February 2024

JOHN REGEHR

School of Computing
50 South Central Campus Drive, Rm 3190 regehr@cs.utah.edu
University of Utah http://www.cs.utah.edu/~regehr
Salt Lake City, UT 84112–9205

Education

PhD, Computer Science, University of Virginia. Charlottesville, VA. May 2001.

Masters of Computer Science, University of Virginia. Charlottesville, VA. May 1997.

BS, Computer Science, Kansas State University. Manhattan, KS. May 1995.

BS, Mathematics, Kansas State University. Manhattan, KS. May 1995.

Positions

Associate Director, Kahlert School of Computing, University of Utah, July 2020–June 2022, July 2023–present.

Professor, Kahlert School of Computing, University of Utah, July 2015–present.

Associate Professor, School of Computing, University of Utah, July 2009–June 2015.

Assistant Professor, School of Computing, University of Utah, August 2003–June 2009.

Postdoctoral Fellow, School of Computing, University of Utah. April 2001–July 2003.

Research Grants

National Science Foundation Software and Hardware Foundations Program. "Formal Methods as a First-Class Citizen of a Mainstream Compiler Framework." PI John Regehr, co-PI Zvonimir Rakamaric. Award CCF–1955688. June 2020–May 2024. $1,114,703

Intel. "Feedback-Driven Compiler Test-Case Generation." PI John Regehr. Two grants: one in 2018 and one in 2020, totaling $150,000.

Office of Naval Research TPCP program. "Slimming, Simplifying and Securing Software Systems." PI Vikram Adve (UIUC), John Regehr (Utah), John Criswell (Rochester). 2017–2020. Budget around $1,000,000.

National Science Foundation Software and Hardware Foundations Program. "Xsmith, A Configurable Generator of Highly Effective Fuzz Testers." Award CCF–1527638. PI Eric Eide, co-PI John Regehr. September 2015–August 2018. $499.998.

National Science Foundation Secure and Trustworthy Cyberspace Program. "XCap: Practical Capabilities and Least Authority for Virtualized Environments." Award CNS–1319076. PI John Regehr, co-PI Anton Burtsev. October 2013–September 2016. $499,912.

Google Faculty Research Award. "Disaggregated System Services Through Lightweight Capability Domains." PI John Regehr. August 2013. $40,600.

National Science Foundation Computer Systems Research Program. "Beating Implementations of C++11 Concurrency Into Shape." Award CNS–1218022. PI John Regehr. September 2012–August 2015. $467,740.

National Science Foundation Software and Hardware Foundations Program. "Diversity and Feedback in Random Testing for Systems Software." Award CCF–1218026. PIs John Regehr and Alex Groce (Oregon State). September 2012–August 2015. Utah award $249,036.

Google Faculty Research Award. "Automated Compiler Bug Reports." PI John Regehr. December 2011. $35,061.

DARPA CRASH program. "Containers for Advanced Adaptive Applications" PI John Regehr, co-PI Eric Eide, PI Partha Pal at BBN. September 2010–August 2014. Approximately $1,400,000.

DARPA Computer Science Study Group program. "A Cross-Layer Approach for Improving Embedded Software Reliability." PI John Regehr. May 2009–April 2011. $474,340.

DARPA Computer Science Study Group program. PI John Regehr. March 17 2008–March 16 2009. $95,989.

National Science Foundation Embedded and Hybrid Systems Program. "Improving Sensor Network Software Reliability through Language, Tool, and OS Co-Design." Award CNS–0615367. PIs John Regehr and Philip Levis (Stanford), co-PI Dawson Engler (Stanford). September 2006–August 2009. Utah award: $210,000, Stanford award $360,000.

National Science Foundation Parallel and Distributed Operation Systems Program. "Experimenting with Garbage Collection in an Otherwise Conventional OS." Award CNS–0509526. PI Matthew Flatt, co-PI John Regehr. June 2005–April 2008. $380,000.

National Science Foundation CAREER Program. "Vertically Integrated Analysis for Embedded Software." Award CNS–0448047. May 2005–April 2010. $400,000.

National Science Foundation Embedded and Hybrid Systems Program. "Components and Aspects for Embedded Middleware." Award CNS–0410285. PIs Matthew Flatt and Raymond Klefstad (UC Irvine), co-PIs Eric Eide and John Regehr. September 2004–August 2007. Utah award: $360,002, UC Irvine award $330,000.

National Science Foundation Embedded and Hybrid Systems Program. "Composable Execution Environments: A Foundation for Building Robust Embedded Systems." Award CCR–0209185. PI Jay Lepreau, co-PI John Regehr. July 2002–June 2005. $310,000.

Teaching

CS 1810, Intro to Computer Systems, Spring 2024. 48 students enrolled. 3 credit hours. Co-instructor Erin Parker.

CS 4500, Senior Capstone Project, Fall 2023. 70 students enrolled. 3 credit hours. Co-instructor J. Davison De St Germain.

CS 4500, Senior Capstone Project, Spring 2022. 98 students enrolled. 3 credit hours. Co-instructor H. James De St. Germain.

CS 6465, Advanced Operating Systems, Fall 2021. 25 students enrolled. 3 credit hours.

CS 4470, Compilers, Spring 2021. 25 students enrolled. 3 credit hours. Co-instructor Pavel Panchekha.

CS 1030, Foundations of Computer Science, Fall 2020. 222 students enrolled. 3 credit hours. Co-instructor Scott Brown.

CS 1030, Foundations of Computer Science, Fall 2019. 178 students enrolled. 3 credit hours.

