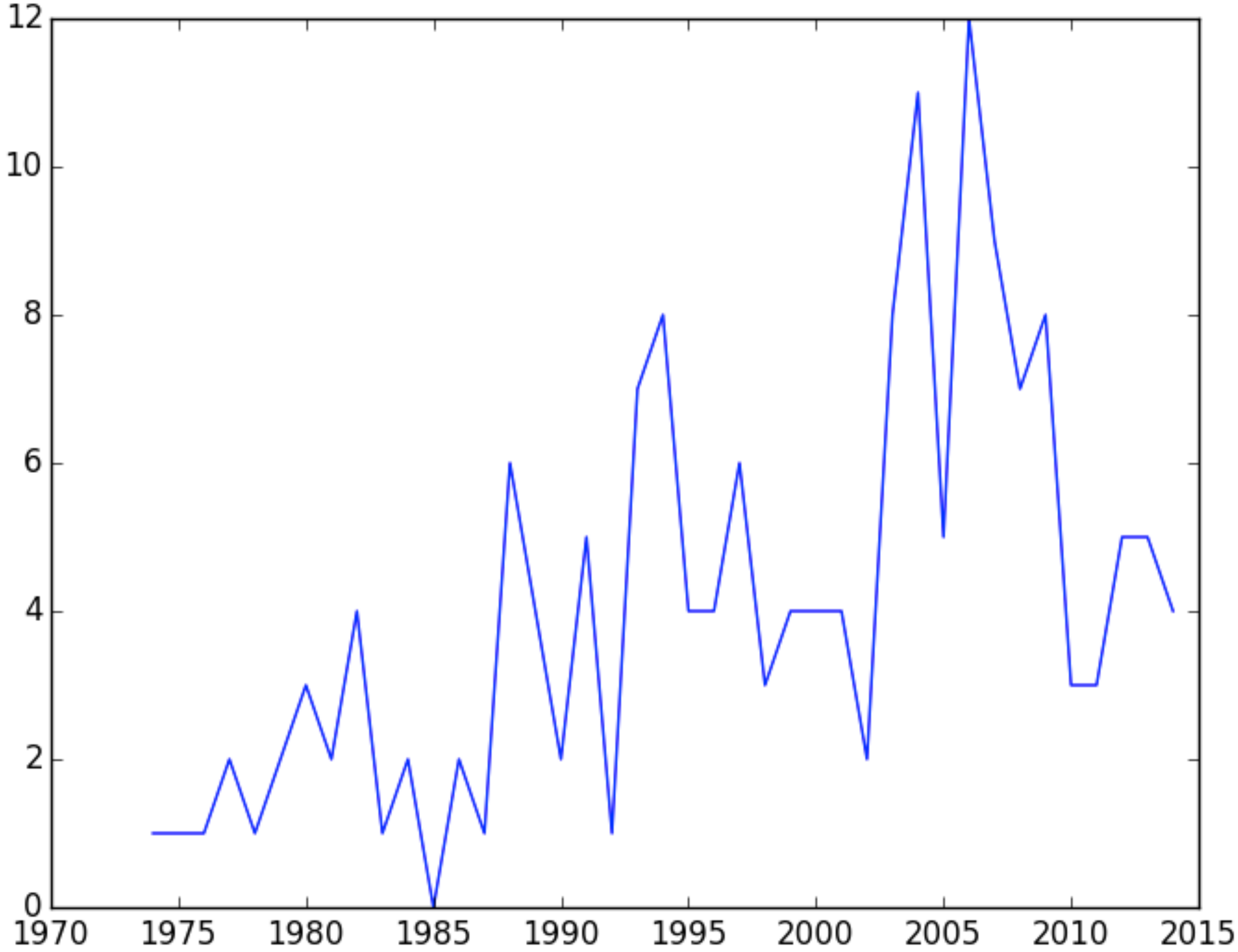
[c2] Harald Ganzinger, Reinhard Wilhelm: **Verschränkung von Compiler-Moduln.** GI Jahrestagung 1975: 654-665

1974

[c1] Reinhard Wilhelm: **Code-Optimierung Mittels Attributierter Transformationsgrammatiken.** GI Jahrestagung 1974: 257-266

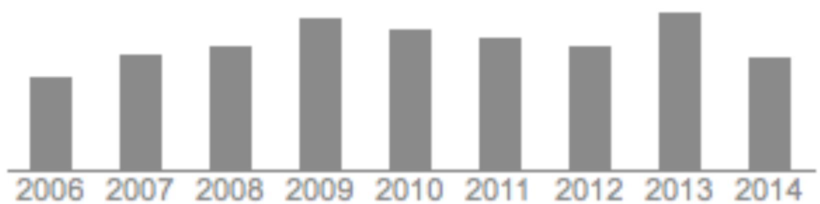
[+] **Coauthor Index**

Great sustainable productivity



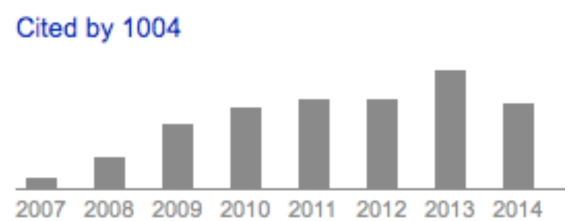
Too much to read so let's have numbers

Citation indices	All	Since 2009
Citations	7053	3497
h-index	35	27
i10-index	75	41



