Written exam
MPRI 2-6, year 2013–2014

Antoine Miné

6 December 2013

Correction

Part I

1. ⋆ By idempotence, $\rho(X)$ is exactly the set of fixpoints of ρ. As ρ is monotonic in a complete lattice, we can apply Tarski's theorem to get that $\rho(X)$ forms a complete lattice. In particular, $\rho(\bot)$, $\rho(\top)$, $\rho(\sqcup A)$, and $\rho(\sqcap A)$ are all fixpoints.
   By monotony, $\forall x : \bot \sqsubseteq x$ implies $\rho(\bot) \sqsubseteq \rho(x)$, and so, $\rho(\bot)$ is the least fixpoint. Likewise, $\forall x : \rho(x) \sqsubseteq \rho(\top)$, and so, $\rho(\top)$ is the greatest fixpoint. Note that, by extensivity, we also have $\top \sqsubseteq \rho(\top)$, which means that, in fact, $\rho(\top) = \top$.
   Assume that $A \subseteq \rho(X)$. We know that $\sqcup A \sqsubseteq \rho(\sqcup A)$ by extensivity. Assume that $y \in \rho(X)$ is another fixpoint greater than $\sqcup A$. By monotony, $\rho(\sqcup A) \sqsubseteq \rho(y) = y$. Hence $\rho(\sqcup A)$ is the least fixpoint greater than $\sqcup A$, i.e., $\sqcup_\rho A$. Likewise $\sqcap_\rho A = \rho(\sqcap A)$.
   Finally, to prove that $\sqcap_\rho = \sqcap$, we prove that $\rho(\sqcap A) = \sqcap A$. By extensivity, $\sqcap A \sqsubseteq \rho(\sqcap A)$. By monotony, $\forall a \in A : \sqcap A \sqsubseteq a$ implies $\rho(\sqcap A) \sqsubseteq \rho(a) = a$, hence $\rho(\sqcap A) \sqsubseteq \sqcap A$, and so, $\rho(\sqcap A) = \sqcap A$.

   ⋆ Application: As $\sqcap_\rho = \sqcap$, the set of fixpoints $\rho(X)$ of an upper closure operator ρ is closed under meet. However, the set $S \overset{\text{def}}{=} \{\, \emptyset,]-\infty, 0], [0, +\infty[, \mathbb{Z} \,\}$ is not closed under meet as $]-\infty, 0] \cap [0, +\infty[= \{0\} \notin S$, hence, no ρ satisfies $\rho(X) = S$.

2. ⋆ We note that, by definition, $\rho(x) = \sqcap \{\, y \in \rho(X) \mid \rho(x) \sqsubseteq y \,\}$. Assume that $y \in \rho(X)$. Then $x \sqsubseteq y$ implies by monotony $\rho(x) \sqsubseteq \rho(y) = y$. Likewise, $\rho(x) \sqsubseteq y$ implies $x \sqsubseteq y$ as, by extensivity, $x \sqsubseteq \rho(x)$. We deduce that $\{\, y \in \rho(X) \mid x \sqsubseteq y \,\} = \{\, y \in \rho(X) \mid \rho(x) \sqsubseteq y \,\}$, hence, $\rho(x) = \sqcap \{\, y \in \rho(X) \mid x \sqsubseteq y \,\}$.

   ⋆ Assume that $\rho \leq \eta$ and $x \in \eta(X)$, i.e., $x = \eta(x)$. As $\rho(x) \sqsubseteq \eta(x)$, we have $x \sqsupseteq \rho(x)$. By extensivity, we have $x \sqsubseteq \rho(x)$, so, $x = \rho(x)$ and $x \in \rho(X)$. We thus deduce that $\eta(X) \subseteq \rho(X)$.
   To prove the converse, assume that $\rho(X) \supseteq \eta(X)$. Then $\forall x : \{\, y \in \rho(X) \mid x \sqsubseteq y \,\} \supseteq \{\, y \in \eta(X) \mid x \sqsubseteq y \,\}$, and so, $\sqcap \{\, y \in \rho(X) \mid x \sqsubseteq y \,\} \sqsubseteq \sqcap \{\, y \in \eta(X) \mid x \sqsubseteq y \,\}$. Using the first property proved in 2, we get that $\rho(x) \sqsubseteq \eta(x)$, and so, $\rho \leq \eta$.

   ⋆ Application: Given $S \overset{\text{def}}{=} \{\, \emptyset,]-\infty, 0], \{0\}, [0, +\infty[, \mathbb{Z} \,\}$, we get the following closure operator:

$$\rho(x) \overset{\text{def}}{=} \begin{cases} \emptyset & \text{if } x = \emptyset \\ \{0\} & \text{if } x = \{0\} \\ ]-\infty, 0] & \text{if } \forall v \in x : v \leq 0 \text{ and } \exists v \in x : v < 0 \\ [0, +\infty[& \text{if } \forall v \in x : v \geq 0 \text{ and } \exists v \in x : v > 0 \\ \mathbb{Z} & \text{if } \exists v \in x : v < 0 \text{ and } \exists v \in x : v > 0 \end{cases}$$

3. ⋆ Let $g \overset{\text{def}}{=} \mathit{clo}(f) \overset{\text{def}}{=} \lambda x.\,\mathrm{lfp}\,\lambda y.x \sqcup f(y)$. We prove that it is an upper closure operator. By definition, $\forall x : g(x) = x \sqcup f(g(x))$. Hence $x \sqsubseteq g(x)$ and g is extensive. If $x \sqsubseteq x'$, we have $\lambda y.x \sqcup f(y) \leq \lambda y.x' \sqcup f(y)$. Using Tarski's characterization of least fixpoints, $\mathrm{lfp}\, f = \sqcap \{\, x \mid f(x) \sqsubseteq x \,\}$ we deduce that $\mathrm{lfp}\,\lambda y.x \sqcup f(y) \sqsubseteq \mathrm{lfp}\,\lambda y.x' \sqcup f(y)$, hence g is monotonic. Finally, $g(g(x)) = \mathrm{lfp}\,\lambda y.g(x) \sqcup f(y)$. On the one hand, as $g(x) = x \sqcup f(g(x))$, we have $g(x) = g(x) \sqcup f(g(x))$, so $g(x)$ is a fixpoint of $\lambda y.g(x) \sqcup f(y)$; on the other hand, its least fixpoint is greater than $g(x)$, hence, it is $g(x)$ and we have $g(g(x)) = g(x)$, i.e., g is idempotent.

