Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Lars Hupel

WS 2017/18
23.01.2018

Semantics of Programming Languages

Exercise Sheet 13

Exercise 13.1 Sign Analysis

Instantiate the abstract interpretation framework to a sign analysis over the lattice $pos, zero, neg, any$, where *pos* abstracts positive values, *zero* abstracts zero, *neg* abstracts negative values, and any abstracts any value.

datatype *sign* = *Pos* | *Zero* | *Neg* | *Any*

Exercise 13.2 AI for Conditionals

Our current constant analysis does not regard conditionals. For example, it cannot figure out, that after executing the program *x*:=*2*; *IF x<2 THEN x*:=*2 ELSE x*:=*1*, *x* will be constant.

In this exercise, we extend our abstract interpreter with a simple analysis of boolean expressions. To this end, modify locale *Val_semilattice* in theory *Abs_Int0.thy* as follows:

- Introduce an abstract domain *'bv* for boolean values, add, analogously to *num'* and *plus'* also functions for the boolean operations and for *less*.
- Modify *Abs_Int0* to accommodate for your changes.

General homework instructions

The first homework is pen & paper (or keyboard & text file). You have the choice of uploading a text file or a PDF scan of hand-written notes to the submission system. Physical paper submissions are not accepted.

Homework 13.1 Parity analysis

Submission until Tuesday, January 30, 2018, 10:00am.

Consider the following program:

```
r := 11;
a := 11 + 11;
WHILE b DO
  r := r + 1;
  a := a - 2;
r := a + 1
```

Add annotations for parity analysis to this program, and iterate on it the *step'* function until a fixed point is reached. (More precisely, let C be the annotated program; you need to compute $(step' \top)^0\ C$, $(step' \top)^1\ C$, $(step' \top)^2\ C$, etc.). Document the results of each iteration in a table.

Homework 13.2 Parity Analysis (II)

Submission until Tuesday, January 30, 2018, 10:00am.

Change the parity analysis from theory *Abs_Int1_parity* to analyze the remainder modulo 3. Instantiate the locales *Val_semilattice* and *Abs_Int* (you can ignore everything after "Termination" in *Abs_Int1_parity*).

Apply your analysis to a non-trivial example program (should contain at least one loop with a non-constant assignment).

Note: Keep in mind that the abstract interpretation has to consider "mixed states", i.e., a value could be divisible by 0 or 1 at a particular point.