- The hit parade (even cited before published!):

Title 1–20	Cited by	Year
Parametric shape analysis via 3-valued logic M Sagiv, T Reps, R Wilhelm ACM Transactions on Programming Languages and Systems (TOPLAS) 24 (3), 217-298	1047	2002
The worst-case execution-time problem—overview of methods and survey of tools R Wilhelm, J Engblom, A Ermedahl, N Holsti, S Thesing, D Whalley, ... ACM Transactions on Embedded Computing Systems (TECS) 7 (3), 36	1004	2008
Solving shape-analysis problems in languages with destructive updating M Sagiv, T Reps, R Wilhelm ACM Transactions on Programming Languages and Systems (TOPLAS) 20 (1), 1-50	470	1998



- The flop:

An abstract machine for an object-oriented language with top-level classes C Böschen, C Fecht, AV Hense, R Wilhelm	2	2011

Total citations Cited by 2

- The flop:

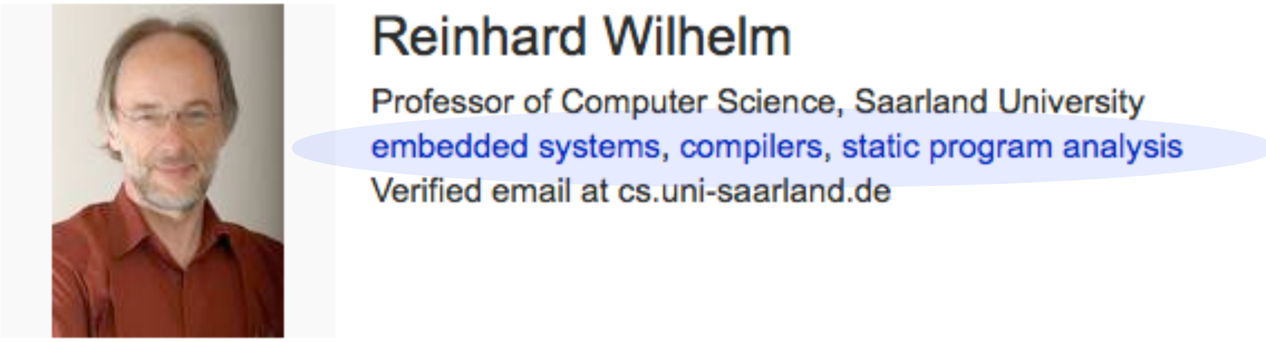
An abstract machine for an object-oriented language with top-level classes C Böschen, C Fecht, AV Hense, R Wilhelm	2	2011

Total citations Cited by 2

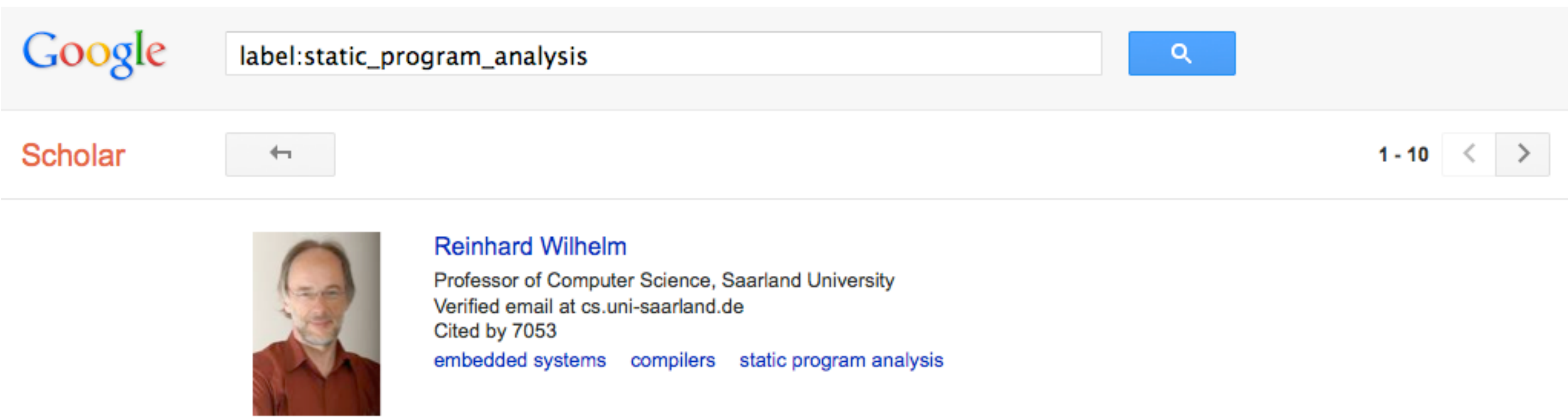
yes, but cited 11 years before the pretend publication date!