   Assume that h is an upper closure operator greater than f. Then $\forall x : x \sqsubseteq h(x)$; moreover, $\forall x : f(x) \sqsubseteq h(x)$ and so $f(h(x)) \sqsubseteq h(h(x)) = h(x)$, hence $x \sqcup f(h(x)) \sqsubseteq h(x)$. Thus, $h(x)$ is a post-fixpoint of the function $\lambda y.x \sqcup f(y)$, and so it is greater than its least fixpoint $g(x)$, i.e., $g \leq h$.

   ⋆ Application: f is monotonic, extensive, but not idempotent as $f(\{\,-1\,\}) =]-\infty, 0]$ and $f(]-\infty, 0]) = \mathbb{Z}$, so, it is not an upper closure operator. $\mathit{clo}(f)$ is the following upper closure:

$$\rho(x) = \begin{cases} \emptyset & \text{if } x = \emptyset \\ [0, +\infty[& \text{if } \forall v \in x : v \geq 0 \\ \mathbb{Z} & \text{if } \exists v \in x : v < 0 \end{cases}$$

   whose fixpoints are $\rho(X) = \{\, \emptyset, [0, +\infty[, \mathbb{Z} \,\}$.

4. ⋆ $\mathit{mon}(X)$ ordered by $\leq$ forms a complete lattice: $(\mathit{mon}(X), \leq, \vee, \wedge, \lambda x.\bot, \lambda x.\top)$, where the join and meet are point-wise: $\vee F \overset{\text{def}}{=} \lambda x. \sqcup \{\, f(x) \mid f \in F \,\}$ and $\wedge F \overset{\text{def}}{=} \lambda x. \sqcap \{\, f(x) \mid f \in F \,\}$. By 3, clo is an upper closure operator on $\mathit{mon}(X)$, whose fixpoints are exactly the upper closure operators on X. We can thus apply 1 to get the desired property. By 1, $\bot_{\mathit{uco}} = \mathit{clo}(\lambda x.\bot) = \lambda x.x$, $\top_{\mathit{uco}} = \mathit{clo}(\lambda x.\top) = \lambda x.\top$, $\vee_{\mathit{uco}} F = \mathit{clo}(\vee F)$, $\wedge_{\mathit{uco}} F = \wedge$.

   ⋆ Application:

$$\begin{array}{rcl} (\rho_1 \vee_{\mathit{uco}} \rho_2)(X) & = & \{\, \emptyset, \mathbb{Z} \,\} \\ (\rho_1 \wedge_{\mathit{uco}} \rho_2)(X) & = & \{\, \emptyset,\ \{0\},\ ]-\infty, 0] \cap (2\mathbb{Z}),\ ]-\infty, 0] \cap (2\mathbb{Z}+1),\ ]-\infty, 0], \\ & & [0, +\infty[\cap (2\mathbb{Z}),\ [0, +\infty[\cap (2\mathbb{Z}+1),\ [0, +\infty[,\ 2\mathbb{Z},\ 2\mathbb{Z}+1,\ \mathbb{Z} \,\} \end{array}$$

5. ⋆ We simply use the closures from question 4. Note that $(\rho_2 \circ \rho_1)(\{0\}) = \rho_2(\{0\}) = 2\mathbb{Z}$, but $(\rho_2 \circ \rho_1)(2\mathbb{Z}) = \rho_2(\mathbb{Z}) = \mathbb{Z} \neq 2\mathbb{Z}$, hence $\rho_2 \circ \rho_1$ is not idempotent and it is not an upper closure operator.

   ⋆ Recall that $\rho_1 \vee_{\mathit{uco}} \rho_2 = \mathit{clo}(\rho_1 \vee \rho_2)$. As $\rho_1 \leq \rho_1 \vee \rho_2$, we have $\rho_1 \leq \rho_1 \vee_{\mathit{uco}} \rho_2$ and, likewise, $\rho_2 \leq \rho_1 \vee_{\mathit{uco}} \rho_2$. Hence, by composition $\rho_1 \circ \rho_2 \leq (\rho_1 \vee_{\mathit{uco}} \rho_2) \circ (\rho_1 \vee_{\mathit{uco}} \rho_2)$ and, by indempotence, $\rho_1 \circ \rho_2 \leq \rho_1 \vee_{\mathit{uco}} \rho_2$. Finally, $\mathit{clo}(\rho_1 \circ \rho_2) \leq \mathit{clo}(\rho_1 \vee_{\mathit{uco}} \rho_2) = \rho_1 \vee_{\mathit{uco}} \rho_2$. To prove the converse inequality, we first note that $\rho_1 \leq \rho_1 \circ \rho_2$ by extensivity of ρ_2 and monotony of ρ_1. Moreover, $\rho_2 \leq \rho_1 \circ \rho_2$ by extensivity of ρ_1. Hence, $\rho_1 \vee \rho_2 \leq \rho_1 \circ \rho_2$, and so, $\rho_1 \vee_{\mathit{uco}} \rho_2 \leq \mathit{clo}(\rho_1 \circ \rho_2)$.

Historical notes: *Upper closures operators provide an alternative to Galois connections to define and study abstractions. We saw in the course that, for any Galois connection (α, γ), $\gamma \circ \alpha$ is an upper closure operator. Conversely, every upper closure operator $\rho : X \to X$ provides a Galois insersion $(\rho, \lambda x.x)$ between X and $\rho(X)$. With upper closure operators, we see the abstract domain as a subset of the concrete domain: the subset of properties that can be exactly represented. This view is particularly useful when studying the semantic aspect of abstract domains,* *their expressiveness, without considering their possible implementations as data-structures in computers. The use of upper closure operators in static analysis has been introduced as early as 1978, in P. Cousot's thesis, alongside Galois connections. The lattice of upper closure operators as well as various closure transformers (such as completion, complementation, shell), have been subsequently studied by R. Giacobazzi et al. (see for instance: A. Cortesi, G. Filé, R. Giacobazzi, C. Palamidessi, and F Ranzato. Complementation in Abstract Interpretation. In Proc. SAS'95, pp. 100–117, LNCS 983, Springer, 1995).*

Part II

1. $\star$ For intervals and zones when $n = 3$ we choose, respectively:

$$\mathbf{M}_{\text{intervals}} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \qquad \mathbf{M}_{\text{zones}} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \\ 1 & -1 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

2. We write the matrix $\mathbf{M}$ as a (transposed) row of m column vectors $\mathbf{M} = [\vec{M}_1 \cdots \vec{M}_m]^T$. Given a set $X \subseteq \mathbb{R}^n$, its best abstraction is a vector $\alpha(X) = [\beta_1 \cdots \beta_n]^T$ such that $\beta_i = \max\{\, \vec{M}_i \cdot \vec{x} \mid \vec{x} \in X \,\}$.

