Exercise Sheet 3

Sebastian Hack
Jan Reineke
Christian Hammer
Tomasz Dudziak

Intervals, relational domains, and widening

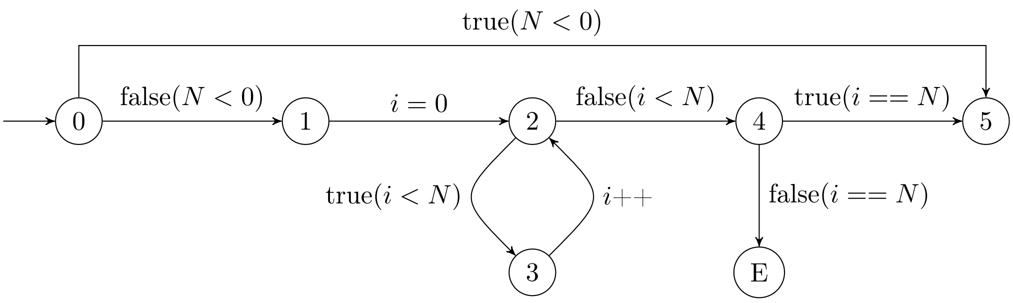

Figure 1: Control flow graph for the example program

Exercise 3.1: 6 points

Consider the following program and its control flow graph in Figure 1.

```
if (N < 0) return;
for (i = 0; i < N; i++) {
  // do something that doesn't affect i or N
}
assert(i == N);
```

Note that we abstract from the loop body (for example by slicing the program on the asserted expression) and the `assert` statement is represented by a conditional jump to an error location E.

By $\text{Var} = \{x_1, \dots, x_n\}$ we will denote the finite set of program variables, $\text{State} = \text{Var} \to \mathbb{Z}$ is the set of program states. Reachability semantics for the CFG edges is defined using the following function $[\![.]\!] \colon \mathcal{P}(\text{State}) \to \mathcal{P}(\text{State})$.

$$
\begin{aligned}
[\![\text{true}(x_i < x_j)]\!]X &= \{\pi \in X \colon \pi(x_i) < \pi(x_j)\} && \text{for } x_i, x_j \in \text{Var} \\
[\![\text{true}(x_i == x_j)]\!]X &= \{\pi \in X \colon \pi(x_i) = \pi(x_j)\} && \text{for } x_i, x_j \in \text{Var} \\
[\![\text{true}(x_i < C)]\!]X &= \{\pi \in X \colon \pi(x_i) < C\} && \text{for } x_i \in \text{Var},\ C \in \mathbb{Z} \\
[\![\text{false}(e)]\!]X &= X \setminus ([\![\text{true}(e)]\!]X) && \\
[\![x_i = C]\!]X &= \{\pi[x_i \mapsto C] \colon \pi \in X\} && \text{for } x_i \in \text{Var},\ C \in \mathbb{Z} \\
[\![x_i{+}{+}]\!]X &= \{\pi[x_i \mapsto \pi(x_i) + 1] \colon \pi \in X\} && \text{for } x_i \in \text{Var}
\end{aligned}
$$

Design an abstract domain $(A, \sqsubseteq)$ that is expressive enough to prove the assertion in the example program. Your domain should be a complete lattice **of finite height**. Define a Galois connection $(\mathcal{P}(\text{State}), \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A, \sqsubseteq)$. For each edge in the control flow graph derive the best abstract operation, i.e.,

$$[\![e]\!]^\# = \alpha \circ [\![e]\!] \circ \gamma$$

Having defined all the operations, perform the analysis on the example program, i.e., provide the least solution to the following system of equations in A.

$$\begin{aligned}
S_0 &= \top_A \\
S_1 &= [\![\text{false}(N < 0)]\!]^\# S_0 \\
S_2 &= ([\![i = 0]\!]^\# S_1) \sqcup ([\![i{+}{+}]\!]^\# S_3) \\
S_3 &= [\![\text{true}(i < N)]\!]^\# S_2 \\
S_4 &= [\![\text{false}(i < N)]\!]^\# S_2 \\
S_E &= [\![\text{false}(i == N)]\!]^\# S_4 \\
S_5 &= ([\![\text{true}(i == N)]\!]^\# S_4) \sqcup ([\![\text{true}(N < 0)]\!]^\# S_0)
\end{aligned}$$

If everything goes well, the abstract value for S_E should be $\perp$. This signifies that the error location is unreachable and the assertion in the program always holds.

Hint: The standard rule-of-signs analysis based on $\text{Var} \to \mathcal{P}(\text{Sign})$ will not work here but you can save a bit of work by also basing your abstract domain on $\mathcal{P}(\text{Sign})$. In your definitions, you are allowed to use the abstract operations in $\mathcal{P}(\text{Sign})$ and the functions α_{Sign} and γ_{Sign} that provide an interpretation of the elements in this lattice via a Galois connection $(\mathcal{P}(\mathbb{Z}), \subseteq) \underset{\alpha_{\text{Sign}}}{\overset{\gamma_{\text{Sign}}}{\leftrightarrows}} (\mathcal{P}(\text{Sign}), \subseteq)$.

Exercise 3.2: 3 points

Consider the following set L

$$\begin{aligned}
L = \{X \subseteq (\mathbb{Z} \cup \{+, -\})\colon\ & X \text{ is finite} \wedge \\
& \wedge (+ \in X \implies \forall x \in X \cap \mathbb{Z}.\ x \leq 0) \wedge \\
& \wedge (- \in X \implies \forall x \in X \cap \mathbb{Z}.\ x \geq 0)\}
\end{aligned}$$

and a function $\gamma\colon L \to \mathcal{P}(\mathbb{Z})$ defined

$$\gamma(X) = \{x \in \mathbb{Z}\colon x \in X \vee (x > 0 \wedge + \in X) \vee (x < 0 \wedge - \in X)\}$$

Define the ordering on L such that γ is monotone. Design a widening operator for L. Prove both safety and termination properties of your operator.

Exercise 3.3: 3 points

Suppose that Var is the finite set of program variables and $\mathcal{P}(\text{Var} \to \mathbb{Z})$ is the concrete domain. The division operation in the concrete semantics is defined as follows.

$$\begin{aligned}
&[\![x := y/z]\!]\ : \mathcal{P}(\text{Var} \to \mathbb{Z}) \to \mathcal{P}(\text{Var} \to \mathbb{Z}) \\
&[\![x := y/z]\!]\ X = \{\pi[x \mapsto \lfloor \pi(y)/\pi(z) \rfloor]\colon \pi \in X \wedge \pi(z) \neq 0\} \qquad \text{for } x, y, z \in \text{Var}
\end{aligned}$$

Note that this means that the program execution does not continue when a division-by-zero error occurs.

Your task is to derive **the most precise** abstract operator $[\![x := y/z]\!]^\#$ for the interval domain from the lecture.