Static Program Analysis

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

Solution Sheet 9

24.07.2014

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.

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 $l_1 = [k, l]$:

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

  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 $[-\infty, +\infty]$.

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")?

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:

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

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

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

and

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

Therefore, the Galois connection is also a Galois insertion.

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 surjections*). First define

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

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

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

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.

Solution

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

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

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. Consider the Monotone Function Space in the book on p. 398.

   First, we observe that α and γ are monotone because α_2 and γ_2 are. The detailed reasoning for α is as follows (the same reasoning applies to γ):

   $$\begin{array}{l} \quad f \sqsubseteq f' \\ \Longrightarrow \forall x : f(x) \sqsubseteq f'(x) \\ \Longrightarrow \forall x : \alpha_2 \circ f(x) \sqsubseteq \alpha_2 \circ f'(x) \text{ (because } \alpha_2 \text{ is monotone)} \\ \Longrightarrow \forall y : \alpha_2 \circ f \circ \gamma_1(y) \sqsubseteq \alpha_2 \circ f' \circ \gamma_1(y) \\ \Longrightarrow \alpha_2 \circ f \circ \gamma_1 \sqsubseteq \alpha_2 \circ f' \circ \gamma_1 \\ \Longrightarrow \alpha(f) \sqsubseteq \alpha(f') \end{array}$$

Next, we show that $\gamma(\alpha(f)) = f$ for $f \in M_1 \to M_2$ and calculate

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

using the monotonicity of f and $\gamma_{\{1,2\}} \circ \alpha_{\{1,2\}} \sqsupseteq \lambda l.l = \mathrm{id}$.

It remains to show that $\alpha(\gamma(f)) = f$ for $f \in M_1 \to M_2$:

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

We have used $\alpha_{\{1,2\}} \circ \gamma_{\{1,2\}} = \lambda l.l = \mathrm{id}$.

Control Flow Analysis

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

let f = fn $y \Rightarrow y$ in
    let g = fn $x \Rightarrow f$ in
        let h = 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.

Solution

When adding labels, the program is given by:

$$\Big(\text{let } f = (\text{fn } y \Rightarrow y^1)^2 \text{ in}$$
$$\Big[\text{let } g = (\text{fn } x \Rightarrow f^3)^4 \text{ in}$$
$$\Big(\text{let } h = (\text{fn } v \Rightarrow v^5)^6 \text{ in}$$
$$[g^7(g^8\ h^9)^{10}]^{11}\Big)^{12}\Big]^{13}\Big)^{14}$$

A solution might be:

	$(\widehat{C}, \widehat{\rho})$
1,5	$\emptyset$
2,3	{fn y $\Rightarrow y^1$}
4,7,8	{fn x $\Rightarrow f^3$}
6,9	{fn v $\Rightarrow v^5$}
10,11,12,13,14	{fn y $\Rightarrow y^1$}
f	{fn y $\Rightarrow y^1$}
g	{fn x $\Rightarrow f^3$}
h	{fn v $\Rightarrow v^5$}
v, y	$\emptyset$
x	{fn v $\Rightarrow v^5$, fn y $\Rightarrow y^1$}

To prove its validity, the following constraints need to hold:

$$(\widehat{C}, \widehat{\rho}) \models (\quad)^{14} \text{ iff}$$
$$(\widehat{C}, \widehat{\rho}) \models (\quad)^{2} \wedge (\widehat{C}, \widehat{\rho}) \models [\quad]^{13} \wedge \widehat{C}(2) \subseteq \widehat{\rho}(f) \wedge \widehat{C}(13) \subseteq \widehat{C}(14)$$
$$(\widehat{C}, \widehat{\rho}) \models (\quad)^{2} \text{ iff } \{\text{fn } y \Rightarrow y^1\} \subseteq \widehat{C}(2)$$
$$(\widehat{C}, \widehat{\rho}) \models [\quad]^{13} \text{ iff}$$
$$(\widehat{C}, \widehat{\rho}) \models (\quad)^{4} \wedge (\widehat{C}, \widehat{\rho}) \models (\quad)^{12} \wedge \widehat{C}(4) \subseteq \widehat{\rho}(g) \wedge \widehat{C}(12) \subseteq \widehat{C}(13)$$
$$(\widehat{C}, \widehat{\rho}) \models (\quad)^{4} \text{ iff } \{\text{fn } x \Rightarrow f^3\} \subseteq \widehat{C}(4)$$
$$(\widehat{C}, \widehat{\rho}) \models (\quad)^{12} \text{iff}$$
$$(\widehat{C}, \widehat{\rho}) \models (\quad)^{6} \wedge (\widehat{C}, \widehat{\rho}) \models [\quad]^{11} \wedge \widehat{C}(6) \subseteq \widehat{\rho}(h) \wedge \widehat{C}(11) \subseteq \widehat{C}(12)$$
$$(\widehat{C}, \widehat{\rho}) \models (\quad)^{6} \text{ iff } \{\text{fn } v \Rightarrow v^5\} \subseteq \widehat{C}(6)$$
$$(\widehat{C}, \widehat{\rho}) \models [g^7(\quad)^{10}]^{11} \text{ iff}$$
$$(\widehat{C}, \widehat{\rho}) \models g^7 \wedge (\widehat{C}, \widehat{\rho}) \models (g^8\ h^9)^{10} \wedge$$
$$(\widehat{C}, \widehat{\rho}) \models f^3 \wedge \widehat{C}(10) \subseteq \widehat{\rho}(x) \wedge \widehat{C}(3) \subseteq \widehat{C}(11)$$
$$(\widehat{C}, \widehat{\rho}) \models (g^8\ h^9)^{10} \text{ iff}$$
$$(\widehat{C}, \widehat{\rho}) \models g^8 \wedge (\widehat{C}, \widehat{\rho}) \models h^9 \wedge$$
$$(\widehat{C}, \widehat{\rho}) \models f^3 \wedge \widehat{C}(9) \subseteq \widehat{\rho}(x) \wedge \widehat{C}(3) \subseteq \widehat{C}(10)$$

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.

Solution

The constraint $(\widehat{C}, \widehat{\rho}) \models t_2^{l_2}$ only needs to be fulfilled if $t_1^{l_1}$ evaluates to a function.

$$
\begin{aligned}
[app] \quad & (\widehat{C}, \widehat{\rho}) \models (t_1^{l_1}\ t_2^{l_2})^l \text{ iff} \\
& (\widehat{C}, \widehat{\rho}) \models t_1^{l_1} \wedge \\
& \Big(\forall [\text{fn } x \Rightarrow t_0^{l_0} \in \widehat{C}(l_1)] : \\
& \qquad (\widehat{C}, \widehat{\rho}) \models t_0^{l_0} \wedge (\widehat{\mathbf{C}}, \widehat{\rho}) \models \mathbf{t_2^{l_2}} \wedge \\
& \qquad \widehat{C}(l_2) \subseteq \widehat{\rho}(x) \wedge \widehat{C}(l_0) \subseteq \widehat{C}(l)\Big) \\
& \wedge \Big(\forall [\text{fun } f\, x \Rightarrow t_0^{l_0} \in \widehat{C}(l_1)] : \\
& \qquad (\widehat{C}, \widehat{\rho}) \models t_0^{l_0} \wedge (\widehat{\mathbf{C}}, \widehat{\rho}) \models \mathbf{t_2^{l_2}} \wedge \\
& \qquad \widehat{C}(l_2) \subseteq \widehat{\rho}(x) \wedge \widehat{C}(l_0) \subseteq \widehat{C}(l) \wedge \\
& \qquad\qquad \{\text{fun } fx \Rightarrow t_0^{l_0}\} \subseteq \widehat{\rho}(f)\Big)
\end{aligned}
$$