   We now prove that (α, γ) forms a Galois connection:

$$\begin{array}{ll} & \alpha(X) \leq \vec{\beta} \\ \iff & \forall i : \max\{\, \vec{M}_i \cdot \vec{x} \mid \vec{x} \in X \,\} \leq \beta_i \\ \iff & \forall i, \vec{x} \in X : \vec{M}_i \cdot \vec{x} \leq \beta_i \\ \iff & \forall \vec{x} \in X : \vec{x} \in \gamma(\vec{\beta}) \\ \iff & X \subseteq \gamma(\vec{\beta}) \end{array}$$

3. $\star$ Consider, in two dimensions, $n = 2$, $m = 3$, $M \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$ and the set of points $X \stackrel{\text{def}}{=} \{\, (x, y) \mid x, y \leq 0 \,\}$. Then, $\gamma([0\ 0\ 1]^T) = \gamma([0\ 0\ 0]^T) = X$.

   $\star$ By property of Galois connections, we know that $\alpha \circ \gamma$ gives a normal form: $\alpha(\gamma(\vec{\beta}))$ is the smallest element in $\mathcal{D}^\sharp$ with the same concretization as $\vec{\beta}$. We apply the definition of α from 2, and note that $\beta'_i \stackrel{\text{def}}{=} \max\{\, \vec{M}_i \cdot \vec{x} \mid \vec{x} \in \gamma(\vec{\beta}) \,\}$ can be computed with linear programming: $\beta'_i = LP(\langle \mathbf{M}, \vec{\beta} \rangle, \vec{M_i})$. On the preceding example, $\alpha(\gamma([0\ 0\ 1]^T)) = [0\ 0\ 0]^T$.

4. $\star$ $(\vec{\beta} \cup^\sharp \vec{\beta'})_i = \max{(\beta_i, \beta'_i)}$ is optimal (but not exact) when $\vec{\beta}$ and $\vec{\beta'}$ are in normal form. To prove the optimality, we use the Galois connection to express the optimal abstraction of $\cup$ as: $\vec{\beta} \cup^\sharp \vec{\beta'} = \alpha(\gamma(\vec{\beta}) \cup \gamma(\vec{\beta'}))$. Using the result of question 3, we get $(\vec{\beta} \cup^\sharp \vec{\beta'})_i = \max{\{\, \vec{M_i} \cdot \vec{x} \mid \vec{x} \in \gamma(\vec{\beta}) \cup \gamma(\vec{\beta'}) \,\}} = \max(\max{\{\, \vec{M_i} \cdot \vec{x} \mid \vec{x} \in \gamma(\vec{\beta}) \,\}}, \max{\{\, \vec{M_i} \cdot \vec{x} \mid \vec{x} \in \gamma(\vec{\beta'}) \,\}}) = \max{(\beta_i, \beta'_i)}$, where the last equality holds only if $\vec{\beta}$ and $\vec{\beta'}$ are in normal form. To show that it does not hold when they are not in normal form, consider the example matrix from 3:

$$\begin{array}{lcl} X_1 \overset{\text{def}}{=} \gamma([0\ 1\ 2]^T) & = & \{\, (x,y) \mid x \leq 0, y \leq 1 \,\} \\ X_2 \overset{\text{def}}{=} \gamma([1\ 0\ 2]^T) & = & \{\, (x,y) \mid x \leq 1, y \leq 0 \,\} \end{array}$$

Then, $\gamma([0\ 1\ 2]^T \cup^\sharp [1\ 0\ 2]^T) = \gamma([1\ 1\ 2]^T) = \{\, (x,y) \mid x \leq 1, y \leq 1 \,\}$, while the optimal join is $\gamma(\alpha(X_1 \cup X_2)) = \{\, (x,y) \mid x \leq 1, y \leq 1, x+y \leq 1 \,\} = \gamma([1\ 1\ 1]^T)$.

Finally, the fact that $\cup^\sharp$ is not exact is a simple consequence of the fact that the abstract domain can only represent convex sets, but the result of $\cup$ is not always convex.

$\star$ $(\vec{\beta} \cap^\sharp \vec{\beta'})_i = \min{(\beta_i, \beta'_i)}$ is an exact abstraction of $\cap$. Indeed, we have:

$$\begin{array}{rcl} \gamma(\vec{\beta} \cap^\sharp \vec{\beta'}) & = & \{\, \vec{x} \in \mathbb{R}^n \mid \mathbf{M}\vec{x} \leq \min(\vec{\beta}, \vec{\beta'}) \,\} \\ & = & \{\, \vec{x} \in \mathbb{R}^n \mid \mathbf{M}\vec{x} \leq \vec{\beta} \wedge \mathbf{M}\vec{x} \leq \vec{\beta'} \,\} \\ & = & \{\, \vec{x} \in \mathbb{R}^n \mid \mathbf{M}\vec{x} \leq \vec{\beta} \,\} \cap \{\, \vec{x} \in \mathbb{R}^n \mid \mathbf{M}\vec{x} \leq \vec{\beta'} \,\} \\ & = & \gamma(\vec{\beta}) \cap \gamma(\vec{\beta'}) \end{array}$$

$\star$ For non-deterministic assignments $\vec{V_k}$, let us note $[\beta'_1 \cdots \beta'_m]^T = [\![\, V_k \leftarrow \mathbf{?} \,]\!]^\sharp [\beta_1 \cdots \beta_m]^T$. We set $\forall i : \beta'_i \overset{\text{def}}{=} \beta_i$ if $M_{ik} = 0$, and $+\infty$ otherwise. The operator is optimal if $\vec{\beta}$ is in normal form.