CS 6015, Software Engineering (MSD program), Spring 2019. 24 students enrolled. 3 credit hours.

CS 6960, Advanced Compilers, Fall 2018. 9 students enrolled. 3 credit hours.

CS 6015, Software Engineering (MSD program), Spring 2018. 19 students enrolled. 3 credit hours.

CS 6960, Advanced Operating Systems, Fall 2017. 16 students enrolled, 3 credit hours.

CS 5460/6460, Operating Systems, Spring 2017. 112 students enrolled. 4 credit hours.

CS 6960, Advanced Compilers, Fall 2016. 18 students enrolled. 3 credit hours.

CS 5460/6460, Operating Systems, Spring 2015. 106 students enrolled. 4 credit hours.

CS/ECE 5785/6785, Advanced Embedded Software, Fall 2014. 38 students enrolled. 3 credit hours.

CS 5959, Writing Solid Code, Spring 2014. 21 students enrolled. 3 credit hours. I received the School of Computing Outstanding Teaching Award for this course.

CS 5962, Advanced Operating Systems, Spring 2014. 7 students enrolled. 3 credit hours.

CS 4400, Computer Systems, Fall 2013. 157 students enrolled. 4 credit hours.

CS 5460/6460, Operating Systems, Spring 2013. 55 students enrolled. 4 credit hours.

CS/ECE 5785/6785, Advanced Embedded Software, Fall 2012. 28 students enrolled. 3 credit hours.

CS 5460/6460, Operating Systems, Spring 2012. 77 students enrolled. 4 credit hours.

CS 7942, Seminar on System Support for Data Centers, Spring 2012, 3 students enrolled. 1–3 credit hours.

CS 5957, Android Projects, Fall 2011. 13 students enrolled. 3 credit hours.

CS 5460/6460, Operating Systems, Fall 2010. 56 students enrolled. 4 credit hours.

CS/ECE 5785/6785, Advanced Embedded Systems, Fall 2010. 29 students enrolled. 3 credit hours.

CS/ECE 5780/6780, Embedded Systems, Spring 2009. 50 students enrolled. 3 credit hours.

CS/ECE 5785/6785, Advanced Embedded Systems, Fall 2008. 20 students enrolled. 3 credit hours.

CS 6470, Advanced Compilers, Spring 2008. 7 students enrolled. 3 credit hours.

CS/ECE 5785/6785, Advanced Embedded Systems, Fall 2007. 26 students enrolled. 3 credit hours.

CS 7933, Seminar on Ultra Large Scale Systems, Fall 2007. 5 students enrolled. 1 credit hour.

CS/ECE 5785/6785, Advanced Embedded Systems, Fall 2006. 27 students enrolled. 3 credit hours.

CS 7962, Embedded Systems, Spring 2006. 17 students enrolled. 3 credit hours.

CS 3400, Computer Systems, Fall 2005. 90 students enrolled. 4 credit hours.

CS 7938, Seminar on Program Analysis, Fall 2005. 11 students enrolled. 1 credit hour.

CS 7962, Embedded Systems, Spring 2005. 15 students enrolled. 3 credit hours.

CS 3400, Computer Systems, Fall 2004. 72 students enrolled. 4 credit hours.

CS 7940, Seminar on Sensor Networks, Fall 2004. 1–3 credit hours. Co-instructor Sneha Kasera.

CS 4400, Computer Systems, Spring 2004. 107 students enrolled. 3 credit hours.

CS 6935, Seminar on Embedded and Networked Systems, Fall 2003. 1–3 credit hours. Co-instructor Sneha Kasera.

CS 5460, Operating Systems, Fall 2002. 75 students enrolled. 3 credit hours.

Conference and Journal Publications

High-Throughput, Formal-Methods-Assisted Fuzzing for LLVM.
Yuyou Fan and John Regehr.
In Proceedings of the International Symposium on Code Generation and Optimization (CGO), March 2024.
https://users.cs.utah.edu/~regehr/papers/cgo24.pdf

Fuzzing Loop Optimizations in Compilers for C++ and Data-Parallel Languages.
Vsevolod Livinskii, Dmitry Babokin, and John Regehr.
In Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2023), June 2023.
https://users.cs.utah.edu/~regehr/pldi23.pdf

Alive2: Bounded Translation Validation for LLVM.
Nuno P. Lopes, Juneyoung Lee, Chung-Kil Hur, Zhengyang Liu John, Regehr.
In Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2021), online, June 2021.
This paper received the Distinguished Paper Award.
https://www.cs.utah.edu/~regehr/alive2-pldi21.pdf

Dataflow-based Pruning for Speeding up Superoptimization.
Manasij Mukherjee, Pranav Kant, Zhengyang Liu, John Regehr.
In Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA) 2020, November 2020.
This paper received the Distinguished Paper Award.
https://www.cs.utah.edu/~regehr/dataflow-pruning.pdf

Random Testing for C and C++ Compilers with YARPGen.
Vsevolod Livinskii, Dmitry Babokin, and John Regehr.
In Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA) 2020, November 2020.
This paper received the Distinguished Paper Award.
https://www.cs.utah.edu/~regehr/yarpgen-oopsla20.pdf

Testing Static Analyses for Precision and Soundness.
Jubi Taneja, Zhengyang Liu, and John Regehr.
In Proceedings of the International Symposium on Code Generation and Optimization (CGO) 2020, San Diego, CA, USA, February 2020.
This paper received the Best Paper and Best Student Presentation Awards.
http://www.cs.utah.edu/~regehr/cgo20.pdf

