Written exam
MPRI 2-6, year 2013–2014

Antoine Miné

6 December 2013

Duration: 3 hours (8:45–11:45)

The only documents allowed are your own printed copy of the course slides and your personal notes. The use of electronic devices (computers, phones) is prohibited.

The questions are written in English. You can answer either in English or French.

The different parts in this exam are independent and can be solved in any order.

It will not be answered to any question during the exam. In case of ambiguity or incorrectness in the definitions or questions, it is part of the exam to correct them and answer to the best of your abilities.

Part I: Exercise

The goal of this exercise is to study upper closure operators, which provide an alternate way to define sets of abstract properties. Let $(X, \sqsubseteq, \sqcup, \sqcap, \bot, \top)$ be a complete lattice. Recall that an operator $\rho : X \to X$ is an upper closure on X if it is monotonic $x \sqsubseteq x' \implies \rho(x) \sqsubseteq \rho(x')$, extensive $x \sqsubseteq \rho(x)$, and idempotent $\rho \circ \rho = \rho$. If $A \subseteq X$, we write $\rho(A) \stackrel{\text{def}}{=} \{\, \rho(a) \mid a \in A \,\}$. We denote by $uco(X)$ the set of upper closures on X and by $mon(X)$ the set of monotonic operators on X; they are ordered by the point-wise order $f \leq g \stackrel{\text{def}}{\iff} \forall x : f(x) \sqsubseteq g(x)$.

1. By Tarski's Theorem, we know that the set of fixpoints of $\rho \in uco(X)$ forms a complete lattice $(\rho(X), \sqsubseteq, \sqcup_\rho, \sqcap_\rho, \bot_\rho, \top_\rho)$. Prove that $\bot_\rho = \rho(\bot)$, $\top_\rho = \rho(\top)$, $\sqcup_\rho A = \rho(\sqcup A)$, and $\sqcap_\rho A = \sqcap A$.

   Application: consider $X \stackrel{\text{def}}{=} \mathcal{P}(\mathbb{Z})$ ordered by $\subseteq$, and prove that there does not exist any closure ρ such that $\rho(X) = \{\, \emptyset,]-\infty, 0], [0, +\infty[, \mathbb{Z} \,\}$.

2. Prove that $\rho(x) = \sqcap \{\, y \in \rho(X) \mid x \sqsubseteq y \,\}$. Deduce from this property that $\rho \leq \eta \iff \rho(X) \supseteq \eta(X)$ and $\rho = \eta \iff \rho(X) = \eta(X)$: a closure is defined by its fixpoints.

   Application: give the closure ρ such that $\rho(X) = \{\, \emptyset,]-\infty, 0], \{0\}, [0, +\infty[, \mathbb{Z} \,\}$.

3. Given $f \in mon(X)$, we define $clo(f) \stackrel{\text{def}}{=} \lambda x.\, \text{lfp}\, \lambda y. x \sqcup f(y)$.[1] Prove that $clo(f)$ is an upper closure operator, and that it is the smallest one greater than f for $\leq$.

   Application: compute the closure $clo(f)$ for the following function f on $\mathcal{P}(\mathbb{Z})$:

$$f(x) \stackrel{\text{def}}{=} \begin{cases} \emptyset & \text{if } x = \emptyset \\ [0, +\infty[& \text{if } \forall v \in x : v \geq 0 \\ ]-\infty, 0] & \text{if } \forall v \in x : v < 0 \\ \mathbb{Z} & \text{otherwise} \end{cases}$$

[1] $\lambda x.e$ denotes the function that associates the value of e to x. lfp f is the least fixpoint of f.

4. Prove that $uco(X)$ forms a complete lattice $(uco(X), \leq, \vee_{uco}, \wedge,_{uco}, \perp_{uco}, \top_{uco})$. Give the definition of $\vee_{uco}$, $\wedge_{uco}$, $\perp_{uco}$, $\top_{uco}$.

   Application: given two closures ρ_1 and ρ_2 defined by their fixpoints $S_1 \stackrel{\text{def}}{=} \{\emptyset,]-\infty, 0], \{0\}, [0, +\infty[, \mathbb{Z}\}$ and $S_2 \stackrel{\text{def}}{=} \{\emptyset, 2\mathbb{Z}, 2\mathbb{Z}+1, \mathbb{Z}\}$, give their join and their meet upper closure operators.