Science

Main contributions



- Coming number one in static analysis, world-wide:



What is static analysis?

Static program analysis

From Wikipedia, the free encyclopedia

Static program analysis is the analysis of computer software that is performed without actually executing programs (analysis performed on executing programs is known as dynamic analysis).

at least the static analyzer must execute!

by a computer

A short introduction to static analysis

The very first static analysis

Brahmagupta (Sanskrit: ब्रह्मगुप्त; listen (help·info)) (598–c.670 CE) was an Indian mathematician and astronomer who wrote two important works on Mathematics and Astronomy: the *Brāhmasphuṭasiddhānta* (Extensive Treatise of Brahma) (628), a theoretical treatise, and the *Khaṇḍakhādyaka*, a more practical text.

Brahmagupta

Born	598 CE
Died	c.670 CE
Fields	Mathematics, Astronomy
Known for	Zero, modern Number system

The rule of signs by Brahmagupta (628)

18.30. [The sum] of two positives is positives, of two negatives negative;

The rule of signs by Brahmagupta (628)

18.30. [The sum] of two positives is positives, of two negatives negative;

- The abstraction is that you do not (always) need to known the absolute value of the arguments to know the sign of the result;

The rule of signs by Brahmagupta (628)

18.30. [The sum] of two positives is positives, of two negatives negative;

- The abstraction is that you do not (always) need to known the absolute value of the arguments to know the sign of the result;
- Sometimes imprecise (don't know the sign of the sum of a positive and a negative)

The rule of signs by Brahmagupta (628)

18.30. [The sum] of two positives is positives, of two negatives negative;

- The abstraction is that you do not (always) need to known the absolute value of the arguments to know the sign of the result;
- Sometimes imprecise (don’t know the sign of the sum of a positive and a negative)
- Useful in practice (if you know what to do when you don’t know the sign)

The rule of signs by Brahmagupta (628)

18.30. [The sum] of two positives is positives, of two negatives negative;

- The abstraction is that you do not (always) need to known the absolute value of the arguments to know the sign of the result;
- Sometimes imprecise (don't know the sign of the sum of a positive and a negative)
- Useful in practice (if you know what to do when you don't know the sign)
- e.g. in compilation: do not optimize (a division by 2 into a shift when positive(*))

(*) Unless processor uses 2's complement and can shift the sign.

The rule of signs by Brahmagupta (628)

18.30. [The sum] of two positives is positives, of two negatives negative; [...]

18.32. A negative minus zero is negative, a positive [minus zero] positive; zero [minus zero] is zero. When a positive is to be subtracted from a negative or a negative from a positive, then it is to be added.

The rule of signs by Brahmagupta (628)

18.30. [The sum] of two positives is positives, of two negatives negative; [...]

18.32. A negative minus zero is negative, a positive [minus zero] positive; zero [minus zero] is zero. When a positive is to be subtracted from a negative or a negative from a positive, then it is to be added.

18.33. The product of a negative and a positive is negative, of two negatives positive, and of positives positive; the product of zero and a negative, of zero and a positive, or of two zeros is zero.

The rule of signs by Brahmagupta (628)

18.30. [The sum] of two positives is positives, of two negatives negative; [...]

18.32. A negative minus zero is negative, a positive [minus zero] positive; zero [minus zero] is zero. When a positive is to be subtracted from a negative or a negative from a positive, then it is to be added.

18.33. The product of a negative and a positive is negative, of two negatives positive, and of positives positive; the product of zero and a negative, of zero and a positive, or of two zeros is zero.

18.34. A positive divided by a positive or a negative divided by a negative is positive; a zero divided by a zero is zero; a positive divided by a negative is negative; a negative divided by a positive is [also] negative.

wrong

The rule of signs by Michel Sintzoff (1972)

For example, a×a+b×b yields the value 25 when a is 3 and b is −4, and when + and × are the arithmetic multiplication and addition. But a×a+b×b yields always the object "pos" when a and b are the objects "pos" or "neg", and when the valuation is defined as follows :

```
pos+pos=pos                    pos×pos=pos
pos+neg=pos,neg                pos×neg=neg
neg+pos=pos,neg                neg×pos=neg
neg+neg=neg                    neg×neg=pos
V(p+q)=V(p)+V(q)               V(p×q)=V(p)×V(q)
V(0)=V(1)=...=pos
V(-1)=V(-2)=...=neg
```