The proof of optimality is similar to that of the join $\cup^\sharp$: by Galois connection, the optimal operator would give $\beta'_i \overset{\text{def}}{=} \max{\{\, \vec{M_i} \cdot \vec{y} \mid \vec{x} \in \gamma(\vec{\beta}), \vec{y} = \vec{x}[V_k \mapsto v], v \in \mathbb{R} \,\}}$, which equals $+\infty$ if $M_{ik} \neq 0$, and $\max{\{\, \vec{M_i} \cdot \vec{x} \mid \vec{x} \in \gamma(\vec{\beta}) \,\}} = \beta_i$ when in normal form otherwise.

To see that it is not always exact, consider the following template:

$$\begin{bmatrix} 1 & 0 \\ -1 & -1 \\ -1 & 1 \end{bmatrix}$$

which can represent exactly the point $(0,0)$, but not the line $\{\, (x,0) \mid x \in \mathbb{R} \,\}$ which is the image of the point by $x \leftarrow \mathbf{?}$.

$\star$ To handle affine tests and assignments, the simplest way is to reuse the operators from the polyhedra domain. A template element $\vec{\beta}$ can be seen as a polyhedron $\langle \mathbf{M}, \vec{\beta} \rangle$ in constraint form. We saw, in the course, exact abstract operators to perform affine tests and assignments in the constraint representation. Let us denote the result polyhedron $\langle \mathbf{N}, \vec{\delta} \rangle$; note that $\mathbf{N}$ and $\mathbf{M}$ may differ, so that the polyhedron is not necessarily in the template domain. However, we can abstract it back in the template domain using α. Computing $\alpha(\{\, \vec{x} \mid \mathbf{N}\vec{x} \leq \vec{\delta} \,\})$ consists in solving m linear programs: we state $\forall i : \beta'_i \overset{\text{def}}{=} LP(\langle \mathbf{N}, \vec{\delta} \rangle, \vec{M_i})$

Computing the best abstraction of an exact polyhedral operation results in an optimal abstract operator in the template domain. Moreover, the operator is exact when the affine expression tested or assigned has the form $a_0+\sum_i a_i V_i$ where $[a_1\ \dots\ a_n]^T$ matches some template vector $\vec{M}_k$.

5. We can extend any widening defined for the interval, zone, or octagon domain to a widening on template polyhedra. For instance, we extend the standard widening as follows: $(\vec{\beta} \triangledown \vec{\beta}')_i \overset{\text{def}}{=} \beta_i$ if $\beta_i \geq \beta'_i$, and $+\infty$ otherwise. As $(\vec{\beta} \triangledown \vec{\beta}')_i \geq \max(\beta_i, \beta'_i)$, $\triangledown$ overapproximates the join and is thus sound. The termination stems form the fact that any growing component is set to $+\infty$, where it can grow no further.

Historical notes: The template abstract domain has been introduced in the last decade by Sankaranarayanan et al. in order to extend the octagon domain to arbitrary linear relations while being more efficient than polyhedra (see: S. Sankaranarayanan, H. B. Sipma, and Z. Manna. Scalable analysis of linear systems using mathematical programming. In Proc. VMCAI'05, LNCS 3385, Springer, 2005).

Part III

Question 1

The proof that $\llbracket\, \textbf{while}\ e \geq 0\ \textbf{do}\ \{\, P \,\}\,\rrbracket$ is well-defined depends on the fact that $\llbracket\, P \,\rrbracket$ is a complete $\cup-$morphism. Hence, we prove simultaneously both properties by induction on the syntax of P.

- ⋆ When P is $V \leftarrow \textbf{?}$, $V \leftarrow e$, or $e \bowtie 0\,?$, it stems directly from the definition that:

$$\llbracket\, P \,\rrbracket E = \cup\, \{\, \llbracket\, P \,\rrbracket(\{\, \rho \,\}) \mid \rho \in E \,\}$$

  which proves that $\llbracket\, P \,\rrbracket$ is a complete $\cup-$morphism.
- ⋆ If f and g are complete $\cup-$morphisms, then so are their composition $f \circ g$ and their pointwise union $\lambda x.(f(x) \cup g(x))$. Assume by induction hypothesis that $\llbracket\, P \,\rrbracket$, $\llbracket\, P_1 \,\rrbracket$, and $\llbracket\, P_2 \,\rrbracket$ are complete $\cup-$morphisms. We deduce that $\llbracket\, P_1; P_2 \,\rrbracket$ and $\llbracket\, \textbf{if}\ e \geq 0\ \textbf{then}\ \{\, P \,\}\,\rrbracket$ are also complete $\cup-$morphisms.
- ⋆ Let $f(R) \overset{\text{def}}{=} R \cup \llbracket\, P \,\rrbracket(\llbracket\, e \geq 0\,?\,\rrbracket R)$. Assuming as above by induction hypothesis that $\llbracket\, P \,\rrbracket$ is a complete $\cup-$morphism, then so is f. Moreover, $E \subseteq f(E)$ and $(\mathcal{P}(\mathcal{E}), \subseteq)$ is a complete partial order. Hence, we can apply Kleene's fixpoint theorem, which proves that $\mathrm{lfp}_E\, f$ is well-defined. By Kleene's theorem, we also have $\mathrm{lfp}_E\, f = \cup_{n\in\mathbb{N}}\, f^n(E)$. As f is a complete $\cup-$morphism, so is every f^n. This implies that $\lambda E.(\mathrm{lfp}_E\, f)$ is also a complete $\cup-$morphism, and so is $\llbracket\, \textbf{while}\ e \geq 0\ \textbf{do}\ \{\, P \,\}\,\rrbracket$.

Question 2

1. If $A \subseteq B$, then $f(\{a\}) \subseteq A$ implies $f(\{a\}) \subseteq B$, and so, $\overleftarrow{f}(A) \subseteq \overleftarrow{f}(B)$, which proves the monotony. Moreover:

$$\begin{array}{rcl}
\overleftarrow{f}(\cap_{i\in I} B_i) & = & \{\, a \mid f(\{a\}) \subseteq \cap_{i\in I} B_i \,\} \\
 & = & \{\, a \mid\ \wedge_{i\in I}\, f(\{a\}) \subseteq B_i \,\} \\
 & = & \cap_{i\in I}\, \{\, a \mid f(\{a\}) \subseteq B_i \,\} \\
 & = & \cap_{i\in I}\, \overleftarrow{f}(B_i)
\end{array}$$

and so, $\overleftarrow{f}$ is a complete $\cap-$morphism. These two properties are true for any f, even if it is neither strict, nor a $\cup-$morphism, nor even monotonic.