5. Show that, even when ρ_1 and ρ_2 are upper closures, $\rho_1 \circ \rho_2$ is not necessary an upper closure. Prove however that $clo(\rho_1 \circ \rho_2) = \rho_1 \vee_{uco} \rho_2$.

Part II: Exercise

The goal of this exercise is to define the *template numeric abstract domain*, a restriction of polyhedra that generalizes intervals, zones, and octagons. We consider a set $\mathcal{V} \stackrel{\text{def}}{=} \{V_1, \ldots, V_n\}$ of n $\mathbb{R}-$valued variables. A concrete element will be seen indifferently as a subset of environments in $\mathcal{D} \stackrel{\text{def}}{=} \mathcal{P}(\mathcal{V} \to \mathbb{R})$ or a subset of a vector space in $\mathcal{P}(\mathbb{R}^n)$. Given a fixed $m \times n$ matrix $\mathbf{M}$ with an arbitrary number m of rows, the template abstract domain for $\mathbf{M}$ is $\mathcal{D}^\sharp \stackrel{\text{def}}{=} (\mathbb{R} \cup \{+\infty\})^m$ ordered point-wise. Each abstract element is a vector $\vec{\beta} \in (\mathbb{R} \cup \{+\infty\})^m$ and represents the set:

$$\gamma(\vec{\beta}) \stackrel{\text{def}}{=} \{\, \vec{x} \in \mathbb{R}^n \mid \mathbf{M}\vec{x} \leq \vec{\beta} \,\}$$

1. Show how to recover the interval and zone domains when $n = 3$ by suitable choices of $\mathbf{M}$.

2. Prove that there is a Galois connection and give the definition of the abstraction function.

3. Show that γ is not always injective. Recall that linear programming consists in computing

   $$LP(\langle \mathbf{A}, \vec{c} \rangle, \vec{v}) \stackrel{\text{def}}{=} \max \{\, \vec{x} \cdot \vec{v} \mid \mathbf{A}\vec{x} \leq \vec{c} \,\}$$

   for an arbitrary polyhedron $\langle \mathbf{A}, \vec{c} \rangle$ and vector $\vec{v}$. Show how to use LP to construct a normal form.

4. Give the optimal abstract version of the following operators:

   (a) set union $\cup$ and set intersection $\cap$;

   (b) non-deterministic assignments: $[\![\, V \leftarrow \mathbf{?} \,]\!]R \stackrel{\text{def}}{=} \{\, \rho[V \mapsto v] \mid \rho \in R,\ v \in \mathbb{R} \,\}$;

   (c) affine assignments: $[\![\, V \leftarrow a_0 + \sum_i a_i V_i \,]\!]R \stackrel{\text{def}}{=} \{\, \rho[V \mapsto a_0 + \sum_i a_i \times \rho(V_i)] \mid \rho \in R \,\}$;

   (d) affine tests: $[\![\, a_0 + \sum_i a_i V_i \leq 0? \,]\!]R \stackrel{\text{def}}{=} \{\, \rho \in R \mid a_0 + \sum_i a_i \times \rho(V_i) \leq 0 \,\}$.

   Which operators are exact?

5. Propose a widening $\triangledown$. Prove its soundness and termination.

Part III: Problem

We consider a simple programming language whose syntax obeys the following grammar:

$$\begin{array}{rcl@{\qquad\qquad}l}
P & ::= & V \leftarrow \mathbf{?} & \text{where} \quad e ::= a_0 + \sum_i a_i V_i,\ V_i \in \mathcal{V},\ a_i \in \mathbb{Q} \\
 & | & V \leftarrow e & \text{and} \quad\ \ \bowtie\, \in \{\, \geq, > \,\} \\
 & | & e \bowtie 0\,? & \\
 & | & \textbf{if } e \geq 0 \textbf{ then } \{\, P \,\} & \\
 & | & \textbf{while } e \geq 0 \textbf{ do } \{\, P \,\} & \\
 & | & P; P &
\end{array}$$

which contains non-deterministic assignments of a random value $V \leftarrow \mathbf{?}$, affine assignments $V \leftarrow e$, affine assertions $e \bowtie 0\,?$, as well as conditionals **if** $e \geq 0$ **then** $\{\, P \,\}$ and loops **while** $e \geq 0$ **do** $\{\, P \,\}$ guarded by affine tests, and instruction sequence $P; P$. Variables V, V_i range in a fixed finite set $\mathcal{V}$, constants a_i are rational, and assertion tests may be strict $>$ or non-strict $\geq$.

