Typing, Analysis, and Verification of Heap-Manipulating Programs
Executive Summary

Mooly Sagiv, Arnd Poetzsch-Heffter, and Peter O'Hearn

[1] Tel Aviv University, IL
[2] University of Kaiserslautern, DE
[3] University of London, GB

1 Topic and General Goal

Most of today's software is written in procedural or object-oriented programming languages. Many of these programs make use of heap-allocated data. This is in particular true for object-oriented programs. Thus, analysis and verification techniques for heap-manipulating programs are crucial to avoid and find errors, to optimize implementations, and to verify properties in a huge class of modern software.

The heap has been a major obstacle to more widespread use of verification and analysis for real-world code. In the last ten years, though, research on analysis and verification for heap-manipulating programs has progressed significantly, in work mainly done by three research communities:

1. Ownership and region types for structuring object heaps, for alias control, and for encapsulation. The main idea is to restrict the way pointers are manipulated and/or restrict the shape of the heap.
2. Verification of heap manipulating programs. The main idea is to specify interesting properties of such programs and to develop formal methods for checking if the specifications are met by the program.
3. Static program analysis for heaps. The main idea is to automatically infer properties of programs. For example, many algorithms infer the shape of the heap at various program points.

Dagstuhl Seminar Proceedings 09301
Typing, Analysis and Verification of Heap-Manipulating Programs
http://drops.dagstuhl.de/opus/volltexte/2010/2435

The central purpose of this Dagstuhl seminar was to bring together top researchers from these three different communities and to investigate the synergies that can result from a combination of the techniques developed by these communities.

2 Participants and Organization

The seminar had 41 participants with a good distribution over the three research communities mentioned above. We were in particularly happy to have a good number of excellent young researchers as participants.

After the Monday morning sessions where each participant gave a short statement of his/her background and interest, we started with four overview talks covering the central topics and views of the different communities:

- Peter Müller: Ownership based types
- K. Rustan M. Leino: Comparing heap models: Ownership, dynamic frames, permissions
- Greta Yorsh: Shape analysis overview
- Viktor Kuncak: Theorem provers and decision procedures

The rest of the seminar was structured into research presentations (31 talks), presentation of challenge problems (three problems were presented and discussed), and discussions on how to exploit potential synergies of the different techniques (see below).

Social events To further foster the communication among the participants we had a hot summer hike in the natural environment of Dagstuhl and a special dinner.

3 Remarks on synergies

Ownership type information can be useful to static analyses and deductive verification. Analysis techniques can support type inference, allow generalizing type systems, and can automatically provide information for verification frameworks. Heap structuring techniques used in verification frameworks, like in separation logic, can be helpful to modularize static analyses. Besides combination of the techniques, another dimension of integration is given by the properties of interest such as, e.g., alias control, access modes, encapsulation, heap structure properties, and behavioral interface properties.

Often these properties have to be analyzed together. E.g., certain heap analyses can only be applied in a modular way if the program satisfies some encapsulation restrictions. Also, programs that satisfy ownership requirements may be amenable to more efficient program analysis. A good witness of the close relation between functional and structural properties is separation logic.