Exercise 1: Interval Analysis

Consider the following program:

1. Perform interval analysis with accelerated widening and narrowing. Make use of the more precise transfer functions \( [\text{Pos}(e)]^\sharp \) and \( [\text{Neg}(e)]^\sharp \) which are defined in the slides (slides 329, 330). Use the program point 4 as a loop separator point. Solve using RR-iteration and present the tables.

2. Show that a bounds check can be removed.

Exercise 2: Points-to analysis (alias analysis idea 1)

Consider the following program:

1. Write down the constraint system for the points-to analysis (slide 374). Use subscripts to access tuple components, e.g., if \( P[5] = (D, M) \), then \( P[5]_1 = D \) and \( P[5]_2 = M \). Start with:

\[
P[0] \equiv (\emptyset, \emptyset) \\
P[1] \equiv (P[0]_1(x \mapsto \{(0, 1)\}), P[0]_2)
\]

2. Define \( \sqsubseteq \) for the domain \((\text{Vars} \to 2^{\text{Addr}^\sharp}) \times (\text{Addr}^\sharp \to 2^{\text{Addr}^\sharp})\).

3. Give the least solution to the constraint system.

4. Assume that the analysis for constant propagation was extended in order to deal with pointers and memory. Then, for this particular program, does the pointer analysis help us in order to perform constant propagation? Give a precise answer why or why not pointer analysis supports constant propagation.
Exercise 3: Flow insensitive alias analysis (alias analysis idea 2)
Consider the following program.

1. Write down the constraint system for the flow insensitive alias analysis (slide 378).
2. Give the least solution to the constraint system.

Exercise 4: Partitioning alias analysis (alias analysis idea 3)
Consider the following program (same as in the previous exercise).

1. Perform the analysis (slide 383) to find out aliasing information.
2. Using your results, show that the second load from \( \text{a}[i] \) is superfluous.