Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D., Dr. Peter Lammich
Johannes Hölzl

WS 2014/15
27. 01. 2015

Semantics of Programming Languages

Exercise Sheet 13

The following exercises are typical exam exercises. You are supposed to solve them on a sheet of paper, without using Isabelle/HOL.

Exercise 13.1 Verification Condition Generation

Consider the following While-program c:

```
a := x;
WHILE 1 < a DO
  a := a - 2
```

Your task is to show that:

$$\models \{x \geq 0\}\ c\ \{a = 0 \implies \mathsf{even}\ x\}$$

Find an invariant for the loop. Let C be the annotated program, and $Q := \{a = 0 \implies \mathsf{even}\ x\}$ be the postcondition. Which proof obligations result when using the verification condition generator? What do $vc\ C\ Q$ and $pre\ C\ Q\ s$ look like?

Exercise 13.2 Parity analysis

Now consider the following While-program:

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

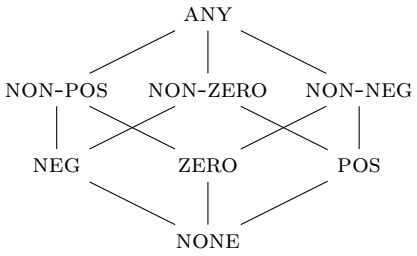
Add annotations for parity analysis to this program, and iterate on it the $step'$ function until a fixed point is reached. (More precisely, let C be the annotated program; you need to compute $(step'\ \top)^0\ C$, $(step'\ \top)^1\ C$, $(step'\ \top)^2\ C$, etc.). Document the results of each iteration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

Exercise 13.3 Abstract Interpretation: Sign Analysis

Design an abstract domain for *sign analysis*: For each program variable, this will calculate which signs (positive, negative, or zero) it could possibly have.

Note: If a proof is done by exhaustive case distinctions that are all proven analogously, it is enough to spell out one of the cases.

The domain of the analysis is given as follows:



Specify the concretization function γ_sign, and the abstract operations *num_sign* and *plus_sign*. Note that your analysis should be as precise as possible, i.e., γ_sign should not return too big concrete sets, and the abstract operations should not return too imprecise abstract values.

Now show that you actually defined an abstract domain, i.e.

- The concretization function is monotone,
- the concretization function maps ANY to $UNIV$,
- the abstract operations correctly implement the concrete ones.

Note: In the Isabelle-formalization, these are the assumptions of locale *Val_semilattice*.

Finally, you must define a measure function on the abstract domain, which can be used to prove that the analysis always terminates. Define a function *m_sign* from the sign domain into the natural numbers such that

- $x < y \Longrightarrow m\_sign\ x > m\_sign\ y$
- $m\_sign\ x \leq h$

where h is the height of the sign domain.