The valuation of a×a+b×b yields "pos" by the following computations :

```
V(a)=pos,neg                V(b)=pos,neg
V(a×a)=pos×pos,neg×neg      V(b×b)=pos×pos,neg×neg
      =pos,pos=pos                =pos,pos=pos
V(a×a+b×b)=V(a×a)+V(b×b)=pos+pos=pos
```

This valuation proves that the result of a×a+b×b is always positive and hence allows to compute its square root without any preliminary dynamic test on its sign. On the other hand, the

The rule of signs by Michel Sintzoff (1972)

For example, a×a+b×b yields the value 25 when a is 3 and b is −4, and when + and × are the arithmetic multiplication and addition. But a×a+b×b yields always the object "pos" when a and b are the objects "pos" or "neg", and when the valuation is defined as follows :

```
pos+pos=pos                  pos×pos=pos
pos+neg=pos,neg              pos×neg=neg
neg+pos=pos,neg              neg×pos=neg
neg+neg=neg                  neg×neg=pos
V(p+q)=V(p)+V(q)             V(p×q)=V(p)×V(q)
V(0)=V(1)=...=pos
V(-1)=V(-2)=...=neg
```

The valuation of a×a+b×b yields "pos" by the following computations :

```
V(a)=pos,neg                 V(b)=pos,neg
V(a×a)=pos×pos,neg×neg       V(b×b)=pos×pos,neg×neg
      =pos,pos=pos                 =pos,pos=pos
V(a×a+b×b)=V(a×a)+V(b×b)=pos+pos=pos
```

This valuation proves that the result of a×a+b×b is always positive and hence allows to compute its square root without any preliminary dynamic test on its sign. On the other hand, the

The rule of signs by Michel Sintzoff (1972)

For example, a×a+b×b yields the value 25 when a is 3 and b is −4, and when + and × are the arithmetic multiplication and addition. But a×a+b×b yields always the object "pos" when a and b are the objects "pos" or "neg", and when the valuation is defined as follows :

pos+pos=pos	pos×pos=pos
pos+neg=pos,neg	pos×neg=neg
neg+pos=pos,neg	neg×pos=neg
neg+neg=neg	neg×neg=pos
V(p+q)=V(p)+V(q)	V(p×q)=V(p)×V(q)
V(0)=V(1)=...=pos	
V(−1)=V(−2)=...=neg	

wrong

$0 \in pos \times -1 \in neg$

$= 0 \notin neg$

The valuation of a×a+b×b yields "pos" by the following computations :

V(a)=pos,neg	V(b)=pos,neg
V(a×a)=pos×pos,neg×neg	V(b×b)=pos×pos,neg×neg
=pos,pos=pos	=pos,pos=pos

V(a×a+b×b)=V(a×a)+V(b×b)=pos+pos=pos

This valuation proves that the result of a×a+b×b is always positive and hence allows to compute its square root without any preliminary dynamic test on its sign. On the other hand, the

The rule of signs by Reinhard Wilhelm (2012/13)

http://www.rw.cdl.uni-saarland.de/teaching/dses12/slides/lecture6_static_analysis.pdf

2 Example — Rules-of-Sign Analysis

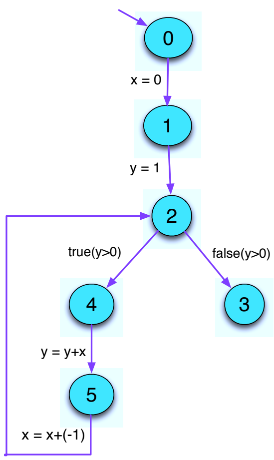
Problem: Determine at each program point the sign of the values of all variables of numeric type.

Example program:

```
1: x = 0;
2: y = 1;
3: while (y > 0) do
4:     y = y + x;
5:     x = x + (-1);
```

Program representation as *control-flow graphs*

8

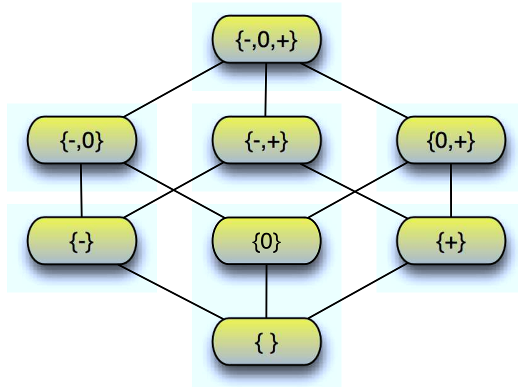


9

All the ingredients:

- a set of information elements, each a set of possible signs,
- a partial order, "$\sqsubseteq$", on these elements, specifying the "relative strength" of two information elements,
- these together form the abstract domain, a lattice,
- functions describing how signs of variables change by the execution of a statement, abstract edge effects,
- these need an abstract arithmetic, an arithmetic on signs.