Reconciling High-level Optimizations and Low-level Code in LLVM.
Juneyoung Lee, Chung-Kil Hur, Ralf Jung, Zhengyang Liu, John Regehr, and Nuno P. Lopes.
In Proceedings of the 2018 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Boston, MA, USA, November 2018.
http://www.cs.utah.edu/~regehr/oopsla18.pdf

Practical Verification of Peephole Optimizations with Alive.
Nuno Lopes, David Menendez, Santosh Nagarakatte, and John Regehr.
Communications of the ACM. Volume 61 Issue 2, February 2018.
https://dl.acm.org/citation.cfm?id=3166064

Research for practice: vigorous public debates in academic computer science.
John Regehr.
Communications of the ACM. Volume 60 Issue 12, December 2017.
https://dl.acm.org/citation.cfm?doid=3132257

Taming Undefined Behavior in LLVM.
Juneyoung Lee, Yoonseung Kim, Youngju Song, Chung-Kil Hur, Sanjoy Das, David Majnemer, John Regehr, and Nuno P. Lopes.
In Proceedings of 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2017), Barcelona, Spain, June 2017.
http://www.cs.utah.edu/~regehr/papers/undef-pldi17.pdf

Abstractions for Practical Virtual Machine Replay.
Anton Burtsev, David Johnson, Mike Hibler, Eric Eide, and John Regehr.
In *Proceedings of the 12th International Conference on Virtual Execution Environments (VEE'16)*, Atlanta, GA, USA, April 2016.
https://www.cs.utah.edu/~regehr/papers/vee16-xentt.pdf

Alex Groce, Mohammad Amin Alipour, Chaoqiang Zhang, Yang Chen, and John Regehr.
Cause reduction: delta debugging, even without bugs.
In Software Testing, Verification and Reliability, Volume 26 Issue 1, January 2016.
http://www.cs.utah.edu/~regehr/papers/mintest.pdf

Understanding Integer Overflow in C/C++.
Will Dietz, Peng Li, John Regehr, and Vikram Adve.
In ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 25, Issue 1, November 2015.
http://www.cs.utah.edu/~regehr/papers/tosem15.pdf

Deniable Backdoors Using Compiler Bugs.
Scotty Bauer, Pascal Cuoq, and John Regehr.
International Journal of PoC||GTFO 0x08, June 2015.
https://www.alchemistowl.org/pocorgtfo/pocorgtfo08.pdf#page=7

Nuno Lopes, David Menendez, Santosh Nagarakatte, and John Regehr.
Provably Correct Peephole Optimizations with Alive.
In *Proceedings of 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2015)*, Portland, OR, USA, June 2015.
This paper received the SIGPLAN Distinguished Paper Award.
http://www.cs.utah.edu/~regehr/papers/pldi15.pdf

Alex Groce, Amin Alipour, Chaoqiang Zhang, Yang Chen, and John Regehr.
Cause Reduction for Quick Testing.

In *Proceedings of the IEEE International Conference on Software Testing, Verification and Validation (ICST)*, Cleveland, Ohio, USA, March-April 2014.
This paper received the ICST 2014 Best Paper Award.
`http://www.cs.utah.edu/~regehr/papers/icst14.pdf`

Alex Groce, Chaoqiang Zhang, Mohammad Amin Alipour, Eric Eide, Yang Chen, John Regehr.
Help, help, I'm being suppressed! The significance of suppressors in software testing.
In *Proceedings of the 24th International Symposium on Software Reliability Engineering (ISSRE)*, Pasadena, CA, USA, November 2013.
`http://www.cs.utah.edu/~regehr/papers/issre13.pdf`

Yang Chen, Alex Groce, Chaoqiang Zhang, Weng-Keen Wong, Xiaoli Fern, Eric Eide, and John Regehr.
Taming Compiler Fuzzers.
In *Proceedings of 34th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2013)*, Seattle, WA, USA, June 2013.
`http://www.cs.utah.edu/~regehr/papers/pldi13.pdf`

Alex Groce, Chaoqiang Zhang, Eric Eide, Yang Chen, and John Regehr.
Swarm Testing.
In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA 2012)*, Minneapolis, MN, USA, July 2012.
`http://www.cs.utah.edu/~regehr/papers/swarm12.pdf`

John Regehr, Yang Chen, Pascal Cuoq, Eric Eide, Chucky Ellison, and Xuejun Yang.
Test-Case Reduction for C Compiler Bugs.
In *Proceedings of 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2012)*, Beijing, China, June 2012.
`http://www.cs.utah.edu/~regehr/papers/pldi12-preprint.pdf`

Will Dietz, Peng Li, John Regehr, and Vikram Adve.
Understanding Integer Overflow in C/C++.
In *Proceedings of the 34th International Conference on Software Engineering (ICSE 2012)*. Zurich, Switzerland, June 2012.
This paper received the ACM SIGSOFT Distinguished Paper Award.
`http://www.cs.utah.edu/~regehr/papers/overflow12.pdf`

Pascal Cuoq, Benjamin Monate, Anne Pacalet, Virgile Prevosto, John Regehr, Boris Yakobowski, and Xuejun Yang.
Testing static analyzers with randomly generated programs.
Short paper in *Proceedings of the 4th NASA Formal Methods Symposium (NFM 2012)*. Norfolk, Virginia, USA, April 2012.
`http://www.cs.utah.edu/~regehr/papers/nfm12.pdf`

Lu Zhao, Guodong Li, and John Regehr.
ARMor: Fully Verified Software Fault Isolation.
In *Proceedings of the International Conference on Embedded Software (EMSOFT)*, Taipei, Taiwan, October 2011.

