Bachelor Thesis „Debugging by Interactive Heap Navigation“
(joint project: Prof. Martin Vechev and Prof. Otmar Hilliges)

Overview

The overall goal of this project is to explore novel ways to aid programmers in understanding and debugging data heavy algorithms and programs. During the thesis the student will build a system which combines advanced program analysis (e.g., shape analysis) with novel touch based user interfaces for the inspection and manipulation of a running program.
Objectives

This project has three main objectives:

1. Develop a clean-state debugging system where the user can invoke various concretization and abstraction heap operations using gestures performed via direct touch and pen interaction.

2. Design and implement new data structure layout algorithms which will enable laying out the data structure in a way which admits as few changes to the physical layout as possible (e.g., upon changing an edge in the graph). This will enable smooth "abstract debugging". The algorithms need to be working in real-time and there should be no noticeable delay by the user.

3. Explore the notion of "abstract breakpoints" where the developer sketches an abstract heap shape and the debugger stops the execution where a concretization of that heap shape is encountered.

Work Steps

1. Get familiar with the existing system, both the code and the paper [1].

2. Implement the key operations described in the paper [1].

3. Implement an appropriate layout algorithm described in [2&3]. Most likely a variation of existing graph layout algorithms will need to be implemented as most graph drawing algorithms do not take temporal changes of the graph into consideration.

4. Implement the abstract breakpoint mechanism

5. Experiment with the system on the algorithms in [3]
References

1. **An Interactive System for Data Structure Development**, Jibin Ou, Otmar Hilliges, Martin Vechev, ACM CHI 2015


3. **Introduction to Graph Layout / Drawing algorithms**: [http://cs.brown.edu/~rt/gdhandbook/](http://cs.brown.edu/~rt/gdhandbook/)


Grading Scheme

- 4.0: Full, clean implementation of the paper
- 4.5: As above + abstract breakpoints working
- 5.0: As above + implementation of a layout algorithm
- 5.5: As above + complete working implementation showing the system works on real examples.
- 6.0: As above + new variant of a layout algorithm working in our setting.