We construct the abstract domain for single variables starting with the lattice $\mathit{Signs} = 2^{\{-,0,+\}}$ with the relation "$\sqsubseteq$" ="$\subseteq$".



The analysis should "bind" program variables to elements in Signs.

So, the abstract domain is $\mathbb{D} = (\mathit{Vars} \to \mathit{Signs})_\perp$, a Sign-environment.

$\perp \in \mathbb{D}$ is the function mapping all arguments to $\{\}$.

The partial order on $\mathbb{D}$ is $D_1 \sqsubseteq D_2$ iff

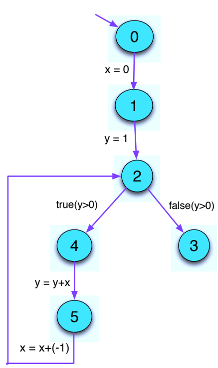
$D_1 = \perp$ or

$D_1\, x \subseteq D_2\, x \quad (x \in \mathit{Vars})$

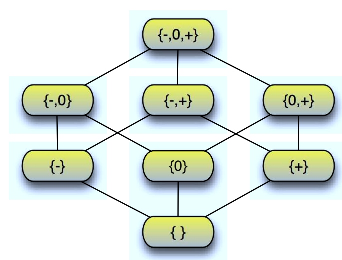
Intuition?

13

How did we analyze the program?



In particular, how did we walk the lattice for y at program point 5?



16

The analysis should "bind" program variables to elements in Signs.

So, the abstract domain is $\mathbb{D} = (\mathit{Vars} \to \mathit{Signs})_\perp$. a Sign-environment.

$\perp \in \mathbb{D}$ is the function mapping all arguments to $\{\}$.

The partial order on $\mathbb{D}$ is $D_1 \sqsubseteq D_2$ iff

$D_1 = \perp$ or

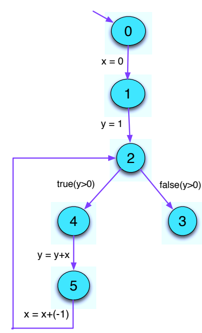
$D_1\, x \subseteq D_2\, x \quad (x \in \mathit{Vars})$

Intuition?

D_1 is at least as precise as D_2 since D_2 admits at least as many signs as D_1

How is a solution found?

Iterating until a fixed-point is reached



15

0		1		2		3		4		5	
x	y	x	y	x	y	x	y	x	y	x	y

17

Idea:

- We want to determine the sign of the values of expressions.
- For some sub-expressions, the analysis may yield $\{+, -, 0\}$ which means, it couldn't find out.

Idea:

- We want to determine the signs of the values of expressions.
- For some sub-expressions, the analysis may yield $\{+, -, 0\}$, which means, it couldn't find out.
- We replace the concrete operators $\Box$ working on values by abstract operators $\Box^\sharp$ working on signs:

The rule of signs by Reinhard Wilhelm (2012/13)

Idea:

- We want to determine the signs of the values of expressions.
- For some sub-expressions, the analysis may yield $\{+,-,0\}$, which means, it couldn't find out.
- We replace the concrete operators $\Box$ working on values by abstract operators $\Box^\sharp$ working on signs:
- The abstract operators allow to define an abstract evaluation of expressions:

$$[\![e]\!]^\sharp \;:\; (Vars \to Signs) \to Signs$$

Determining the sign of expressions in a Sign-environment works as follows:

$$\begin{array}{lcl}
[\![c]\!]^\sharp\, D & = & \begin{cases} \{+\} & \text{if } c > 0 \\ \{-\} & \text{if } c < 0 \\ \{0\} & \text{if } c = 0 \end{cases} \\
[\![v]\!]^\sharp & = & D(v) \\
[\![e_1 \Box e_2]\!]^\sharp\, D & = & [\![e_1]\!]^\sharp\, D \,\Box^\sharp\, [\![e_2]\!]^\sharp\, D \\
[\![\Box e]\!]^\sharp\, D & = & \Box^\sharp [\![e]\!]^\sharp\, D
\end{array}$$

Abstract operators working on signs (Addition)

$+^\#$	{0}	{+}	{-}	{-, 0}	{-, +}	{0, +}	{-, 0, +}
{0}	{0}	{+}					
{+}							
{-}							
{-, 0}							
{-, +}							
{0, +}							
{-, 0, +}	{-, 0, +}						

22

23

Abstract operators working on signs (Multiplication)

$\times^\#$	{0}	{+}	{-}	{-, 0}	{-, +}	{0, +}	{-, 0, +}
{0}	{0}	{0}					
{+}							
{-}							
{-, 0}							
{-, +}							
{0, +}							
{-, 0, +}	{0}						

Abstract operators working on signs (unary minus)