2. f and $\overleftarrow{f}$ are monotonic. It remains to prove that $A \subseteq \overleftarrow{f}(B) \iff f(A) \subseteq B$.

$$\begin{array}{lll} & A \subseteq \overleftarrow{f}(B) & \\ \iff & \forall a \in A : f(\{a\}) \subseteq B & \{\, \text{def. of } \overleftarrow{\cdot} \,\} \\ \iff & \cup_{a \in A} f(\{a\}) \subseteq B & \\ \iff & f(A) \subseteq B & \{\, \text{complete } \cup-\text{morphism} \,\} \end{array}$$

To prove that this does not necessarily hold when f is not a $\cup-$morphism, consider f such that $f(\{\, a \,\}) = f(\{\, b \,\}) = \emptyset$, $f(\{\, a, b \,\}) = \{\, a, b \,\}$. Then $\{\, a, b \,\} = \overleftarrow{f}(\{\, a \,\})$, but $f(\{\, a, b \,\}) = \{\, a, b \,\} \not\subseteq \{\, a \,\}$.

3. If f is monotonic, then we have $\overleftarrow{f \circ g} \subseteq \overleftarrow{g} \circ \overleftarrow{f}$ as:

$$\begin{array}{lll} & a \in \overleftarrow{f \circ g}(B) & \\ \iff & (f \circ g)(\{a\}) \subseteq B & \{\, \text{def. of } \overleftarrow{\cdot} \,\} \\ \implies & \forall b \in g(\{a\}) : f(\{b\}) \subseteq B & \{\, \text{monotony of } f \,\} \\ \iff & \forall b \in g(\{a\}) : b \in \overleftarrow{f}(B) & \{\, \text{def. of } \overleftarrow{\cdot} \,\} \\ \iff & g(\{a\}) \subseteq \overleftarrow{f}(B) & \\ \iff & a \in (\overleftarrow{g} \circ \overleftarrow{f})(B) & \{\, \text{def. of } \overleftarrow{\cdot} \,\} \end{array}$$

If, moreover, f is a complete $\cup-$morphism, then $(\forall b \in g(\{a\}) : f(\{b\}) \subseteq B) \implies (f \circ g)(\{a\}) = \cup \{\, f(\{b\}) \mid b \in g(\{a\}) \,\} \subseteq B$, which proves the equality $\overleftarrow{f \circ g} = \overleftarrow{g} \circ \overleftarrow{f}$. These properties are true even if g is not a complete $\cup-$morphism.

4. The property $\overleftarrow{\lambda x.(f(x) \cup g(x))} = \lambda x.(\overleftarrow{f}(x) \cap \overleftarrow{g}(x))$ is always true, even if neither f nor g is a complete $\cup-$morphism.

$$\begin{array}{lll} & a \in (\overleftarrow{f \cup g})(B) & \\ \iff & f(\{a\}) \cup g(\{a\}) \subseteq B & \{\, \text{def. of } \overleftarrow{\cdot} \,\} \\ \iff & f(\{a\}) \subseteq B \wedge g(\{a\}) \subseteq B & \\ \iff & a \in \overleftarrow{f}(B) \wedge a \in \overleftarrow{g}(B) & \{\, \text{def. of } \overleftarrow{\cdot} \,\} \\ \iff & a \in (\overleftarrow{f} \cap \overleftarrow{g})(B) & \end{array}$$

5. Let us note $h(z) \stackrel{\text{def}}{=} z \cup f(z)$. As it is a complete $\cup-$morphism and any x is a pre-fixpoint of h, we can apply Kleene's fixpoint theorem. We note that each h^i is also a complete $\cup-$morphism. Then:

$$\begin{array}{lll} & \mathrm{lfp}_x\, h \subseteq y & \\ \Longleftrightarrow & \cup_{i\in\mathbb{N}}\, h^i(x) \subseteq y & \{\,\text{Kleene's theorem}\,\} \\ \Longleftrightarrow & \forall i \in \mathbb{N} : h^i(x) \subseteq y & \\ \Longleftrightarrow & \forall i \in \mathbb{N} : x \subseteq \overleftarrow{h^i}(y) & \{\,\text{property } 2\,\} \\ \Longleftrightarrow & \forall i \in \mathbb{N} : x \subseteq \overleftarrow{h}^i(y) & \{\,\text{property } 3\,\} \\ \Longleftrightarrow & x \subseteq \cap_{i\in\mathbb{N}}\, \overleftarrow{h}^i(y) & \\ \Longleftrightarrow & x \subseteq \mathrm{gfp}_y\, \overleftarrow{h} & \{\,\text{Kleene's theorem}\,\} \end{array}$$

i.e., $\overleftarrow{\lambda x.(\mathrm{lfp}_x\, h)} = \lambda y.(\mathrm{gfp}_y\, \overleftarrow{h})$. We conclude using property 4 and the obvious fact that $\overleftarrow{\lambda x.x} = \lambda x.x$ to get $\overleftarrow{h} = \overleftarrow{\lambda x.(x \cup f(x))} = \lambda x.(\overleftarrow{\lambda y.(y)}(x) \cap \overleftarrow{f}(x)) = \lambda x.(x \cap \overleftarrow{f}(x))$.

To show that $\overleftarrow{f}$ is not always strict, even if f is strict or a complete $\cup-$morphism, consider the function f such that $f(\emptyset) = f(\{\, a \,\}) = \emptyset$. Then, $\overleftarrow{f}(\emptyset) = \{\, a \,\} \neq \emptyset$.

Question 3

⋆ The base cases $\overleftarrow{\llbracket V \leftarrow \mathbf{?} \rrbracket}$, $\overleftarrow{\llbracket V \leftarrow e \rrbracket}$, and $\overleftarrow{\llbracket e\,? \rrbracket}$ are solved by applying directly the definition of $\overleftarrow{\cdot}$ to the definition of $\llbracket \cdot \rrbracket$. The cases of loops, tests, and sequences are handled by structural induction and by applying the results of the previous question.