http://www.cs.utah.edu/~regehr/papers/emsoft11.pdf

Xuejun Yang, Yang Chen, Eric Eide, and John Regehr.
Finding and Understanding Bugs in C Compilers.
In *Proceedings of 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2011)*, San Jose, CA, USA, June 2011.
http://www.cs.utah.edu/~regehr/papers/pldi11-preprint.pdf

Peng Li and John Regehr.
T-Check: Bug Finding for Sensor Networks.
In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, SPOTS track, Stockholm, Sweden, April 2010.
http://www.cs.utah.edu/~regehr/papers/ipsn553s-li.pdf

Yang Chen, Omprakash Gnawali, Maria Kazandjieva, Philip Levis, and John Regehr.
Surviving Sensor Network Software Faults.
In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP 2009)*, Big Sky, MT, USA, October 2009.
http://www.sigops.org/sosp/sosp09/papers/chen-sosp09.pdf

Xuejun Yang, Nathan Cooprider, and John Regehr.
Eliminating the Call Stack to Save RAM.
In *Proceedings of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2009)*, Dublin, Ireland, June 2009.
http://www.cs.utah.edu/~regehr/papers/lctes062-yang.pdf

Jon Rafkind, Adam Wick, John Regehr, and Matthew Flatt.
Precise Garbage Collection for C.
In *Proceedings of the 2009 International Symposium on Memory Management (ISMM)*, Dublin, Ireland, June 2009.
http://www.cs.utah.edu/~regehr/papers/ismm15-rafkind.pdf

Eric Eide and John Regehr.
Volatiles are miscompiled, and what to do about it.
In *Proceedings of the ACM Conference on Embedded Software (EMSOFT)*, Atlanta, GA, October 2008.
http://www.cs.utah.edu/~regehr/papers/emsoft08-preprint.pdf

Venkat Chakravarthy, John Regehr, and Eric Eide.
Edicts: Implementing Features with Flexible Binding Times.
In *Proceedings of the 7th International Conference on Aspect-Oriented Software Development (AOSD)*, Brussels, Belgium, April 2008.
http://www.cs.utah.edu/~regehr/papers/aosd08-preprint.pdf

Nathan Cooprider, William Archer, Eric Eide, David Gay, and John Regehr.
Efficient Memory Safety for TinyOS.
In *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys 2007)*, Sydney, Australia, November 2007.
http://www.cs.utah.edu/~regehr/papers/coop-sensys07.pdf

Nathan Cooprider and John Regehr.
Offline Compression for On-Chip RAM.
In *Proceedings of the ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation (PLDI 2007)*, pages 363–372, San Diego, CA, June 2007.
http://www.cs.utah.edu/~regehr/papers/pldi075-cooprider.pdf

John Regehr and Nathan Cooprider.
Interrupt Verification via Thread Verification.
Electronic Notes in Theoretical Computer Science (ENTCS), 174(9):139–150, June 2007.
http://www.sciencedirect.com/science/article/pii/S1571066107003623

Will Archer, Philip Levis, and John Regehr.
Interface Contracts for TinyOS.
In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN) 2007, SPOTS track*, pages 158–165, Cambridge, MA, April 2007.
http://www.cs.utah.edu/~regehr/papers/spots07.pdf

Nathan Cooprider and John Regehr.
Pluggable Abstract Domains for Analyzing Embedded Software.
In *Proceedings of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2006)*, pages 44–53, Ottawa, Canada, June 2006.
http://www.cs.utah.edu/~regehr/papers/lctes06_1

John Regehr and Usit Duongsaa.
Deriving Abstract Transfer Functions for Analyzing Embedded Software.
In *Proceedings of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2006)*, pages 34–43, Ottawa, Canada, June 2006.
http://www.cs.utah.edu/~regehr/papers/lctes06_2

John Regehr, Alastair Reid, and Kirk Webb.
Eliminating stack overflow by abstract interpretation.
ACM Transactions on Embedded Computing Systems, 4(4):751–778, November 2005.
http://portal.acm.org/citation.cfm?id=1113830.1113833

John Regehr.
Random testing of interrupt-driven software.
In *Proceedings of the ACM Conference on Embedded Software (EMSOFT)*, pages 290–298, Jersey City, NJ, September 2005.
http://www.cs.utah.edu/~regehr/papers/emsoft05

John Regehr and Usit Duongsaa.
Preventing interrupt overload.
In *Proceedings of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2005)*, pages 50–58, Chicago, IL, June 2005.
http://www.cs.utah.edu/~regehr/papers/lctes05/

John Regehr and Alastair Reid.
HOIST: A system for automatically deriving static analyzers for embedded systems.

In *Proceedings of the Eleventh International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 133–143, Boston, MA, October 9–13 2004.
http://www.cs.utah.edu/~regehr/papers/asplos04/

Eric Eide, Tim Stack, John Regehr, and Jay Lepreau.
Dynamic CPU Management for Real-Time, Middleware-Based Systems.
In *Proceedings of the Real-Time Technology and Applications Symposium (RTAS)*, pages 286–295, Toronto, Canada, May 25–28 2004.
http://www.cs.utah.edu/flux/papers/cpubroker-rtas04-base.html

John Regehr, Alastair Reid, Kirk Webb, Michael Parker, and Jay Lepreau.
Evolving real-time systems using hierarchical scheduling and concurrency analysis.
In *Proceedings of the 24th IEEE Real-Time Systems Symposium (RTSS)*, pages 25–36, Cancun, Mexico, December 3–5 2003.
http://www.cs.utah.edu/flux/papers/cee-rtss03/

