Rust: From POPL to Practice (Keynote)

Aaron Turon
Mozilla, USA
aturon@mozilla.com

Abstract

In 2015, a language based fundamentally on substructural typing–Rust–hit its 1.0 release, and less than a year later it has been put into production use in a number of tech companies, including some household names. The language has started a trend, with several other mainstream languages, including C++ and Swift, in the early stages of incorporating ideas about ownership. How did this come about?

Rust's core focus is safe systems programming. It does not require a runtime system or garbage collector, but guarantees memory safety. It does not stipulate any particular style of concurrent programming, but instead provides the tools needed to guarantee data race freedom even when doing low-level shared-state concurrency. It allows you to build up high-level abstractions without paying a tax; its compilation model ensures that the abstractions boil away.

These benefits derive from two core aspects of Rust: its ownership system (based on substructural typing) and its trait system (a descendant of Haskell's typeclasses). The talk will cover these two pillars of Rust design, with particular attention to the key innovations that make the language usable at scale. It will highlight the implications for concurrency, where Rust provides a unique perspective. It will also touch on aspects of Rust's development that tend to get less attention within the POPL community: Rust's governance and open development process, and design considerations around language and library evolution. Finally, it will mention a few of the myriad open research questions around Rust.

Categories and Subject Descriptors D.3.3 [**Programming Languages**]: Language Constructs and Features;

Keywords Substructural types; ownership; uniqueness; concurrency; typeclasses

Biography

Aaron Turon is Research Engineering Manager for the Rust team at Mozilla. He received his PhD from Northeastern University, where he studied programming language design, program verification, and low-level concurrency. His dissertation was awarded the SIGPLAN John C. Reynolds Doctoral Dissertation Award in 2014. After his PhD studies, he continued his research in concurrency verification and programming techniques as a postdoc at MPI-SWS. He joined Mozilla in 2014, and has played an active role in Rust's development since then.

POPL'17, January 15–21, 2017, Paris, France
ACM. 978-1-4503-4660-3/17/01...$15.00
http://dx.doi.org/10.1145/3009837.3011999