1. $\overleftarrow{\llbracket V \leftarrow \mathbf{?} \rrbracket} E = \{\, \rho \mid \forall v \in \mathbb{Q} : \rho[V \mapsto v] \in E \,\}$
2. $\overleftarrow{\llbracket V \leftarrow a_0 + \sum_i a_i V_i \rrbracket} E = \{\, \rho \mid \rho[V \mapsto a_0 + \sum_i a_i \times \rho(V_i)] \in E \,\}$
3. $\overleftarrow{\llbracket a_0 + \sum_i a_i V_i \geq 0\,? \rrbracket} E = E \cup \{\, \rho \in \mathcal{E} \mid a_0 + \sum_i a_i \times \rho(V_i) < 0 \,\}$
   and
   $\overleftarrow{\llbracket a_0 + \sum_i a_i V_i > 0\,? \rrbracket} E = E \cup \{\, \rho \in \mathcal{E} \mid a_0 + \sum_i a_i \times \rho(V_i) \leq 0 \,\}$
4. $\overleftarrow{\llbracket \textbf{if } e \geq 0 \textbf{ then } \{\, P \,\} \rrbracket} E = \overleftarrow{\llbracket e \geq 0\,? \rrbracket}(\overleftarrow{\llbracket P \rrbracket} E) \cap \overleftarrow{\llbracket e < 0\,? \rrbracket} E$
5. $\overleftarrow{\llbracket \textbf{while } e \geq 0 \textbf{ do } \{\, P \,\} \rrbracket} E = \mathrm{gfp}_{\overleftarrow{\llbracket e<0\,? \rrbracket} E}\, \lambda R.(R \cap \overleftarrow{\llbracket e \geq 0\,? \rrbracket}(\overleftarrow{\llbracket P \rrbracket} R)$

⋆ The fact that $\llbracket P \rrbracket(\overleftarrow{\llbracket P \rrbracket} O) \subseteq O$ (correctness of $\overleftarrow{\llbracket P \rrbracket} O$), and that if $\llbracket P \rrbracket I \subseteq O$ then $I \subseteq \overleftarrow{\llbracket P \rrbracket} O$ (maximality of $\overleftarrow{\llbracket P \rrbracket} O$) is a direct consequence of the Galois connection $\mathcal{P}(\mathcal{E}) \underset{\llbracket P \rrbracket}{\overset{\overleftarrow{\llbracket P \rrbracket}}{\leftrightarrows}} \mathcal{P}(\mathcal{E})$ proved in the previous question.

Question 4

⋆ Let us note $E' \overset{\text{def}}{=} \overleftarrow{\llbracket V \leftarrow \mathbf{?} \rrbracket} E = \{\, \rho \in \mathcal{E} \mid \forall v \in \mathbb{Q} : \rho[V \mapsto v] \in E \,\}$. We note that $E' \subseteq E$: by choosing $v = \rho(V)$, $\rho[V \mapsto v] \in E$ reduces to $\rho \in E$.
We now prove that, if E is convex and closed, and $E' \neq \emptyset$, then $E' = E$. Assume that $E' \neq \emptyset$ and, *ad absurdum*, that $E' \subsetneq E$, i.e., there exist $\rho, \rho' \in E$ such that $\rho \in E'$ but $\rho' \notin E'$. Thus, $\exists v' \in \mathcal{E} : \rho'[V \mapsto v'] \notin E$. For any $\epsilon \in (0, 1]$, we now construct a point ρ'_ϵ in E that is at distance less than ϵ from $\rho'[V \mapsto v']$. We take ρ'_ϵ on the segment between $\rho' \in E$ and $\rho[V \mapsto M_\epsilon] \in E$: $\rho'_\epsilon \overset{\text{def}}{=} (1 - \alpha_\epsilon)\rho' + \alpha_\epsilon \rho[V \mapsto M_\epsilon]$, for some well-chosen M_ϵ and α_ϵ. More precisely, we choose:

$$\begin{aligned} &\alpha_\epsilon = \epsilon / \max\{\, 1,\ |\rho(W) - \rho'(W)| \mid W \neq V \,\} \\ &M_\epsilon = \rho'(V) + (v' - \rho'(V))/\alpha_\epsilon \ . \end{aligned}$$

This implies $\forall W \neq V$:

$$\begin{array}{rcl} |\rho'_\epsilon(W) - \rho'[V \mapsto v'](W)| & = & |((1-\alpha_\epsilon)\rho'(W) + \alpha_\epsilon \rho(W)) - \rho'(W)| \\ & = & \alpha_\epsilon |\rho(W) - \rho'(W)| \\ & \leq & \epsilon \ . \end{array}$$

Moreover:

$$\begin{array}{rcl} |\rho'_\epsilon(V) - \rho'[V \mapsto v'](V)| & = & |((1-\alpha_\epsilon)\rho'(V) + \alpha_\epsilon M_\epsilon) - v'| \\ & = & |\rho'(V) - \alpha_\epsilon \rho'(V) + \alpha_\epsilon \rho'(V) + v' - \rho'(V) - v'| \\ & = & 0 \ . \end{array}$$

So, we indeed have $|\rho'_\epsilon - \rho'[V \mapsto v']|_\infty \leq \epsilon$. Finally, by convexity of E, we have $\rho'_\epsilon \in E$. We can thus construct a sequence of points in E that converges to $\rho'[V \mapsto v']$. As E is closed, this implies $\rho'[V \mapsto v'] \in E$, and so, our hypothesis $\rho' \notin E'$ is false.

$\star$ We now prove the additional property: $[\![\, V \leftarrow \mathbf{?} \,]\!]E = E \iff E = \overleftarrow{[\![\, V \leftarrow \mathbf{?} \,]\!]}E$, so that the case $\overleftarrow{[\![\, V \leftarrow \mathbf{?} \,]\!]}E = E$ can be tested using a forward random assignment operator. Indeed, we have by Galois connection $E \subseteq \overleftarrow{[\![\, V \leftarrow \mathbf{?} \,]\!]}E \iff [\![\, V \leftarrow \mathbf{?} \,]\!]E \subseteq E$. The proof is completed by using the fact that $E \subseteq [\![\, V \leftarrow \mathbf{?} \,]\!]E$, and $\overleftarrow{[\![\, V \leftarrow \mathbf{?} \,]\!]}E \subseteq E$, which we just proved.

