Lecture: Program analysis
Exercise 8
http://proglang.informatik.uni-freiburg.de/teaching/programanalysis/2010ss/

Abstract interpretation

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.

Solution

- ∇ is an upperbound operator: Let $l_1 = [i_1, j_1], l_2 = [i_2, j_2]$.

$$\begin{array}{ll} i_2 < i_1, j_2 > j_1: & l_1 \sqsubseteq [-\infty, +\infty] \sqsupseteq l_2 \\ i_2 < i_1, j_2 \leq j_1: & l_1 \sqsubseteq [-\infty, j_1] \sqsupseteq l_2 \\ i_2 \geq i_1, j_2 > j_1: & l_1 \sqsubseteq [i_1, +\infty] \sqsupseteq l_2 \\ i_2 \geq i_1, j_2 \leq j_1: & l_1 \sqsubseteq [i_1, j_1] \sqsupseteq l_2 \end{array}$$

- For all ascending chains $(l_n)_n$, the ascending chain $l_0, l_0 \nabla l_1, (l_0 \nabla l_1) \nabla l_2, \ldots$ eventually stabilizes.
  For an arbitrary element $l_0 = [n, m]$, we have to consider the following cases for $l_1 = [k, l]$:

$$\begin{array}{lll} k < n, l > m & \Rightarrow & l_0 \nabla l_1 = [-\infty, +\infty] \\ k = n, l > m & \Rightarrow & l_0 \nabla l_1 = [n, +\infty] \\ k < n, l = m & \Rightarrow & l_0 \nabla l_1 = [-\infty, m] \\ k = n, l = m & \Rightarrow & l_0 \nabla l_1 = [n, m] \end{array}$$

  Hence, if the chain $(l_n)_n$ eventually stabilizes, then so will the chain $(l_i^\nabla)_i$. Otherwise, it converges to the upper bound $[-\infty, +\infty]$.

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?

Solution

Let Σ_s be the set of all of the letters that occur in a particular string. We define the abstraction and concretisation function as follows:

$$\begin{array}{lll} \alpha(S) & = & \bigcup \{\Sigma_s \mid s \in S\} \\ \gamma(\sigma) & = & \{s \mid \Sigma_s \subseteq \sigma\} \end{array}$$

α and γ are clearly monotone. Further, for a set of strings $S = \{s_1, \ldots, s_n\}$:

$$\gamma(\alpha(S)) = \gamma(\cup\{\Sigma_s \,|\, s \in S\}) = \{s' \,|\, \Sigma_{s'} \subseteq \cup\{\Sigma_{s'} \,|\, s \in S\}\} \supseteq S$$

and

$$\alpha(\gamma(\sigma)) = \alpha(\{s \,|\, \Sigma_s \subseteq \sigma\}) = \bigcup\{\Sigma_s \,|\, s \in \{s \,|\, \Sigma_s \subseteq \sigma\}\} = \sigma$$

Therefore, the Galois connection is also a Galois insertion.

3 Galois insertions

Let $(L_1, \alpha_1, \gamma_1, M_1)$ and $(L_2, \alpha_2, \gamma_2, M_2)$ be Galois insertions. 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 $(L_1 \to L_2, \alpha, \gamma, M_1 \to M_2)$ is a Galois insertion.

Solution

We have to show that α and γ are monotone, and that

$$\begin{aligned} \gamma \circ \alpha &\sqsupseteq \lambda l.l \\ \alpha \circ \gamma &= \lambda m.m \end{aligned}$$

1. α and γ are monotone, because $\alpha_1, \alpha_2, \gamma_1$, and γ_2 are monotone. Further, let $l = (l_1, l_2) \in L_1 \times L2$.

$$l \sqsubseteq \gamma(\alpha(l)) \quad \Leftrightarrow \quad l_1 \sqsubseteq \gamma(\alpha(l_1)) \text{ and } l_2 \sqsubseteq \gamma(\alpha(l_2))$$

