Report
Scottish Theorem Proving Seminar Meeting
Heriot-Watt University, March 13, 2015

Gudmund Grov and Manuel Maarek
2015-03-30

Scottish Theorem Proving Seminars

Theorem proving research is notably strong in Scottish universities, with active groups and researchers across many SICSA institutions; including Dundee, Edinburgh, Glasgow, Heriot-Watt, Stirling and Strathclyde. The Scottish Theorem Proving (STP) Seminar Series, which has been running for 18 years, provides a common venue for communication and sharing of ideas by all these researchers. See:

http://www.macs.hw.ac.uk/stp/

The first STP meeting in 2015 was hosted by Manuel Maarek and Gudmund Grov at Heriot-Watt University on Friday March 13. See:

http://www.macs.hw.ac.uk/stp/1503.html

Attendance list

Scottish PhD students as well as academics and senior academics from across Scotland attended the meeting.

1. Rajiv Murali (PhD student, Heriot-Watt University)
2. Mike Just (Associate Professor, Heriot-Watt University)
3. Colin Farquhar (PhD student, Heriot-Watt University)
4. Roy Dyckhoff (Honorary Senior Lecturer, University of St Andrews)
5. Lavinia Burski (PhD student, Heriot-Watt University)
6. Peng Fu (PhD student, Dundee University)
7. Ekaterina Komendantskaya (Senior Lecturer, University of Dundee)
8. James McKinna (Senior Research Fellow, University of Edinburgh)
9. Andriana Gkaniatsou (PhD student, University of Edinburgh)
10. Gudmund Grov (Assistant Professor, Heriot-Watt University)
11. Manuel Maarek (Assistant Professor, Heriot-Watt University)

Discussions and presentations

The STP meeting took place in the afternoon of Friday March 13, 2015. Discussions among participants took places around three presentations.

Horn-formulas as Types for Structural Resolution

Peng Fu (University of Dundee)

This is a joint work with Ekaterina Komendantskaya.
First-order logic programming traditionally relies on SLD-resolution, in which termination of derivations is crucial for deciding entailment, whereas the exact proof content of derivations may be opaque. In Coalgebraic Logic Programming (CoALP), derivations can proceed infinitely (or corecursively), as long as intermediate proof structures can be finitely observed. We propose a type-theoretic semantics for this new kind of structural resolution. We formalize several derivation strategies possible in logic programming (SLD-resolution, resolution by term-matching, structural resolution), and show that they are sound w.r.t. the type system we propose. We also establish an exact formal relation between structural resolution and SLD-resolution within this typed framework.

Logging-in Insecurely: Reverse-Engineering the Smart-Card Communication Layer

Andriana Gkaniatsou (University of Edinburgh)

Smart-cards are seen as one of the most secure, tamper-proof, and trusted devices for implementing confidential operations, such as secure log-in, for financial, communication, security and data management purposes.
These operations typically involve communication between smart-cards and third-party systems. Such communication must be secure, and developers usually prefer proprietary implementations which create the illusion of security as hide the card's code.
In this talk we will present REPROVE, a first-order based system that reverse-engineers the communication trace and deduces the card's functionalities. REPROVE does not require access to the card, and deals with both inter-industry and proprietary implementations.

Improving Predictability, Efficiency and Trust of Model-Based Proof Activity

Manuel Maarek (Heriot-Watt University)

This is a joint work with Jean-Frédéric Etienne, Florent Anseaume and Véronique Delebarre from SafeRiver which we will present at ICSE SEIP 2015.
We report on our industrial experience in using formal methods for the analysis of safety-critical systems developed in a model-based design framework. We first highlight the formal proof workflow devised for the verification and validation of embedded systems

developed in Matlab/Simulink. In particular, we show that there is a need to: determine the compatibility of the model to be analysed with the proof engine; establish whether the model facilitates proof convergence or when optimisation is required; and avoid over-specification when specifying the hypotheses constraining the inputs of the model during analysis. We also stress on the importance of having a certain harness over the proof activity and present a set of tools we developed to achieve this purpose. Finally, we give a list of best practices, methods and any necessary tools aiming at guaranteeing the validity of the verification results obtained.