

Semantics of Programming Languages

Exercise Sheet 13

Exercise 13.1 Abstract Interpretation 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_abs* in theory *Abs_Int0.thy* as follows:

- Define, analogously to *plus'*, a function *less'* :: *'av ⇒ 'av ⇒ bool option* that approximates less expressions: *Some b* means, the result is definitely *b*, and *None* means unknown. Insert also an appropriate assumption *gamma_less'* to the locale.
- Do not modify the non-executable locale *Abs_Int_Fun* in *Abs_Int0.thy*. Simply comment it out, it is not needed for the executable analysis. The same holds for the locale *Abs_Int_Fun_mono*.
- Define a function *bval'* :: *bxp ⇒ 'av st ⇒ bool option* in *Abs_Int1.thy* (analogous to *aval'*), and show a lemma *bval_sound* (analogous to *aval'_sound*).
- Modify the step'-function in locale *Abs_Int*, to accomodate for known boolean values. It's enough to modify the step for IF, leave WHILE unchanged! Hint: If you know that a branch of the IF is non executable, let the step-function pass the annotation *None* into this branch, which is the abstraction for the empty set of states.
- Finally, adapt all theories necessary to get a more precise constant analysis.

Hint: Start with a fresh copy of the IMP/ folder. Make sure to download a recent version, as we changed some theories on Jan 21.

Homework 13 Parity Analysis for Multiplication

Submission until Tuesday, 29. 1. 2013, 10:00am. Add multiplication to arithmetic expressions, and adapt all theories necessary to get parity analysis working.

Submit this homework as archive file, named `FirstLastname.{tgz,zip}`. The archive should contain all required files in its root-directory. Please no unusual formats like rar!

Hint: Start with a fresh copy of the IMP/ folder, modify AExp accordingly, and then try to compile `Abs_Int1_parity` and adapt all places where proofs fail. Make sure to download a recent version, as we changed some theories on Jan 21.