Synthesis of Safe Pointer-Manipulating Programs

Nadia Polikarpova
University of California in San Diego, La Jolla, CA, USA

Abstract

Low-level pointer-manipulating code is ubiquitous in operating systems, networking stacks, and browsers, which form the backbone of our digital infrastructure. Unfortunately, this code is susceptible to many kinds of bugs, which lead to crashes and security vulnerabilities. A promising approach to eliminating bugs and reducing programmer effort at the same time is to use *program synthesis* technology to generate provably correct low-level code automatically from high-level specifications.

In this talk I will present a program synthesizer SuSLik, which accepts as input a specification written in *separation logic*, and produces as output a provably correct C program. SuSLik is the first synthesizer capable of generating a wide range of operations on linked data structures (such as singly- and doubly-linked lists, binary trees, and rose trees) without additional hints from the user. It is also the first synthesizer to automatically discover recursive auxiliary functions required for nested data structure traversal. To make this possible, SuSLik relies on a novel proof system – *synthetic separation logic* – to derive correct-by-construction programs directly from their specifications. Program proofs generated by SuSLik can be automatically translated into three foundational verification frameworks embedded in Coq: Hoare Type Theory (HTT), Iris, and Verified Software Toolchain (VST).

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Automated reasoning; Software and its engineering → Automatic programming

Keywords and phrases Program Synthesis, Separation Logic, Proof Search

Digital Object Identifier 10.4230/LIPIcs.ITP.2021.2

Category Invited Talk

Funding *Nadia Polikarpova*: This work was supported by the National Science Foundation under Grant No. 1911149.