Exercise 10

Exercise 1

Give two programs that are output equivalent (i.e. for the same initial state they result in the same final state) under the concrete domain, and they are not output equivalent under the interval domain.

Exercise 2

1. The pointer analysis abstract domain is as the mapping:

   $$\mathsf{Labs} \to ((\mathsf{PtrVar} \to \mathcal{P}(\mathsf{AbsObj})) \times (\mathsf{AbsObj} \times \mathsf{Field} \to \mathcal{P}(\mathsf{AbsObj})))$$

   Define:

   (a) the partial order $\sqsubseteq$
   (b) the least ($\bot$) and greatest ($\top$) elements
   (c) the meet $\sqcap$
   (d) the join $\sqcup$

2. The abstraction function α maps sets of states to pointer maps:

   $$\mathcal{P}(\Sigma) \to \big(\mathsf{Labs} \to ((\mathsf{PtrVar} \to \mathcal{P}(\mathsf{AbsObj})) \times (\mathsf{AbsObj} \times \mathsf{Field} \to \mathcal{P}(\mathsf{AbsObj})))\big)$$

   and the abstraction function γ maps pointer maps to sets of states:

   $$\big(\mathsf{Labs} \to ((\mathsf{PtrVar} \to \mathcal{P}(\mathsf{AbsObj})) \times (\mathsf{AbsObj} \times \mathsf{Field} \to \mathcal{P}(\mathsf{AbsObj})))\big) \to \mathcal{P}(\Sigma)$$

   Here the states are defined as $\sigma = (l, \langle \rho, r, h \rangle) \in \Sigma$ where
   $l \in \mathsf{Labs}$
   $\rho : \mathsf{Var} \to Z$,
   $r : \mathsf{PtrVar} \to \mathsf{Objs} \cup \{\mathsf{null}\}$
   $h : \mathsf{Objs} \to (\mathsf{Field} \to \mathsf{Objs} \cup \{\mathsf{null}\} \cup Z)$

   (a) Define α and γ.

(b) Apply α on the following concrete set of states.

To avoid clutter, we do not write the function ρ in states. The initialization locations for the objects are $loc(o_1) = a_1$, $loc(o_2) = a_1$, and $loc(o_3) = a_2$.

$$\begin{aligned} S = \{ \quad & (1, \{p \mapsto o_1, q \mapsto o_2\}, \{o_1.k \mapsto o_3\}), \\ & (2, \{p \mapsto o_2, q \mapsto o_3\}, \{o_1.k \mapsto o_2\}), \\ & (2, \{p \mapsto o_1, q \mapsto o_1\}, \{o_1.k \mapsto o_2\}), \\ & (3, \{p \mapsto o_1, q \mapsto o_3\}, \{o_1.k \mapsto o_1\}) \quad \} \end{aligned}$$

(c) Apply γ on

$$m = \{1 \mapsto (\{p \mapsto \{a_1, a_2\}, q \mapsto \{a_2\}\}, \{a_1.k \mapsto \{a_2\}\})\}$$

Exercise 3

In the lecture you have seen the abstract transformer for *pointer heap store* $p.f := q$. Define the remaining abstract transformers for the interval domain:

(object creation)	$p := newObject^l$
(compare two pointers)	$p = q$
(pointer assignment)	$p := q$
(pointer heap load)	$p := q.f$