# Trace Logic for Automating Loop Reasoning

## (Invited Talk)

Laura Kovacs
Vienna University of Technology, Austria
lkovacs@forsyte.tuwien.ac.at

One of the main challenges in automating software verification comes with handling inductive reasoning over programs containing loops. Until recently, automated reasoning in formal verification was the primary domain of SMT solvers, yielding powerful advancements for inferring and proving loop properties with linear arithmetic and limited use of quantifiers. Formal verification however also requires reasoning about unbounded data types, such as arrays, and inductively defined data types. In this talk we address such and similar aspects of first-order reasoning in formal verification. We describe trace logic, an instance of many-sorted first-order logic, to automate the partial correctness verification of program loops. We express program semantics in trace logic and use trace logic in combination with superposition-based first-order theorem proving. We adjust consequence finding in first-order theorem proving to both generate and prove system properties as logical consequences of trace logic. We further guide and automate inductive loop reasoning in trace logic by using generic trace lemmas capturing inductive loop invariants. Our initial experiments show that automated reasoning in trace logic allow us to prove correctness of many programs whose functional behavior are best summarized in the first-order theories of linear integer arithmetic, arrays and inductive data types.

This is a joint work with Pamina Georgiou and Bernhard Gleiss.

Submitted to:
VPT 2021