$-^\#$	{0}	{+}	{-}	{-, 0}	{-, +}	{0, +}	{-, 0, +}
	{0}	{-}	{+}	{+, 0}	{-, +}	{0, -}	{-, 0, +}

Working an example: $D = \{x \mapsto \{+\}, y \mapsto \{+\}\}$

$$\begin{array}{lcl}
[\![x+7]\!]^\sharp\, D & = & [\![x]\!]^\sharp\, D +^\sharp [\![7]\!]^\sharp\, D \\
& = & \{+\} +^\sharp \{+\} \\
& = & \{+\} \\
[\![x+(-y)]\!]^\sharp\, D & = & \{+\} +^\sharp (-^\sharp [\![y]\!]^\sharp\, D) \\
& = & \{+\} +^\sharp (-^\sharp \{+\}) \\
& = & \{+\} +^\sharp \{-\} \\
& = & \{+,-,0\}
\end{array}$$

Thus, we obtain the following effects of edges $[\![lab]\!]^\sharp$:

$$\begin{array}{lcl}
[\![;]\!]^\sharp\, D & = & D \\
[\![\mathrm{true}\,(e)]\!]^\sharp\, D & = & D \\
[\![\mathrm{false}\,(e)]\!]^\sharp\, D & = & D \\
[\![x = e;]\!]^\sharp\, D & = & D \oplus \{x \mapsto [\![e]\!]^\sharp\, D\} \\
[\![x = M[e];]\!]^\sharp\, D & = & D \oplus \{x \mapsto \{+,-,0\}\} \\
[\![M[e_1] = e_2;]\!]^\sharp\, D & = & D
\end{array}$$

... whenever $D \neq \bot$

Attention to details

The rule of signs by Reinhard Wilhelm (2012/13)

Idea:

- We want to determine the signs of the values of expressions.
- For some sub-expressions, the analysis may yield $\{+,-,0\}$, which means, it couldn't find out.
- We replace the concrete operators $\Box$ working on values by abstract operators $\Box^\sharp$ working on signs:
- The abstract operators allow to define an abstract evaluation of expressions:

$$[\![e]\!]^\sharp \;:\; (\mathit{Vars} \to \mathit{Signs}) \to \mathit{Signs}$$

Determining the sign of expressions in a Sign-environment works as follows:

$$\begin{array}{lcl}
[\![c]\!]^\sharp\, D & = & \begin{cases} \{+\} & \text{if } c > 0 \\ \{-\} & \text{if } c < 0 \\ \{0\} & \text{if } c = 0 \end{cases} \\
[\![v]\!]^\sharp & = & D(v) \\
[\![e_1 \Box e_2]\!]^\sharp\, D & = & [\![e_1]\!]^\sharp\, D \,\Box^\sharp\, [\![e_2]\!]^\sharp\, D \\
[\![\Box e]\!]^\sharp\, D & = & \Box^\sharp [\![e]\!]^\sharp\, D
\end{array}$$

Abstract operators working on signs (Addition)

$+^\#$	{0}	{+}	{-}	{-, 0}	{-, +}	{0, +}	{-, 0, +}
{0}	{0}	{+}					
{+}							
{-}							
{-, 0}							
{-, +}							
{0, +}							
{-, 0, +}	{-, 0, +}						

22

23

Abstract operators working on signs (Multiplication)

$\times^\#$	{0}	{+}	{-}	{-, 0}	{-, +}	{0, +}	{-, 0, +}
{0}	{0}	{0}					
{+}							
{-}							
{-, 0}							
{-, +}							
{0, +}							
{-, 0, +}	{0}						

Abstract operators working on signs (unary minus)

$-^\#$	{0}	{+}	{-}	{-, 0}	{-, +}	{0, +}	{-, 0, +}
	{0}	{-}	{+}	{+, 0}	{-, +}	{0, -}	{-, 0, +}

Working an example: $D = \{x \mapsto \{+\}, y \mapsto \{+\}\}$

$$\begin{array}{lcl}
[\![x+7]\!]^\sharp\, D & = & [\![x]\!]^\sharp\, D +^\sharp [\![7]\!]^\sharp\, D \\
& = & \{+\} +^\sharp \{+\} \\
& = & \{+\} \\
[\![x+(-y)]\!]^\sharp\, D & = & \{+\} +^\sharp (-^\sharp [\![y]\!]^\sharp\, D) \\
& = & \{+\} +^\sharp (-^\sharp \{+\}) \\
& = & \{+\} +^\sharp \{-\} \\
& = & \{+,-,0\}
\end{array}$$

Thus, we obtain the following effects of edges $[\![lab]\!]^\sharp$:

$$\begin{array}{lcl}
[\![;]\!]^\sharp\, D & = & D \\
[\![\mathrm{true}\,(e)]\!]^\sharp\, D & = & D \\
[\![\mathrm{false}\,(e)]\!]^\sharp\, D & = & D \\
[\![x = e;]\!]^\sharp\, D & = & D \oplus \{x \mapsto [\![e]\!]^\sharp\, D\} \\
[\![x = M[e];]\!]^\sharp\, D & = & D \oplus \{x \mapsto \{+,-,0\}\} \\
[\![M[e_1] = e_2;]\!]^\sharp\, D & = & D
\end{array}$$

... whenever $D \neq \bot$

if the program does not terminate isn't it correct to say that x is 0 upon its termination?

