Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Simon Wimmer

WS 2016/17
31. 1. 2017

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.

For this exercise, you best modify the parity analysis *src/HOL/IMP/Abs_Int1_parity*.

Exercise 13.2 Complete Lattices: GLB of UBs is LUB

Show that the least upper bound is the greatest lower bound of all upper bounds:

definition *"Sup' (S::'a::complete_lattice set) ≡ Inf {u. ∀ s∈S. s≤u}"*

lemma *Sup'_upper*: *"∀ s∈S. s ≤ Sup' S"*
lemma *Sup'_least*:
  assumes *upper*: *"(∀ s∈S. s≤u)"*
  shows *"Sup' S ≤ u"*

Homework 13.1 Lattice Theory

Submission until Tuesday, 07.02.2017 (Pen & Paper), 10:00am.

A type $'a$ is a $\sqcup$-semilattice if it is a partial order and there is a supremum operation $\sqcup$ of type $'a \Rightarrow {}'a \Rightarrow {}'a$ that returns the least upper bound of its arguments:

- Upper bound: $x \leq x \sqcup y$ and $y \leq x \sqcup y$
- Least: $x \leq z \wedge y \leq z \longrightarrow x \sqcup y \leq z$

Is every finite $\sqcup$-semilattice with a bottom element $\bot$ also a complete lattice? Prove or disprove on paper!

Homework 13.2 AI for the Extended Reals

Submission until Tuesday, 07.02.2017 (Isabelle), 10:00am. For this exercise, we will consider a modified variant of IMP that computes on real numbers extended with $-\infty$ and ∞. The corresponding type is *ereal*. We will consider "$-\infty + \infty$" and "$\infty + (-\infty)$" erroneous computations. We propagate errors by using the *option* type, i.e. we set *val = ereal option*. The files *AExp.thy*, *BExp.thy*, *BigStep.thy*, *Collecting.thy* for this variant are provided for you on the website. Your task is now to design an abstract interpreter on the domain consisting of subsets of $\{\infty^-, \infty^+, NaN, Real\}$ where *NaN* signals a computation error and all other values have their obvious meaning. First adopt *Abs_Int0.thy* and *Abs_Int1.thy* to accommodate for the changed semantics, and then instantiate the abstract interpreter (*Abs_Int*, *Abs_Int_mono*, and *Abs_Int_measure*) with your analysis. For this step you best modify the parity analysis *Abs_Int1_parity.thy*.

Hints: To benefit from proof automation it can be helpful to slightly change the format of the rules for addition in *Val_semilattice*. For instance, you could reformulate *gamma_plus'* as: $i1 \in \gamma\ a1 \Longrightarrow i2 \in \gamma\ a2 \Longrightarrow i = i1 + i2 \Longrightarrow i \in \gamma(plus'\ a1\ a2)$. (You will need to change the interface *Val_semilattice*).

You can start the formalization of the AI like this:

datatype *bound* = *NegInf* ("∞^-") | *PosInf* ("∞^+") | *NaN* | *Real*

datatype *bounds* = *S* "*bound set*"
instantiation *bounds* :: *order*
begin
definition *less_eq_bounds* **where**
"$x \le y = (case\ (x, y)\ of\ (S\ x, S\ y) \Rightarrow x \subseteq y)$"
definition *less_bounds* **where**
"$x < y = (case\ (x, y)\ of\ (S\ x, S\ y) \Rightarrow x \subset y)$"