Forward semantics. A program state ρ is a map associating a rational value $\rho(V) \in \mathbb{Q}$ to each variable $V \in \mathcal{V}$. We denote by $\mathcal{E}$ the set of possible program states: $\mathcal{E} \stackrel{\text{def}}{=} \mathcal{V} \to \mathbb{Q}$. The forward semantics in denotational form $[\![\, P \,]\!]$ of a program P is a function $[\![\, P \,]\!] : \mathcal{P}(\mathcal{E}) \to \mathcal{P}(\mathcal{E})$ that associates to a set of states at the program entry (input states), the set of states at the program exit (output states). It is defined by induction on the syntax of P as:[2]

$$\begin{array}{lcl}
[\![\, V \leftarrow \mathbf{?} \,]\!]E & \stackrel{\text{def}}{=} & \{\, \rho[V \mapsto v] \mid \rho \in E,\, v \in \mathbb{Q} \,\} \\
[\![\, V \leftarrow a_0 + \sum_i a_i V_i \,]\!]E & \stackrel{\text{def}}{=} & \{\, \rho[V \mapsto a_0 + \sum_i a_i \times \rho(V_i)] \mid \rho \in E \,\} \\
[\![\, a_0 + \sum_i a_i V_i \bowtie 0\,? \,]\!]E & \stackrel{\text{def}}{=} & \{\, \rho \in E \mid a_0 + \sum_i a_i \times \rho(V_i) \bowtie 0 \,\} \\
[\![\, \textbf{if } e \geq 0 \textbf{ then } \{\, P \,\} \,]\!]E & \stackrel{\text{def}}{=} & [\![\, P \,]\!]([\![\, e \geq 0\,? \,]\!]E) \cup [\![\, e < 0\,? \,]\!]E \\
[\![\, \textbf{while } e \geq 0 \textbf{ do } \{\, P \,\} \,]\!]E & \stackrel{\text{def}}{=} & [\![\, e < 0\,? \,]\!](\mathrm{lfp}_E\, \lambda R.(R \cup [\![\, P \,]\!]([\![\, e \geq 0\,? \,]\!]R))) \\
[\![\, P_1; P_2 \,]\!] & \stackrel{\text{def}}{=} & [\![\, P_2 \,]\!] \circ [\![\, P_1 \,]\!]
\end{array}$$

Question 1

⋆ Prove that the fixpoint used in $[\![\, \textbf{while } e \geq 0 \textbf{ do } \{\, P \,\} \,]\!]$ is well-defined and that, for any program P, $[\![\, P \,]\!]$ is a complete $\cup-$morphism.[3]

Backward semantics. The goal of the problem is to infer, given a set of output states O, a set of input states I as large as possible such that $[\![\, P \,]\!]I \subseteq O$. Given a function $f : \mathcal{P}(\mathcal{E}) \to \mathcal{P}(\mathcal{E})$, we define its *backward* version $\overleftarrow{f} : \mathcal{P}(\mathcal{E}) \to \mathcal{P}(\mathcal{E})$ as follows:

$$\overleftarrow{f}(x) \stackrel{\text{def}}{=} \{\, y \in \mathcal{E} \mid f(\{\, y \,\}) \subseteq x \,\} \ .$$

Question 2

⋆ Assuming that f and g are complete $\cup-$morphism on $\mathcal{P}(\mathcal{E})$, prove the following properties:

1. $\overleftarrow{f}$ is monotonic and a complete $\cap-$morphism.
2. The pair $(f, \overleftarrow{f})$ forms a Galois connection: $\mathcal{P}(\mathcal{E}) \underset{f}{\overset{\overleftarrow{f}}{\leftrightarrows}} \mathcal{P}(\mathcal{E})$.
3. $\overleftarrow{f \circ g} = \overleftarrow{g} \circ \overleftarrow{f}$.
4. $\overleftarrow{\lambda x.(f(x) \cup g(x))} = \lambda x.(\overleftarrow{f}(x) \cap \overleftarrow{g}(x))$.
5. $\overleftarrow{\lambda x.(\mathrm{lfp}_x\, \lambda z.(z \cup f(z)))} = \lambda x.(\mathrm{gfp}_x\, \lambda z.(z \cap \overleftarrow{f}(z)))$, where $\mathrm{gfp}_x\, g$ is the greatest fixpoint of g smaller than x.

