Static Program Analysis

http://proglang.informatik.uni-freiburg.de/teaching/programanalysis/2014ss/

Exercise Sheet 9

17.07.2014

Abstract interpretation

Exercise 1 (Widening operators)
Show that the operator ∇ on **Interval** with

$$\bot \nabla X = X \nabla \bot = X$$

and

$$[i_1, j_1]\nabla[i_2, j_2] = [\text{ if } i_2 < i_1 \text{ then } -\infty \text{ else } i_1, \text{ if } j_2 > j_1 \text{ then } +\infty \text{ else } j_1]$$

is a widening operator. First, state precisely what you need to show, and then show that these properties are indeed fulfilled.

Exercise 2 (Abstractions)
Let S be the set of strings over a (finite) alphabet Σ. An abstraction of the string is the set of characters/symbols of which the string is built. Example: `Program analysis` is abstracted by {`P,r,o,g,a,m, ' ',n,l,y,s,i`}.

Specify the details of the Galois connection $(\mathcal{P}(S), \alpha, \gamma, \mathcal{P}(\Sigma))$ formally. Is this Galois connection also a Galois insertion (also called *Galois surjection* on the slides "Abstraction III")?

Exercise 3 (Galois insertions)
Let $(L_1, \alpha_1, \gamma_1, M_1)$ and $(L_2, \alpha_2, \gamma_2, M_2)$ be *Galois insertions* (*Galois surjection*). First define

$$\begin{aligned}\alpha(l_1, l_2) &= (\alpha_1(l_1), \alpha_2(l_2)) \\ \gamma(m_1, m_2) &= (\gamma_1(m_1), \gamma_2(m_2))\end{aligned}$$

and show that $(L_1 \times L_2, \alpha, \gamma, M_1 \times M_2)$ is a Galois insertion. Then define

$$\begin{aligned}\alpha(f) &= \alpha_2 \circ f \circ \gamma_1 \\ \gamma(g) &= \gamma_2 \circ g \circ \alpha_1\end{aligned}$$

and show that $(L_1 \to L_2, \alpha, \gamma, M_1 \to M_2)$, where $L_1 \times L_2$ and $M_1 \times M_2$ are *Monotone Function Spaces* (see book on p. 398), is a Galois insertion.

Control Flow Analysis

Exercise 4 (Analyzing a program by hand)
Consider the following program:

let $f = \text{fn } y \Rightarrow y$ in
    let $g = \text{fn } x \Rightarrow f$ in
        let $h = \text{fn } v \Rightarrow v$ in
            g (g h)

Add labels to the program, and guess an analysis result. Use Table 3.1 in the book, p. 146, to verify that it is indeed an acceptable guess.

Enhancing the analysis

Modify the Control Flow Analysis of Table 3.1. to take account of the left to right evaluation order imposed by a call-by-value semantics: In the clause [*app*] there is no need to analyze the operand if the operator cannot produce any closures. Try to find a program where the modified analysis accepts a result which is rejected by Table 3.1.

Submission In PDF format via email to `geffken AT informatik.uni-freiburg.de`. Please name your single file with the scheme: `ex09-name.pdf`.

- Deadline: **24.07.2014, 12:00**
- Late submissions will not be marked.
- Do not forget to write your name on the exercise sheet.