$\star$ As the forward random assignment is an exact operator on polyhedra, we can design the following exact backward random assignment on the polyhedra domain:

$$\overleftarrow{[\![\, V \leftarrow \mathbf{?} \,]\!]}^\sharp P^\sharp \overset{\text{def}}{=} \begin{cases} P^\sharp & \text{if } [\![\, V \leftarrow \mathbf{?} \,]\!]^\sharp P^\sharp =^\sharp P^\sharp \\ \bot^\sharp & \text{otherwise} \end{cases}$$

Question 5

$\star$ Recall from question 3 that $\overleftarrow{[\![\, a_0 + \sum_i a_i V_i \geq 0\,? \,]\!]}E = E \cup \{\, \rho \in \mathcal{E} \mid a_0 + \sum_i a_i \times \rho(V_i) < 0 \,\}$, hence, it is not convex in general and cannot be exactly represented as a convex polyhedra. Consider, for instance $\overleftarrow{[\![\, x \geq 0\,? \,]\!]}\{\, x \mid x = 1 \,\} = \{\, x \mid x < 0 \vee x = 1 \,\}$, which is not a polyhedron while $\{\, x \mid x = 1 \,\}$ is.

$\star$ We look for I such that $[\![\, P \,]\!]I \subseteq O$. Recall from question 3 that $\overleftarrow{[\![\, P \,]\!]}O$ is the maximal such I. By monotony of $[\![\, P \,]\!]$, any $I' \subseteq \overleftarrow{[\![\, P \,]\!]}O$ also satisfies $[\![\, P \,]\!]I' \subseteq O$ and so is acceptable.

In the abstract, given $O^\sharp$ that under-approximates O: $\gamma(O^\sharp) \subseteq O$, and $\overleftarrow{[\![\, P \,]\!]}^\sharp$ that under-approxiamtes $\overleftarrow{[\![\, P \,]\!]}$: $\forall X^\sharp : \gamma(\overleftarrow{[\![\, P \,]\!]}^\sharp X^\sharp) \subseteq \overleftarrow{[\![\, P \,]\!]}(\gamma(X^\sharp))$, we deduce that $\gamma(\overleftarrow{[\![\, P \,]\!]}^\sharp O^\sharp)$ under-approximates $\overleftarrow{[\![\, P \,]\!]}(\gamma(O^\sharp))$, and so $[\![\, P \,]\!](\gamma(\overleftarrow{[\![\, P \,]\!]}^\sharp O^\sharp)) \subseteq [\![\, P \,]\!](\overleftarrow{[\![\, P \,]\!]}(\gamma(O^\sharp))) \subseteq \gamma(O^\sharp) \subseteq O$. Hence, soundness here requires under-approximating operators.

$\star$ Given a set C of affine constraints representing a polyhedron $\gamma(C) = \{\, \rho \mid \forall c \in C : \rho \text{ satisfies } c \,\}$, and a constraint $d \overset{\text{def}}{=} a_0 + \sum_i a_i V_i \geq 0$, we show that $C \setminus \{d\}$ is a sound abstraction of $\overleftarrow{[\![\, d\,? \,]\!]}\gamma(C)$. Indeed, we have, by question 3, that

$$\overleftarrow{[\![\, d\,? \,]\!]}\gamma(C) = \gamma(C) \cup \{\, \rho \mid \rho \text{ does not satisfy } d \,\} \ .$$

Consider $\rho \in \gamma(C \setminus \{d\})$. Then, either ρ satisfies d in addition to $C \setminus \{d\}$ and then $\rho \in \gamma(C)$, or ρ does not satisfy d. In both cases, $\rho \in \overleftarrow{[\![\, d\,? \,]\!]}\gamma(C)$, so that $\gamma(C \setminus \{d\}) \subseteq \overleftarrow{[\![\, d\,? \,]\!]}\gamma(C)$, which proves the soundness.

⋆ We note that $\overleftarrow{[\![\, a_0 + \sum_i a_iV_i \geq 0\,?\,]\!]}\gamma(C) \subseteq \overleftarrow{[\![\, a_0 + \sum_i a_iV_i > 0\,?\,]\!]}\gamma(C)$, hence $\gamma(C \setminus \{a_0 + \sum_i a_iV_i \geq 0\})$ is also a sound approximation of $\overleftarrow{[\![\, a_0 + \sum_i a_iV_i > 0\,?\,]\!]}\gamma(C)$.

⋆ Assume now that the constraint $e \in C$ is redundant with d, i.e., $e \neq d$ but $\gamma(C \setminus \{e\} \cup \{d\}) = \gamma(C)$. Then, in the concrete, $\overleftarrow{[\![\, d\,?\,]\!]}\gamma(C) = \overleftarrow{[\![\, d\,?\,]\!]}\gamma(C \setminus \{e\} \cup \{d\})$, so, we can abstract the former as we would abstract the later. As the later is abstracted as $(C \setminus \{e\} \cup \{d\}) \setminus \{d\} = C \setminus \{e\}$, this shows that we can remove any constraint redundant with d when computing $\overleftarrow{[\![\, d\,?\,]\!]}^\sharp C$.

Question 6

Consider the assignment $V_j \leftarrow a_0 \sum_i a_iV_i$. As for the forward polyhedra assignment, we distinguish two cases: invertibles assignments and non-invertible assignments.

