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 $\text{Var} = \{x_1, \ldots, x_n\}$ we will denote the finite set of program variables, $\text{State} = \text{Var} \rightarrow \mathbb{Z}$ is the set of program states. Reachability semantics for the CFG edges is defined using the following function $\llbracket\cdot\rrbracket: \mathcal{P}(\text{State}) \rightarrow \mathcal{P}(\text{State})$.

- $\llbracket \text{true}(x_i < x_j) \rrbracket X = \{\pi \in X : \pi(x_i) < \pi(x_j)\}$ for $x_i, x_j \in \text{Var}$
- $\llbracket \text{true}(x_i == x_j) \rrbracket X = \{\pi \in X : \pi(x_i) = \pi(x_j)\}$ for $x_i, x_j \in \text{Var}$
- $\llbracket \text{true}(x_i < C) \rrbracket X = \{\pi \in X : \pi(x_i) < C\}$ for $x_i \in \text{Var}, C \in \mathbb{Z}$
- $\llbracket \text{false}(e) \rrbracket X = X \setminus (\llbracket \text{true}(e) \rrbracket X)$
- $\llbracket x_i == C \rrbracket X = \{\pi[x_i \mapsto C] : \pi \in X\}$ for $x_i \in \text{Var}, C \in \mathbb{Z}$
- $\llbracket x_i++ \rrbracket X = \{\pi[x_i \mapsto \pi(x_i) + 1] : \pi \in X\}$ for $x_i \in \text{Var}$

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 $(\mathcal{P}(\text{State}), \subseteq) \xrightarrow{\alpha} (A, \sqsubseteq)$. For each edge in the control flow graph derive the best abstract operation, i.e.,

$\llbracket e \rrbracket^\# = \alpha \circ \llbracket e \rrbracket \circ \gamma$
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 $A$.

\[
\begin{align*}
S_0 &= \top_A \\
S_1 &= [\text{false}(N < 0)]^#S_0 \\
S_2 &= ([i = 0]^#S_1) \cup ([i + 1]^#S_3) \\
S_3 &= [\text{true}(i < N)]^#S_2 \\
S_4 &= [\text{false}(i < N)]^#S_2 \\
S_E &= [\text{false}(i == N)]^#S_4 \\
S_5 &= ([\text{true}(i == N)]^#S_4) \sqcup ([\text{true}(N < 0)]^#S_0)
\end{align*}
\]

If everything goes well, the abstract value for $S_E$ should be $\bot$. 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 $\text{Var} \rightarrow \mathcal{P}(\text{Sign})$ 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 $\alpha_{\text{Sign}}$ and $\gamma_{\text{Sign}}$ that provide an interpretation of the elements in this lattice via a Galois connection $(\mathcal{P}(\mathbb{Z}), \subseteq) \xrightarrow{\alpha_{\text{Sign}}} (\mathcal{P}(\text{Sign}), \subseteq)$.

**Exercise 3.2:** 3 points

Consider the following set $L$

\[
L = \{X \subseteq (\mathbb{Z} \cup \{+, -\}) : \text{X is finite} \land \forall x \in X \cap \mathbb{Z}. (x \leq 0) \land \forall x \in X \cap \mathbb{Z}. (x \geq 0)\}
\]

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

\[
\gamma(X) = \{x \in \mathbb{Z} : x \in X \lor (x > 0 \land + \in X) \lor (x < 0 \land - \in X)\}
\]

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

**Exercise 3.3:** 3 points

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

\[
\begin{align*}
[x := y/z] : \mathcal{P}(\text{Var} \rightarrow \mathbb{Z}) & \rightarrow \mathcal{P}(\text{Var} \rightarrow \mathbb{Z}) \\
[x := y/z] X &= \{\pi[x \mapsto \lceil \pi(y)/\pi(z) \rceil] : \pi \in X \land \pi(z) \neq 0\} \quad \text{for } x, y, z \in \text{Var}
\end{align*}
\]

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.