Attention to details

That’s where you recognize a great scientist: make simple what is complicated!

Suggestions for an happy retirement

Have ambitious objectives!

Have ambitious objectives!

- Move Dagstuhl close to an airport (or an airport close to Dagstuhl)

About Dagstuhl | Program | Publications

You are here: About Dagstuhl » Arrival » Arrival by Plane

Arrival by Plane

The nearest airports

Airports	IATA-Code	Flights	Transfer to Dagstuhl	Distance from Dagstuhl	Driving time to Dagstuhl
Frankfurt am Main	FRA	International	Train	190 km	120 min
Frankfurt-Hahn	HHN	European economy carriers	Bus, taxi, rental car	70 km	80 min
Saarbrücken	SCN	Berlin Hamburg Luxemburg	Train, then bus or taxi	80 km	50 min
Luxemburg	LUX	Europe	Taxi direct or train via Trier	80 km	50 min

Ground transportation from the airports

At **Frankfurt Main Airport** the "Frankfurt-Flughafen-Regiobahnhof" (Frankfurt Airport Regional Train Station) is located in the basement (level 0) of concourse B (terminal 1). Take the regional express (RE) from there to St. Wendel, or take the "S-Bahn" (urban train) to Mainz and then change to the regional express. The trains leave every hour. No advance booking needed. For further travel to St. Wendel, see Arrival By Train.
How to get from Frankfurt Airport to Dagstuhl by car can be found here.

Upon arriving at **Frankfurt-Hahn Airport** the best solution is taking a Taxi. Reserving a Rental Car is also suitable. If choosing Bus and train please be aware that it will take about 5 hours to arrive at Dagstuhl (for information on the train connections follow the link and enter "HHN Flughafen Hahn" as your departure station and "Dagstuhl Bahnhofstr., Wadern" as your arrival station).

Upon arriving at **Saarbrücken Airport** take the bus (Line R10) or a taxi to the Saarbrücken main train station, then continue to St. Wendel by train, bus or local taxi. (see Arrival by Train)
Alternative: Order Taxi Martin service from Wadern.

Upon arriving at **Luxembourg Airport** order Taxi Martin service from Wadern.
Alternative: Take a Rental Car or come by bus & train via Luxemburg station, Trier and Merzig. (see Arrival by Train)

Remain active in science!

Remain active in science!

- Start working on cyberimbedded systems

Remain active in science!

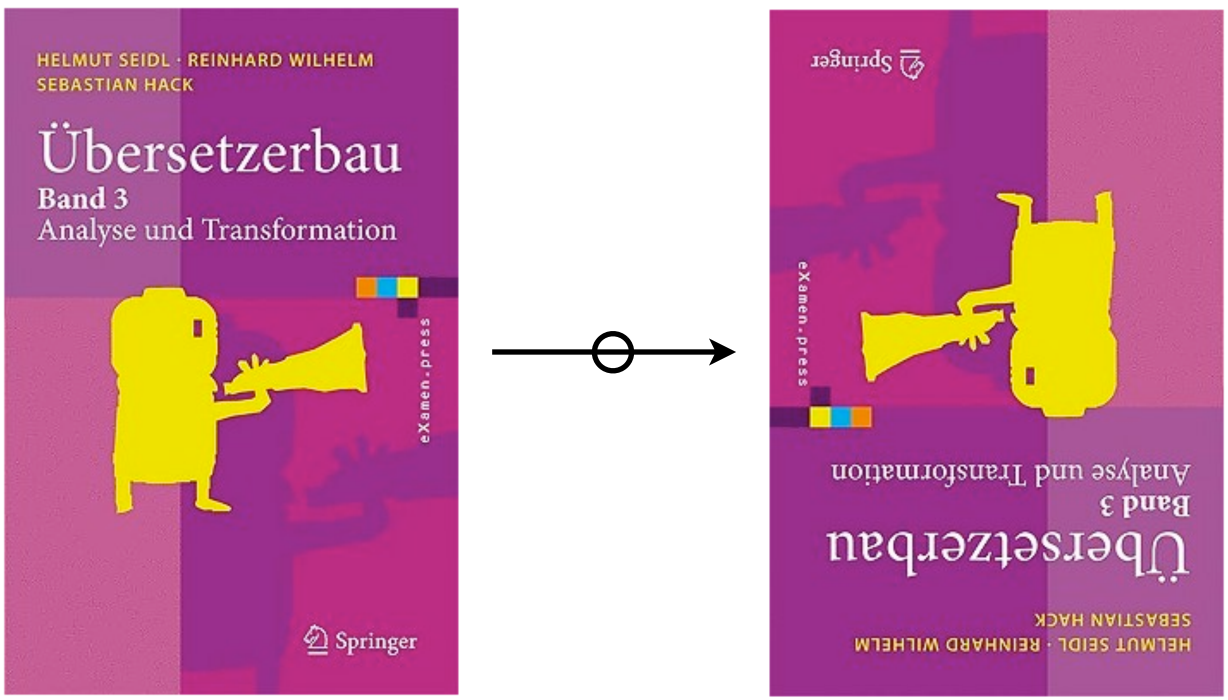
- Start working on cyberimbedded systems
- Consider dynamic methods for static analysis