⋆ Invertible case: $a_j \neq 0$. We have, in the forward, $[\![\, V_j \leftarrow a_0 + \sum_i a_iV_i \,]\!]X = \{\, \rho[V_j \mapsto a_0 + \sum_i a_i\rho(V_i)] \mid \rho \in X \,\}$. By definition of the $\overleftarrow{\cdot}$ operator, we get:

$$\begin{aligned}
\overleftarrow{[\![\, V_j \leftarrow a_0 \textstyle\sum_i a_iV_i \,]\!]}X &= \{\, \rho \mid [\![\, V_j \leftarrow a_0 + \textstyle\sum_i a_iV_i \,]\!]\, \{\rho\} \in X \,\} \\
&= \{\, \rho \mid \rho[V_j \mapsto a_0 + \textstyle\sum_i a_i\rho(V_i)] \in X \,\} \\
&= \{\, \rho[V_j \mapsto (\rho(V_j) - a_0 - \textstyle\sum_{i\neq j} a_i\rho(V_i))/a_j] \mid \rho \in X \,\} \\
&= [\![\, V_j \leftarrow (V_j - a_0 - \textstyle\sum_{i\neq j} a_iV_i)/a_j \,]\!]X \ .
\end{aligned}$$

We note that the backward version an invertible assignment equals the forward version of the inverse assignment. As the polyhedra abstract affine assignment is exact, we can use it to implement an exact (hence sound) abstract backward assignment:

$$\overleftarrow{[\![\, V_j \leftarrow a_0 + \textstyle\sum_i a_iV_i \,]\!]}^\sharp P^\sharp \;=\; [\![\, V_j \leftarrow (V_j - a_0 - \textstyle\sum_{i\neq j} a_iV_i)/a_j \,]\!]^\sharp P^\sharp \ .$$

⋆ Non-invertible case: $a_j = 0$. Recall from the course that:

$$[\![\, V_j \leftarrow a_0 + \textstyle\sum_i a_iV_i \,]\!] \;=\; [\![\, a_0 + \textstyle\sum_i a_iV_i - V_j = 0\,?\,]\!] \circ [\![\, V_j \leftarrow \mathbf{?} \,]\!]$$

Hence, by the question 2, we get:

$$\overleftarrow{[\![\, V_j \leftarrow a_0 + \textstyle\sum_i a_iV_i \,]\!]} \;=\; \overleftarrow{[\![\, V_j \leftarrow \mathbf{?} \,]\!]} \circ \overleftarrow{[\![\, a_0 + \textstyle\sum_i a_iV_i - V_j = 0\,?\,]\!]}$$

which can be abstracted as $\overleftarrow{[\![\, V_j \leftarrow \mathbf{?} \,]\!]}^\sharp \circ \overleftarrow{[\![\, a_0 + \sum_i a_iV_i - V_j = 0\,?\,]\!]}^\sharp$. We can thus reuse and combine the sound abstractions of the random assignment from question 4 and the assertions from question 5.

Question 7

⋆ Recall that $\overleftarrow{[\![\, \mathbf{while}\; e \geq 0 \;\mathbf{do}\; \{\, P \,\} \,]\!]}E = \mathrm{gfp}_{\overleftarrow{[\![e<0\,?\,]\!]}E}\, \lambda R.(R \cap \overleftarrow{[\![\, e \geq 0\,?\,]\!]}(\overleftarrow{[\![\, P \,]\!]}R))$. To compute an abstract under-approximation of this fixpoint in the polyhedra domain, we assume first that we have constructed a sound abstract under-approximation $F^\sharp$ of $F(R) = R \cap \overleftarrow{[\![\, e \geq 0\,?\,]\!]}(\overleftarrow{[\![\, P \,]\!]}R)$, and an abstract under-approximation $X^\sharp$ of $\overleftarrow{[\![\, e < 0\,?\,]\!]}E$, for instance by using the previous questions. We then construct the iteration $X_0^\sharp \overset{\text{def}}{=} X^\sharp$, $X_{n+1}^\sharp = X_n^\sharp \,\underline{\triangledown}\, F^\sharp(X_n^\sharp)$. By the second part of the definition of $\underline{\triangledown}$, this iteration stabilizes in finite time. Let us denote i this stable iterate: we have $X_i^\sharp = X_i^\sharp \,\underline{\triangledown}\, F^\sharp(X_i^\sharp)$. Using the first part of the definition of $\underline{\triangledown}$, we have $\gamma(X_i^\sharp) = \gamma(X_i^\sharp \,\underline{\triangledown}\, F^\sharp(X_i^\sharp)) \subseteq \gamma(X_i^\sharp) \cap \gamma(F^\sharp(X_i^\sharp))$. In particular, $\gamma(X_i^\sharp) \subseteq \gamma(F^\sharp(X_i^\sharp))$ and, by soundness of $F^\sharp$, $\gamma(X_i^\sharp) \subseteq F(\gamma(X_i^\sharp))$; hence, $\gamma(X_i^\sharp)$ is a pre-fixpoint of F. We also have, by induction on the number of iterations, $\gamma(X_i^\sharp) \subseteq \gamma(X_{i-1}^\sharp) \subseteq \cdots \subseteq \gamma(X_0^\sharp) = \gamma(X^\sharp) \subseteq \overleftarrow{[\![\, e < 0\,?\,]\!]}E$.

By Tarksi's characterization of greatest fixpoints (i.e., $\mathrm{gfp}_x\, F$ is the largest pre-fixpoint of F smaller than x) we deduce that $\gamma(X_i^\sharp) \subseteq \mathrm{gfp}_{\overleftarrow{\llbracket e<0\,?\rrbracket} E}\, F$, hence, it is a sound under-approximation.

- ⋆ Given two polyhedra $\gamma(G_1)$ and $\gamma(G_2)$ given as sets of generators G_1, G_2, we propose the following lower widening: $G_1 \underline{\triangledown}\, G_2 = \{\, g \in G_1 \mid \gamma(G_2 \cup \{g\}) = \gamma(G_2)\,\}$, i.e., we only keep the generators of G_1 that are already included in the polyhedron $\gamma(G_2)$. As $G_1 \underline{\triangledown}\, G_2$ is a subset of the generators of G_1, we have $\gamma(G_1 \underline{\triangledown}\, G_2) \subseteq \gamma(G_1)$. As $G_1 \underline{\triangledown}\, G_2$ contains only generators included in the polyhedron $\gamma(G_2)$, we also have $\gamma(G_1 \underline{\triangledown}\, G_2) \subseteq \gamma(G_2)$. Moreover, as the set of generators in a sequence with lower widening is finite and decreases, the sequence necessarily stabilizes in finite time.

Historical notes: *There is not much work on abstract operators for under-approximations. Most works focus on over-approximations, and so, are suitable for invariant analysis as well as necessary condition inference, but not sufficient condition inference. This problem was inspired by the article: A. Miné. Backward under-approximations in numeric abstract domains to automatically infer sufficient program conditions. In SCP, 33 pages, Oct. 2013, Elsevier.*