

Felix S. Klock II

42 Pinecliff Dr, Marblehead, MA, 01945 USA, phone: +1 857.472.3757, e-mail: felix.klock@gmail.com

OBJECTIVE	“Stop the bugs; save the world.” Innovative software development, leveraging my experience in compiler and language runtime technology, in a challenging environment with an enthusiastic, smart, and respected peer group.	
SOFTWARE DEVELOPMENT SKILLS & INTERESTS	Programming languages: runtime design, memory management, JIT, static analysis Software engineering: functional programming, debugging tool design, CS education Languages: Rust, C/C++, Scheme/Lisp, Java, Python, C#, assembly (x86, ARM).	
EDUCATION	Northeastern University , CCIS, Boston, MA	2003 – 2010
	<i>Doctor of Philosophy in Computer Science</i>	
	Massachusetts Institute of Technology , Cambridge, MA	1996 – 2001
	<i>Bachelor of Science in Computer Science</i> , 2000	
	<i>Master of Engineering in Electrical Engineering and Computer Science</i> , 2001	
PROFESSIONAL EXPERIENCE	Amazon Web Services , remote	October 2020 – December 2024
	<i>Principal Engineer, Rust Platform Team</i>	
	Rust Roles: (1) Compiler team co-lead, (2) Specification team co-lead (3) Language Design team member, (4) Compiler performance working group member.	
	Project home page: https://www.rust-lang.org/	
	Achievements: Async Rust tooling+education (TurboWish blog posts, <code>tokio-console</code> performance debugger, <code>re:Invent</code> workshop tutorial); Developer screencasts (<code>rustc</code> development tutorials, “debugging as a science,” <code>rr</code> and <code>pernos.co</code>); Safety validation (AWS-internal distribution of <code>miri</code> , the Krabcake <code>valgrind</code> tool, Rust Contracts).	
	Mozilla , Paris, France and remote	2012 – August 2020
	<i>Staff Research Engineer, Rust language</i>	
	Achievements: Spun up working groups: <i>wg-prioritization</i> (bug triage), <i>wg-incr-comp</i> (incremental compilation), <i>wg-rustc-perf</i> (monitoring rustc performance). Oversaw non-lexical lifetimes (NLL; Rust RFC 2094), migration from lexical to NLL. Future-incompatibility reporting (Rust RFC 2834), user-defined destructors with type/lifetime generics (Rust RFC 769), non-zeroing dynamic drop (Rust RFC 320), and Rust’s user-defined allocators (Rust RFC 1398).	
	Adobe Systems Incorporated , Waltham, MA	2010 – 2012
	<i>Computer Scientist, Actionscript 3 Virtual Machine (aka Tamarin) for Flash Runtime</i>	
	Roles: Garbage Collection (GC) expert, integration lead, cross-platform build.	
	Project history through May 2012: http://hg.mozilla.org/tamarin-redux/	
	Achievements: Improved telescoping GC inverse load-factor, reducing overhead from 20× to 4× mark/cons ratio. Implemented Native JSON integrated with AS3, and extended with serialization of public members of AS3 classes. Added JIT-support for efficient indexing of “simple dense” <code>ArrayObject</code> , yielding 10–20% speedup.	
	Northeastern University , Boston, MA	2005 – 2010
	<i>Developer and maintainer of Larceny Scheme compiler and runtime system</i>	
	Project history: http://github.com/larcenists/larceny/	
	Achievements: Regional GC, bounding mutator utilization. In-heap dynamic machine code emission for x86 backend. Dyanmic .NET bytecode emission for Common Larceny.	

	<p>Northeastern University, Boston, MA 2003 – 2009 <i>Instructor of Record/Teaching Assistant</i></p>
	<p>Green Hills Software, Santa Barbara, CA 2001 – 2003 <i>Compiler Engineer for End-User Product Development</i> Achievements: Move-coalescing register allocation; data-load optimizations; reassignment of zero- and uninitialized arrays to blank static segment (bss).</p>
	<p>Massachusetts Institute of Technology, Cambridge, MA 2000 – 2001 <i>Teaching Assistant, 6.170; Head Teaching Assistant Fall 2000</i></p>
	<p>MIT Laboratory for Computer Science, Cambridge, MA 1999 – 2001 <i>Undergraduate Research Assistant for Computer Architecture Group</i> Assisted with <i>FLEX</i> compiler for transforming Java bytecode to machine code .</p>
	<p>MIT Media Laboratory, Cambridge, MA 1997 – 1998 <i>Undergraduate Research Assistant for Software Agents Group</i> Helped develop <i>Footprints</i>, a tool for visualizing navigation of users on Web.</p>
RESEARCH PUBLICATIONS	<p>Felix S Klock II, “A Declarative DSL for Customized Rendering of Text-Based Art” , in <i>Proceedings of the 2017 International Symposium on Practical Aspects of Declarative Languages</i>, PADL 2017, Paris, France January 2017.</p> <p>Felix S Klock II and William D Clinger. “Bounded-latency regional garbage collection” , In <i>Proceedings of the 2011 Dynamic Languages Symposium</i>, DLS 2011, Portland, OR, 24 October 2011, pages 73-83.</p> <p>William D Clinger and Felix S Klock II. “Scalable Garbage Collection with Guaranteed MMU” , In <i>Proceedings of the 2009 Workshop on Scheme and Functional Programming</i>, Northeastern University, Boston, MA, 22 August 2009</p> <p>Felix S Klock II, “The Layers of Larceny’s Foreign Function Interface” , In <i>Proceedings of the 2009 Workshop on Scheme and Functional Programming</i>, Victoria, British Columbia, 20 September 2008</p>
SELECTED PRESENTATIONS	<p>“Contracts for Rust, Revisited” Rust Formal Methods Interest Group 2024 https://www.youtube.com/watch?v=j60gcdpHNAM</p> <p>“Fireside chat: Felix Klock and Greg Morrisett” ICFP 2023 https://www.youtube.com/watch?v=msw47HLb3zo</p> <p>“Krabcake: A Rust UB detector” Rust Verification Workshop 2023 http://pnkfx.org/presentations/krabcake-rust-verification-2023-april.pdf</p> <p>“Rust: Fearless at all levels” Algolia Search Party 2018 https://www.youtube.com/watch?v=9QuI2Z0stbs&t=2755</p> <p>“Subtyping in Rust and Clarke’s Third Law” Rust Fest 2016 https://www.youtube.com/watch?v=fI4RG_uq-WU</p>
HONORS AND AWARDS	<p>Teaching Award, Northeastern University, 2008</p> <p>Northern Telecom/BNR Digital Systems Laboratory Project Award, 2001</p>
INTERESTS	<p>Cooking; reading, especially historical discussions of mathematics, logic, and language development; skiing; graphics programming</p>