John Regehr, Alastair Reid, and Kirk Webb.
Eliminating stack overflow by abstract interpretation.
In *Proceedings of the Third International Conference on Embedded Software (EMSOFT)*, pages 751–778, Philadelphia, PA, October 15–17 2003.
http://www.cs.utah.edu/flux/papers/emsoft03

John Regehr.
Scheduling Tasks with Mixed Preemption Relations for Robustness to Timing Faults.
In *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS)*, pages 315–326, Austin, TX, December 3–5 2002.
http://www.cs.utah.edu/flux/papers/spak-flux-tn-02-01/

John Regehr.
Inferring Scheduling Behavior with Hourglass.
In *Proceedings of the 2002 USENIX Annual Technical Conference FREENIX track*, pages 143–156, Monterey, CA, June 10–15 2002.
http://www.cs.utah.edu/flux/papers/hourglass-usenix02/

Eric Eide, Alastair Reid, John Regehr, and Jay Lepreau.
Static and Dynamic Structure in Design Patterns.
In *Proceedings of the 2002 International Conference on Software Engineering (ICSE)*, pages 208–218, Orlando, FL, May 19–25 2002.
http://www.cs.utah.edu/flux/papers/knit-icse02-base.html

John Regehr and John A. Stankovic.
HLS: A Framework for Composing Soft Real-Time Schedulers.
In *Proceedings of the 22nd IEEE Real-Time Systems Symposium (RTSS)*, pages 3–14, London, UK, December 3–6 2001.
http://www.cs.utah.edu/flux/papers/hls-rtss01/

John Regehr and John A. Stankovic.
Augmented CPU Reservations: Towards Predictable Execution on General-Purpose Operating Systems.

In *Proceedings of the 7th Real-Time Technology and Applications Symposium (RTAS)*, pages 141–148, Taipei, Taiwan, May 30–June 1 2001.
http://www.cs.utah.edu/~regehr/papers/augmented/

Michael B. Jones, John Regehr, and Stefan Saroiu.
Two Case Studies in Predictable Application Scheduling Using Rialto/NT.
In *Proceedings of the 7th Real-Time Technology and Applications Symposium (RTAS)*, pages 157–164, Taipei, Taiwan, May 30–June 1 2001.
http://www.cs.utah.edu/~regehr/papers/rialtont_apps/

Michael B. Jones and John Regehr.
CPU Reservations and Time Constraints: Implementation Experience on Windows NT.
In *Proceedings of the 3rd USENIX Windows NT Symposium*, Seattle, WA, July 1999.
http://www.cs.utah.edu/~regehr/papers/usenixnt99/

Workshop Publications

Raimondas Sasnauskas and John Regehr.
Intent fuzzer: crafting intents of death.
In *Proceedings of the 2014 Joint International Workshop on Dynamic Analysis (WODA) and Software and System Performance Testing, Debugging, and Analytics (PERTEA)*. San Jose, CA, USA, July 2014.
http://www.cs.utah.edu/~regehr/papers/p1-sasnauskas.pdf

Partha Pal, Rick Schantz, Aaron Paulos, John Regehr, and Mike Hibler.
Advanced Adaptive Application (A3) Environment: Initial Experience.
In *Proceedings of the Middleware 2011 Industry Track Workshop (Middleware '11)*. New York, NY, USA. December 2011.
http://www.cs.utah.edu/~regehr/papers/middleware11.pdf

Lu Zhao, Guodong Li, and John Regehr.
A Practical Logic Framework for Verifying Safety Properties about Executables.
In *Proceedings of the Workshop on Syntax and Semantics of Low-Level Languages (LOLA 2011)*, Toronto, Canada, June 2011.
http://www.cs.utah.edu/~regehr/papers/lola11.pdf

Jianjun Duan and John Regehr.
Correctness Proofs for Device Drivers in Embedded Systems.
In *Proceedings of the 5th International Workshop on Systems Software Verification (SSV)*, Vancouver, Canada, October 2010.
http://www.cs.utah.edu/~regehr/papers/ssv10.pdf

Usa Sammapun, John Regehr, Insup Lee, and Oleg Sokolsky.
Runtime Verification for Wireless Sensor Network Applications.
In *Proceedings of the Dagstuhl Seminar 07011 on Runtime Verification*, January 2007.
http://drops.dagstuhl.de/portals/index.php?semnr=07011

John Regehr and Phil Levis.

High Confidence TinyOS.
In *Proceedings of the Composable and Systems Technology for High Confidence Cyber-Physical Systems Workshop*, Arlington, VA, July 2007.
`http://www.cs.utah.edu/~regehr/papers/hccps07.pdf`

John Regehr, Nathan Cooprider, and David Gay.
Atomicity and Visibility in Tiny Embedded Systems.
In *Proceedings of the PLOS 2006 Workshop on Linguistic Support for Modern Operating Systems*, San Jose, CA, October 2006.
`http://www.cs.utah.edu/~regehr/papers/plos06b.pdf`

John Regehr, Nathan Cooprider, Will Archer, and Eric Eide.
Efficient Type and Memory Safety for Tiny Embedded Systems.
In *Proceedings of the PLOS 2006 Workshop on Linguistic Support for Modern Operating Systems*, San Jose, CA, October 2006.
`http://www.cs.utah.edu/~regehr/papers/plos06a.pdf`

John Regehr.
Thread Verification vs. Interrupt Verification.
In *Proceedings of the Workshop on Multithreading in Hardware and Software: Formal Approaches to Design and Verification (TV06)*, Seattle, WA, August 2006.
`http://www.cs.utah.edu/~regehr/papers/tv06.pdf`

John Regehr, Konrad Slind, and Elsa Gunter.
Proofs as a substrate for tool integration supporting high-confidence embedded software.
In *Proceedings of the High Confidence Medical Device Software and Systems (HCMDSS) Workshop*, Philadelphia, PA, June 2005.
`http://www.cs.utah.edu/~regehr/papers/hcmdss05.pdf`

John Regehr.
Opportunities and Challenges for the Real-Time Community.
Invited paper at the 2003 Workshop on Challenges in Embedded Real-Time Systems, Cancun, Mexico, December 2 2003.

