A Program Logic for Automated P4 Verification

Nate Foster[1,2], Cole Schlesinger[1], Robert Soulé[1,3], and Han Wang[1]

[1] Barefoot Networks
[2] Cornell University
[3] Università della Svizzera italiana

1 Abstract

Programs written in the P4 language can *go wrong* in a variety of interesting and frustrating ways:

- Reading uninitialized variables may corrupt outgoing packets;
- Unhandled exceptions in parsers may cause packets to be silently dropped;
- Indexing out of the bounds of a header stack may produce an undefined result.

But unlike other languages, P4 is designed for network programming and contains first-class notions of parsers, tables, actions, and controls. In particular, a P4 program is essentially a skeleton that describes the structure of a packet processing pipeline, but not its content—that is filled in later, by the control plane. Hence, errors in P4 programs arise in two different ways: unconditionally, when a P4 program directly implements erroneous behavior, and conditionally, when incorrect behavior emerges due to the rules installed by the control plane.

In this talk, we show how a Hoare-style program logic, combined with an SMT solver, can automatically determine whether a P4 program will go unavoidably wrong and also generate a set of constraints for each table to determine whether the program will go conditionally wrong during network operation. Along the way, we develop a formal semantics of P4 by way of translation to a simple imperative language with functions, and we report on our experiences analyzing several existing programs.

2 Presenter Bio

Cole Schlesinger is a Research Engineer at Barefoot Networks, where his work focuses on P4 language design, compilation, and verification. In the past, he has also worked on languages and abstractions for software-defined networks, control and data plane verification, type systems and inference algorithms, and formal semantics. Cole received a PhD in Computer Science from Princeton University and a BS in Computer Science from Ithaca College. His industrial experience includes positions as a Research Scientist at Samsung Research America and a Software Engineer at GrammaTech, Inc.