Remain active in science!

- Start working on cyberimbedded systems
- Consider dynamic methods for static analysis
- Write a book on decompilation

Remain active in science!

- Start working on cyberimbedded systems
- Consider dynamic methods for static analysis
- Write a book on decompilation, by duality



Time for a serious conclusion

Thanks a lot for 30 years of friendship

Thanks a lot for 30 years of friendship, with lots of problems!

Thanks a lot for 30 years of friendship, with lots of problems!

shared scientific

The End, thank you

The beginning, thank you

of retirement

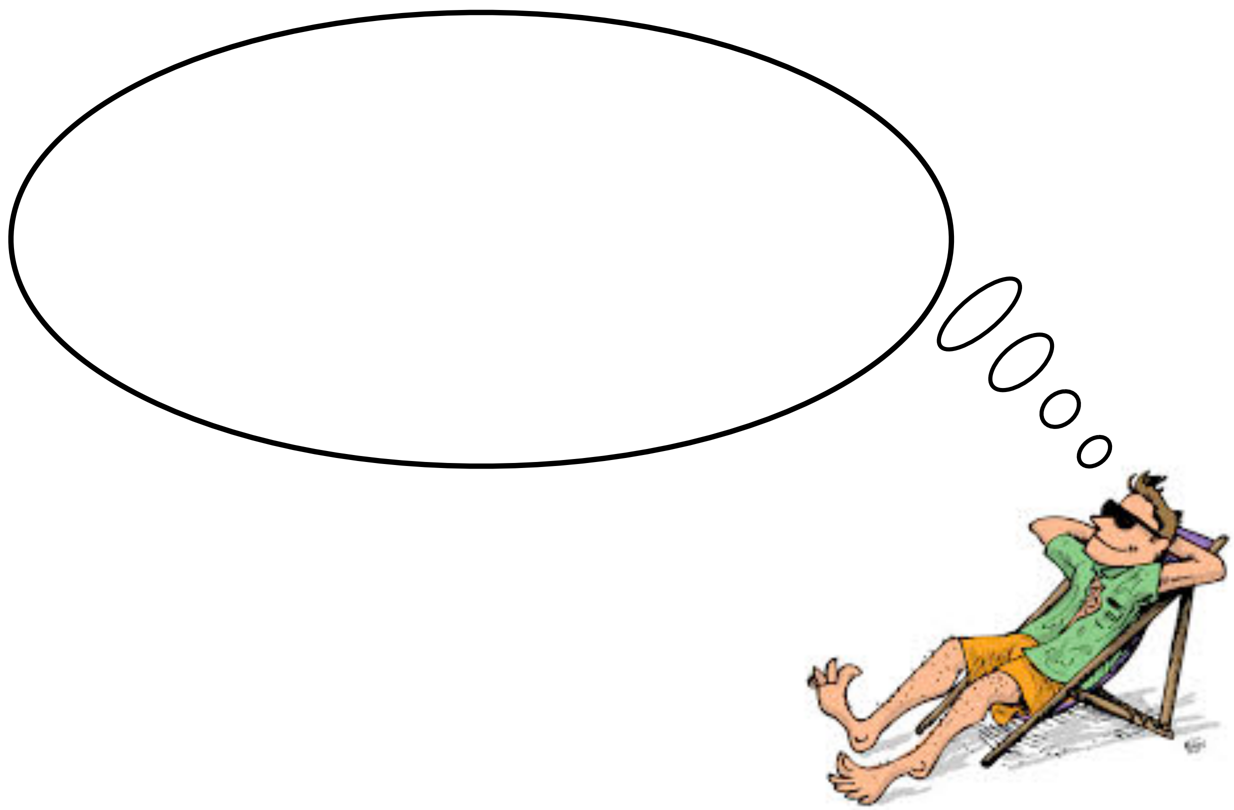
The beginning, thank you

of retirement

The beginning, thank you

of retirement

The beginning, thank you



of retirement

The beginning, thank you