⋆ Do these properties hold when f and g are not complete $\cup-$morphism?
⋆ Is $\overleftarrow{f}$ always strict (i.e., $\overleftarrow{f}(\emptyset) = \emptyset$)?

[2] $\mathrm{lfp}_x\, f$ is the least (with respect to set inclusion $\subseteq$) fixpoint greater than x of the function f.

[3] f is a complete $\cup-$morphism if, for any family $(A_i)_{i \in I}$ of sets, $f(\cup_{i \in I} A_i) = \cup_{i \in I} f(A_i)$. It is a complete $\cap-$morphism if $f(\cap_{i \in I} A_i) = \cap_{i \in I} f(A_i)$.

Question 3

⋆ Deduce from the preceding question and the definition of $[\![P]\!]$ a definition of $\overleftarrow{[\![P]\!]}$ by induction on the syntax of P.

⋆ Prove that $\overleftarrow{[\![P]\!]}O$ is the maximal set I such that $[\![P]\!]I \subseteq O$.

Approximation. We now consider the approximate computation of $\overleftarrow{[\![P]\!]}$ in the abstract domain $\mathcal{D}^\sharp$ of closed, convex *polyhedra*. We note by $\gamma : \mathcal{D}^\sharp \to \mathcal{P}(\mathcal{E})$ the polyhedral concretization function. The polyhedral abstraction of $\overleftarrow{[\![P]\!]}$ we construct will be denoted by $\overleftarrow{[\![P]\!]}^\sharp$.

Question 4

⋆ Prove that, if $E \in \mathcal{P}(\mathcal{E})$ is a convex and closed set, then $\overleftarrow{[\![V \leftarrow \mathbf{?}]\!]}E$ is either $\emptyset$ or E.

⋆ Deduce an (exact) polyhedral abstraction $\overleftarrow{[\![V \leftarrow \mathbf{?}]\!]}^\sharp$.

Question 5

⋆ Show that assertions $\overleftarrow{[\![e \geq 0\,?]\!]}$ cannot be abstracted exactly on polyhera in general.

⋆ Justify that, in order to obtain a meaningful result, an abstraction $\overleftarrow{[\![P]\!]}^\sharp$ of a concrete semantic function $\overleftarrow{[\![P]\!]}$ must satisfy the following soundness relation: $\forall X^\sharp \in \mathcal{D}^\sharp : \gamma(\overleftarrow{[\![P]\!]}^\sharp X^\sharp) \subseteq \overleftarrow{[\![P]\!]}(\gamma(X^\sharp))$.

⋆ Given a polyhedron P described by a set of constraints, show how to construct sound abstractions $\overleftarrow{[\![e \geq 0\,?]\!]}^\sharp$ and $\overleftarrow{[\![e > 0\,?]\!]}^\sharp$ by removing the constraint $e \geq 0$ in P as well as all the constraints that are redundant with it in P.[4] Prove the soundness of your abstract operators.

Question 6

Propose a polyhedral abstraction of assignments $\overleftarrow{[\![V \leftarrow a_0 + \sum_i a_i V_i]\!]}^\sharp$.

Question 7

A *lower widening* is a binary operator $\underline{\triangledown}$ on $\mathcal{D}^\sharp$ such that:

1. $\gamma(X^\sharp \,\underline{\triangledown}\, Y^\sharp) \subseteq \gamma(X^\sharp) \cap \gamma(Y^\sharp)$
2. For any sequence $(X_n^\sharp)_{n \in \mathbb{N}}$, the sequence $Y_0^\sharp \overset{\text{def}}{=} X_0^\sharp$, $Y_{n+1}^\sharp = Y_n^\sharp \,\underline{\triangledown}\, X_{n+1}^\sharp$ has a stable value: $\exists i \in \mathbb{N} : Y_i^\sharp = Y_{i+1}^\sharp$.

⋆ Show how a lower widening operator can be used to compute $\overleftarrow{[\![\textbf{while } e \geq 0 \textbf{ do } \{ P \}]\!]}^\sharp$.

⋆ Propose a lower widening operator on polyhedra.

[4] A constraint c is redundant with a constraint $d \in P$ if $\gamma(P) = \gamma(P \setminus d \cup \{ c \})$.