Theory Exploration:
Conjecturing, Testing and Reasoning about Programs

(Invited Talk)

Moa Johansson
Chalmers University, Gothenburg, Sweden
moa.johansson@chalmers.se

Theory Exploration is a technique for automatically discovering (and proving) interesting properties about programs. This has successfully been used for automation of inductive proofs, where theory exploration can be used to discover auxiliary lemmas needed in all but the simplest inductive proofs. A richer background theory, consisting of additional equational properties, is constructed automatically, allowing harder theorems to be proved automatically. I believe theory exploration can also have applications in program transformation and I look forward to discuss such possibilities at this workshop.

We have developed several systems for theory exploration, the most recent ones being QuickSpec and Hipster. QuickSpec is a tool written in Haskell concerned with the task of efficiently conjecturing candidate properties about functional programs, using term generation and property based testing. It produces a set of candidate properties, which can either be directly presented to the user, or passed to an automated theorem prover. Hipster is our link to such a theorem prover, and is built on top of the proof assistant Isabelle/HOL. Hipster communicates with QuickSpec, and will attempt to automatically prove candidate properties, possibly using other discovered properties as lemmas. Naturally, we only want to present "interesting" properties to the user, not swamping the output with trivial equations. But how do we define what is to be considered interesting? In Hipster, the user can control this by configuring the system with two reasoning strategies: If the system can prove something by the "easy reasoning strategy" (e.g. simplification) we may discard it as uninteresting, while properties requiring proof by the "hard reasoning strategy" (e.g. induction) are presented to the user.

I will also mention some of our ongoing work, where we consider extensions of theory exploration for very large theories (or programs), which otherwise are computationally expensive.

Submitted to:
VPT 2020