John Regehr.
Vertically Integrated Analysis and Transformation for Embedded Software.
In *Proceedings of the 2003 Workshop on Compilers and Tools for Constrained Embedded Systems (CTCES)*, San Jose, CA, October 29 2003.
`http://www.cs.utah.edu/flux/papers/ctces03/`

John Regehr and Alastair Reid.
Lock Inference for Systems Software.
In *Proceedings of the Second AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS)*, Boston, MA, March 17 2003.
`http://www.cs.utah.edu/flux/papers/lock-inference-03`

John Regehr and Jay Lepreau.
The Case for Using Middleware to Manage Diverse Soft Real-Time Schedulers.

In *Proceedings of the International Workshop on Multimedia Middleware (M3W)*, Ottawa, Canada, October 2001.
http://www.cs.utah.edu/flux/papers/crm-m3w01/

Michael B. Jones and John Regehr.
The Problems You're Having May Not Be the Problems You Think You're Having: Results from a Latency Study of Windows NT.
In *Proceedings of the 7th Workshop on Hot Topics in Operating Systems (HotOS)*, pages 96–101, Rio Rico, AZ, March 1999.
http://www.cs.utah.edu/~regehr/papers/hotos7/

Michael B. Jones and John Regehr.
Issues in Using Commodity Operating Systems for Time-Dependent Tasks: Experiences from a Study of Windows NT.
In *Proceedings of the 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Cambridge, England, July 1998.

Other Writing

Nuno P. Lopes and John Regehr.
Future Directions for Optimizing Compilers.
2019. https://arxiv.org/abs/1809.02161

John Regehr.
Do small-RAM devices have a future?
Circuit Cellar 25th anniversary issue, 2013.

John Regehr.
Safe and Structured Use of Interrupts in Real-Time and Embedded Software.
Chapter in *Handbook of Real-Time and Embedded Systems*, CRC Press, 2007.
http://www.cs.utah.edu/~regehr/papers/interrupt_chapter.pdf

John Regehr.
Teaching Reliability.
IEEE Distributed Systems Online, vol. 7, no. 5, 2006, art. no. 0605-05002.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1642629

John Regehr.
Say No to Stack Overflow.
Embedded Systems Programming, 17(10), October 2004.
http://www.embedded.com/showArticle.jhtml?articleID=47101892

Patent

Michael B. Jones and John Regehr.
Providing predictable scheduling of programs using repeating precomputed schedules on discretely scheduled and/or multiprocessor operating systems.

US patent 7,000,232. Awarded February 2006.

Invited Talks

"Fuzzing and Formal Methods: A Happy Partnership for Compiler Validation." School of Informatics, University of Edinburgh, January 2023.

"Minotaur: A Synthesizing Superoptimizer." School of Informatics, University of Edinburgh, September 2022.

"Alive2: Verifying Existing Optimizations." With Nuno P. Lopes. LLVM Developers' Meeting, October 2019.

"Building Compilers Using Data and Formal Methods." Programming Language Enthusiasts Mind Melt (PLEMM), September 2019.

"Undefined Behavior and Compiler Optimizations." C++Now, Aspen, Colorado, May 2018.

"Undefined Behavior in 2017." CppCon, Seattle, Washington, October 2017.

"Superoptimizing LLVM." Department of Computer Science, University of Washington, December 2014.

"Integer Safety." Workshop on Cyber Risk and Information Security, EPFL, Switzerland, June 2014.

"Fuzzing and Rust." Rust Meetup, San Francisco CA, May 2014.

"Towards Beautiful Test Cases for Compiler Bugs." Department of Computer Science, University of Iowa, April 2012.

"Towards Beautiful Test Cases for Compiler Bugs." Google Tech Talk, March 2012.

"Finding C Compiler Bugs with Random Testing." Invited talk at the Department of Computer Science, Purdue University, February 2012.

"Testing Embedded Software." Keynote at ISSTA 2011, the International Symposium on Software Testing and Analysis.

"Finding and Understanding Bugs in C Compilers." Invited talk at the Department of Computer Science, University of Illinois, April 2011.

"Finding and Understanding Bugs in C Compilers." Invited talk at the Department of Computer Science, Northwestern University, April 2011.

"Random Testing for C Compilers." Invited talk at SPIN 2011, the 18th International Workshop on Model Checking of Software.

"Exposing Difficult Compilers Bugs With Random Testing." Talk at the GCC Summit, Ottawa Canada, October 2010.

"Safety Analysis for Wireless Embedded Networks." Series of three lectures given at RWTH Aachen, February 17–19 2010.

"Safe TinyOS." Part of a three-hour tutorial on TinyOS given at IPSN 2009 with other members of the TinyOS Core Working Group. April 16 2009.

"Static Analysis of Interrupt-Driven Embedded C." Invited talk at the Department of Computer Science, University of Virginia, April 7 2008.

"Static Analysis of Interrupt-Driven Embedded C." Invited talk at the Department of Computer Science, Washington University, January 18 2008.