   This holds because $(L_1, \alpha_1, \gamma_1, M_1)$ and $(L_2, \alpha_2, \gamma_2, M_2)$ are Galois insertions. Similarly, for $(m_1, m_2) \in M_1 \times M_2$, we have

$$m = \alpha(\gamma(m)) \quad \Leftrightarrow \quad m_1 = \alpha(\gamma(m_1)) \text{ and } m_2 = \alpha(\gamma(m_2))$$

2. For the first part, consider the Monotone Function Space in the book on p. 398. It remains to show that $\alpha(\gamma(f)) = f$ for $f \in M_1 \to M_2$:

$$\alpha(\gamma(f)) = \alpha(\gamma_2 \circ f \circ \alpha_1) = \alpha_2 \circ \gamma_2 \circ f \circ \alpha_1 \circ \gamma_1 = id \circ f \circ id = f$$

4 Types and Effects

Consider the following FUN program:

```
new_A x := 1 in
new_B y := 9 in
let f = fn z => x := !y in
let g = fn z => x := 8 in
let h = fn z => !x in
(fn w => w f + w h) (fn v => v 4)
```

What is the result of evaluating this program? What are the types and effects for the functions in this program?

Solution

The program evaluates to 18.

`fn z => x := !y`	$int \xrightarrow{\{A{:=},!B\}} int$
`fn z => x := 8`	$int \xrightarrow{\{A{:=}\}} int$
`fn z => !x`	$int \xrightarrow{\{!A\}} int$
`fn w => w f + w h`	$(int \xrightarrow{\{A{:=},!A,!B\}} int) \xrightarrow{\{A{:=},!A,!B\}} int$
`fn v => v 4`	$(int \to int) \to int$

5 Control Flow Analysis in a Type and Effect System

The type and effect system for Control Flow Analysis in Chapter 5.1. uses annotations ϕ to denote the set of function definitions that can result in a function of a given type.

Extend the analysis with annotations for the base type `bool` to denote the set of constants that may be the result of evaluating the expression of a respective type.

Solution

We extend the annotations to also include boolean constants:

$$\phi ::= \{\texttt{tt}\} | \{\texttt{ff}\} | \cdots$$

And we now also annotate the boolean type with some effect:

$$\hat{\tau} ::= \texttt{bool}_{\phi} | \ldots$$

Then, we can adapt the rules for the Control Flow Analysis as shown:

$$[const_1] \qquad \hat{\Gamma} \vdash \texttt{true} : \texttt{bool}_{\{\texttt{tt}\}}$$

$$[const_2] \qquad \hat{\Gamma} \vdash \texttt{false} : \texttt{bool}_{\{\texttt{ff}\}}$$

$$[if_1] \qquad \frac{\hat{\Gamma} \vdash e_0 : \texttt{bool}_{\{\texttt{tt}\}} \qquad \hat{\Gamma} \vdash e_1 : \hat{\tau}_1 \qquad \hat{\Gamma} \vdash e_2 : \hat{\tau}_2 \qquad \lfloor \hat{\tau}_1 \rfloor = \lfloor \hat{\tau}_2 \rfloor}{\hat{\Gamma} \vdash \texttt{ if } e_0 \texttt{ then } e_1 \texttt{ else } e_2 : \hat{\tau}_1}$$

$$[if_2] \qquad \frac{\hat{\Gamma} \vdash e_0 : \texttt{bool}_{\{\texttt{ff}\}} \qquad \hat{\Gamma} \vdash e_1 : \hat{\tau}_1 \qquad \hat{\Gamma} \vdash e_2 : \hat{\tau}_2 \qquad \lfloor \hat{\tau}_1 \rfloor = \lfloor \hat{\tau}_2 \rfloor}{\hat{\Gamma} \vdash \texttt{ if } e_0 \texttt{ then } e_1 \texttt{ else } e_2 : \hat{\tau}_2}$$

$$[and_1] \qquad \frac{\hat{\Gamma} \vdash e_1 : \texttt{bool}_{\{\texttt{tt}\}} \qquad \hat{\Gamma} \vdash e_2 : \texttt{bool}_{\{\texttt{tt}\}}}{\hat{\Gamma} \vdash e_1 \,\&\&\, e_2 : \texttt{bool}_{\{\texttt{tt}\}}}$$

All other rules are adapted similarly or remain unchanged.