Exercise 1: Pointer Analyses

Consider the following simple program:

\[
\begin{align*}
0: & \quad x := \text{new}(); \\
1: & \quad y := \text{new}(); \\
2: & \quad x[0] = y; \\
3: & \quad y[0] = x; \\
4: & \quad x = y; \\
5: & \quad x := \text{new}();
\end{align*}
\]

Give the result of applying the following analyses which correspond to the three “ideas” presented during the Lectures 10–11 (Nov. 22–25).

a) For the flow-sensitive points-to analysis, give the destination of pointers at each program point, as on page 377, except with one more column for the value of \((1, 2)\).

b) For the flow-insensitive analysis, give the corresponding constraint system, and then the final results, as on page 381.

c) For the flow-insensitive alias analysis, show the equivalence classes as you apply each statement from top to bottom, as on page 388.

Exercise 2: Worklist Iteration

Consider the following constraint system:

\[
\begin{align*}
x_1 & \supseteq \{a\} \cap x_2 \\
x_2 & \supseteq (a \in x_3) \cup \{a, b\} : \{a\} \\
x_3 & \supseteq x_2
\end{align*}
\]

The conditional operator in the second constraint is defined as follows:

\[
(c) ? e_1 : e_2 = \begin{cases} 
  e_1 & \text{if } c \text{ is true} \\
  e_2 & \text{if } c \text{ is false}
\end{cases}
\]

(It is the same operator as you have in C or Java.)

a) Trace the execution of the worklist iteration. It suffices to make a similar table to the one on page 409 of the lecture notes (Lecture 12, Nov 29).

b) Assume we replace the second constraint with \(x_2 \supseteq (a \in x_3) \cup \{b\} : \{a\}\), which is not monotonic. Will the worklist iteration terminate? Explain why.