"Static Analysis of Interrupt-Driven Embedded C." Invited talk at Australia's Information and Communications Technology Centre of Excellence (NICTA), November 7 2007.

"Using Application- and System-Specific Heuristics to Compile Embedded Software for Real-Time and Other Resource Constraints." Invited talk at the Department of Computer Science, University of Pittsburgh, November 5 2004.

"Say No to Stack Overflow." Invited talk at the Department of Computer Science, University of Maryland, June 8 2004.

"Say No to Stack Overflow." Invited talk at the Department of Computer Science, Brigham Young University, March 4 2004.

"Vertically Integrated Analysis and Transformation for Embedded Software." Invited talk at the 2003 Workshop on Constraint-Aware Embedded Software, Cancun, Mexico, December 2 2003.

"High-Level Optimizations for Low-Level Software." Presented as a work in progress at the 19th ACM Symposium on Operating Systems Principles, Bolton Landing, New York, October 21 2003.

"High-Level Optimizations for Low-Level Software." Invited talk at the Department of Computer and Information Science, University of Pennsylvania, October 17 2003.

"Creating Embedded Software." Talk at the School of Computing, University of Utah, October 12 2003.

"How to Rapidly Prototype a Real-Time Scheduler," with Luca Abeni. Presented as a work in progress at the 23rd IEEE Real-Time Systems Symposium (RTSS), Austin, TX, December 4 2002.

"Some Guidelines for Proportional Share CPU Scheduling in General-Purpose Operating Systems." Presented as a work in progress at the 22nd IEEE Real-Time Systems Symposium (RTSS), London, UK, December 5 2001.

"Hierarchical Schedulers, Performance Guarantees, and Resource Management," with John A. Stankovic. Presented as a work in progress at the 17th ACM Symposium on Operating Systems Principles, Kiawah Island, South Carolina, December 1999.

"The Problems You're Having May Not Be the Problems You Think You're Having: Results from a Latency Study of Windows NT," with Michael B. Jones. Invited talk at the Real-Time Applications and Systems Symposium, Vancouver, Canada, June 1999.

"Myricom and Linux," with Bob Felderman. Invited talk at the First Extreme Linux Workshop, Santa Fe, New Mexico, February 1998.

Current Advisees

Chloe Pronovost, PhD, June 2023–present

Yuyou Fan, PhD, August 2021–present

Manasij Mukherjee, PhD, August 2018–present

Zhengyang Liu, PhD, August 2017–present

Graduated Students

Vsevolod Livinskii, PhD, May 2024

Jubi Taneja, PhD, May 2022

Pranav Kant, MS, May 2020

Xuejun Yang, PhD, May 2014

Yang Chen, PhD, December 2013

Jianjun Duan, PhD, July 2013

Anton Burtsev, PhD, December 2012

Lu Zhao, PhD, May 2012

Rohit Pagariya, MS, December 2010

Venkat Chakravarthy, MS, May 2009

Nathan Cooprider, PhD, August 2008

Usit Duongsaa, MS, May 2006

Postdocs

Nader Bushehri, January 2021–December 2022

Raimondas Sasnauskas, July 2013–May 2015

Undergraduate researchers supervised

Stefan Mada, August 2021–present

Ryan Berger, August 2021–May 2022

Currently Maintained Open Source Software

Alive2: translation validation for LLVM IR
`https://github.com/AliveToolkit/alive2`

Souper: a superoptimizer for LLVM
`https://github.com/google/souper`

Alive: a verifier for LLVM optimizations
`https://github.com/nunoplopes/alive`

C-Reduce: a tool for minimizing the size of C and C++ programs that trigger compiler bugs
`http://embed.cs.utah.edu/creduce/`

Csmith: a tool for finding bugs in C compilers
`http://embed.cs.utah.edu/csmith/`

Integer Overflow Checker for C and C++ code
this software is now part of the LLVM/Clang compiler's undefined behavior sanitizer

Other Awards

Most Influential PLDI Paper Award, 2022, for our 2012 PLDI paper "Test-Case Reduction for C Compiler Bugs"

Most Influential PLDI Paper Award, 2021, for our 2011 PLDI paper "Finding and understanding bugs in C compilers"

Google Open Source Award for C-Reduce, March 2018. Co-awardees: Eric Eide and Yang Chen.

Professional Activities

Program Chair, PLDI 2024

Area Chair, PLDI 2023

Member, LLVM Foundation Board of Directors, September 2016–August 2020

Artifact evaluation co-chair, PLDI 2016

NSF Panelist in 2021, 2018, 2016, 2014, 2013, 2011, 2008, 2007, 2003

Artifact evaluation co-chair, PLDI 2015

Poster and Lightning Round Chair, ASPLOS 2014

Area Chair, NSF Workshop on Formal Methods: Future Directions and Its Transition To Practice, 2012

Program Chair, International Conference on Embedded Software (EMSOFT) 2012, co-chair Florence Maraninchi

Associate Editor, ACM Transactions on Sensor Networks, October 2010–December 2011

Program Chair, IEEE 2010 Real-Time Systems Symposium (RTSS), track on sensor networks

Demo co-Chair ACM Conference on Embedded Networked Sensor Systems (SenSys) 2010

Program Chair, ACM SIGPLAN/SIGBED 2008 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)

Program Chair, IEEE 2006 Real-Time Technology and Applications Symposium (RTAS), track on Development, Verification, and Debugging Tools for Real-Time and Embedded Systems

Work in Progress Co-chair, Real-Time Systems Symposium (RTSS) 2004

Student Travel Awards Chair, Conference on Embedded Networked Sensor Systems (SenSys) 2004

Web Chair, ACM 2004 Conference on Languages, Compilers, and Tools for Embedded Systems

Scribe, Symposium on Operating Systems Principles (SOSP) 1999

Program Committee Member

ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2022

Fuzzing Workshop 2022

International Symposium on Code Generation and Optimization (CGO) 2022

18th Workshop on Hot Topics in Operating Systems (HotOS) 2021

ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2020

European Conference on Computer Systems (EuroSys) 2019

International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2019

First Workshop on Speculative Side Channel Analysis (WoSSCA) 2018

Compiler Construction (CC) 2018

International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2017

Compiler Construction (CC) 2017

New Ideas and Emerging Results at the International Conference on Software Engineering (NIER-ICSE) 2017

LLVM Performance Workshop at CGO 2017

International Conference on Software Engineering (ICSE) 2016

2015 European LLVM Conference

ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2015

International Conference on Compiler Construction (CC) 2015

International Conference on Embedded Software (EMSOFT) 2014

European Conference on Wireless Sensor Networks (EWSN) 2014

International Conference on Embedded Software (EMSOFT) 2013

NASA Formal Methods Symposium (NFM) 2013

ACM International Symposium on Memory Management (ISMM) 2012

Real-Time and Embedded Technology and Applications Symposium (RTAS) 2012, Applications, Systems, RTOS and Tools track

USENIX Annual Technical Conference 2012

Computer Aided Verification (CAV) 2011

Real-Time and Embedded Technology and Applications Symposium (RTAS) 2011, Sensor Network track

International Conference on Distributed Computing Systems (ICDCS) 2011, Sensor Network track

ACM SIGPLAN/SIGBED 2011 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)

USENIX Annual Technical Conference 2010

European Conference on Computer Systems (EuroSys) 2010

International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES) 2010

ACM Conference on Embedded Networked Sensor Systems (SenSys) 2009

Design Automation Conference (DAC) 2009

Real-Time and Embedded Technology and Applications Symposium (RTAS) 2009

Real-Time Systems Symposium (RTSS) 2009

Real-Time Systems Symposium (RTSS) track on Wireless Sensor Networks 2008

IEEE 17th International Conference on Computer Communications and Networks (ICCCN) 2008

IEEE/IFIP International Conference On Embedded and Ubiquitous Computing (EUC) 2008

Real-Time and Embedded Technology and Applications Symposium (RTAS) 2008

International Conference on Embedded Software (EMSOFT) 2007

Workshop on the Interaction between Compilers and Computer Architecture (INTERACT) 2007

International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) 2007

Workshop on Embedded Sensor Networks (EmNets) 2007

ACM SIGPLAN/SIGBED 2007 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)

International Conference on High Performance Embedded Architectures and Compilers (HiPEAC) 2007

Real-Time Systems Symposium (RTSS) 2007

Real-Time and Embedded Technology and Applications Symposium (RTAS) 2007

International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) 2006

International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS) 2006

Workshop on Operating Systems Platforms for Embedded Real-Time applications (2006)

Workshop on Multithreading in Hardware and Software: Formal Approaches to Design and Verification (TV) 2006

Real-Time Systems Symposium (RTSS) 2006

Conference on Distributed Computing in Sensor Systems (DCOSS) 2006

Workshop on Parallel and Distributed Real-Time Systems (WPDRTS) 2006

Java Technologies for Real-Time and Embedded Systems (JTRES) 2005

Real-Time Systems Symposium (RTSS) 2005

ACM SIGPLAN/SIGBED 2005 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)

Real-Time and Embedded Technology and Applications Symposium (RTAS) 2005

Workshop on High Performance, Fault Adaptive, Large Scale Embedded Real-Time Systems (FALSE-II) 2005

Workshop on Compilers and Tools for Constrained Embedded Systems (CTCES) 2004

Real-Time Systems Symposium (RTSS) Track on Real-Time Middleware and Software Engineering 2004

Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES) 2004

Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS) 2004

Real-Time and Embedded Technology and Applications Symposium (RTAS) 2004

Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES) 2003

Real-Time Systems Symposium (RTSS) 2003

Real-Time Systems Symposium (RTSS) Work in Progress Session 2003

International Conference on Distributed Computing Systems (ICDCS) 2003

Real-Time and Embedded Technology and Applications Symposium (RTAS) 2003

External Review Committee Member

International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2020

International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2018

ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2017

International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2016

International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2015

ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2013

ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2012

ACM International Symposium on Memory Management (ISMM) 2011

Internal Service

Associate Director, School of Computing, 2020–2022, 2023–present

Member, executive committee, 2017–2019

Member, hiring committee, 2019

Member, gemstone hiring committee, 2018

Chair, Retention, Promotion, and Tenure committee, 2017–2019

Member, academic senate, 2017–2019

Facility oversight, 2007–2018

Member, database hiring committee, 2017

Member, architecture hiring committee, 2015

Director of graduate admissions, 2014–2015

Chair, operating systems faculty search committee, 2014

Chair, security faculty search committee, 2013

Member, computer engineering track committee, 2006–2016

Director, BS/MS program, 2005–2008

Member, curriculum committee, 2005–2008

Member, graduate admissions committee, 2004–2005, 2005–2006, 2007–2008, 2010–2011

Organizer, Distinguished lecture series, 2004–2005, 2006–2008

Member, systems faculty recruiting committee, 2007–2008

Member, graphics faculty recruiting committee, 2007–2008

Member, facility committee, 2006–2007

Member, external relations committee, 2006–2007

Member, undergraduate studies committee, 2003–2004