Written exam
MPRI 2-6, year 2014–2015

Antoine Miné

3 December 2014

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 in French.
The different parts of 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 an ambiguity or an error in the definitions or questions, it is part of the exam to correct them and answer to the best of your abilities.

Part I: Problem

This problem studies several ways to extent the numeric static analyses seen in the course to programs with arrays.

Syntax. Consider the programming language extending the language used in the course:

$$
\begin{array}{rcll}
P & ::= & V \leftarrow e & \\
 & | & A[e] \leftarrow e' & \\
 & | & V \leftarrow A[e] & \\
 & | & \textbf{if } e \bowtie 0 \textbf{ then } P \textbf{ fi} & \\
 & | & \textbf{while } e \bowtie 0 \textbf{ do } P \textbf{ od} & \\
 & | & P; P' & \bowtie \in \{<, >, \leq, \geq, =, \neq\}
\end{array}
\qquad
\begin{array}{rcll}
e & ::= & [c,d] & c \in \mathbb{Z} \cup \{-\infty\}, d \in \mathbb{Z} \cup \{+\infty\}, c \leq d \\
 & | & V & V \in \mathbb{V} \\
 & | & -e & \\
 & | & e \diamond e' & \diamond \in \{+, -, \times\}
\end{array}
$$

Variables V range in a finite set $\mathbb{V}$ of (scalar) variables and there is a *single array* A. The scalar variables and the array hold integer values. We assume that the size of A is denoted by a special variable $N \in \mathbb{V}$, and that $N \geq 2$. Note the special assignments from the array $V \leftarrow A[e]$ and into the array $A[e] \leftarrow e'$. The other statements are standard: scalar assignments $V \leftarrow e$, tests **if** $e \bowtie 0$ **then** P **fi**, loops **while** $e \bowtie 0$ **do** P **od**, sequence $P; P'$. Note that expressions e used in tests, in right-hand parts of assignments and as array indices can only feature variables in $\mathbb{V}$, not the array A; hence, assignments feature at most one array access (either a read or a write), and other statements have no array access.

Concrete semantics. A concrete environment $(\rho, a) \in \mathcal{E}$ maps each scalar variable $V \in \mathbb{V}$ to an integer $\rho(V)$, and each integer index in $i \in [0, \rho(N) - 1]$ to an integer array element $a(i)$:

$$\mathcal{E} \stackrel{\text{def}}{=} \{ (\rho, a) \mid \rho \in \mathbb{V} \to \mathbb{Z},\ \rho(N) \geq 2,\ a \in [0, \rho(N) - 1] \to \mathbb{Z} \} \ .$$

The concrete semantics $\mathsf{E}[\![e]\!] : \mathcal{E} \to \mathcal{P}(\mathbb{Z})$ and $\mathsf{C}[\![P]\!] : \mathcal{P}(\mathcal{E}) \to \mathcal{P}(\mathcal{E})$ is defined as usual for expressions e and for non-array statements P:

$$
\begin{array}{ll}
\mathsf{E}[\![V]\!] \rho \stackrel{\text{def}}{=} \{ \rho(V) \} & \mathsf{E}[\![[c,d]]\!] \rho \stackrel{\text{def}}{=} \{ v \in \mathbb{Z} \mid c \leq v \leq d \} \\
\mathsf{E}[\![-e]\!] \rho \stackrel{\text{def}}{=} \{ -v \mid v \in \mathsf{E}[\![e]\!] \} & \mathsf{E}[\![e \diamond e']\!] \rho \stackrel{\text{def}}{=} \{ v \diamond v' \mid v \in \mathsf{E}[\![e]\!] \rho,\ v' \in \mathsf{E}[\![e']\!] \rho \}
\end{array}
$$

$$
\begin{array}{lcl}
\mathsf{C}[\![V \leftarrow e]\!]\, R & \stackrel{\text{def}}{=} & \{\, (\rho[V \mapsto v], a) \mid (\rho, a) \in R,\, v \in \mathsf{E}[\![e]\!]\, \rho \,\} \\
\mathsf{C}[\![e \bowtie 0?]\!]\, R & \stackrel{\text{def}}{=} & \{\, (\rho, a) \in R \mid \exists v \in \mathsf{E}[\![e]\!]\, \rho : v \bowtie 0 \,\} \\
\mathsf{C}[\![\textbf{if } e \bowtie 0 \textbf{ then } P \textbf{ fi}]\!]\, R & \stackrel{\text{def}}{=} & \mathsf{C}[\![P]\!]\, (\mathsf{C}[\![e \bowtie 0?]\!]\, R) \cup \mathsf{C}[\![e \not\bowtie 0?]\!]\, R \\
\mathsf{C}[\![\textbf{while } e \bowtie 0 \textbf{ do } P \textbf{ od}]\!]\, R & \stackrel{\text{def}}{=} & \mathsf{C}[\![e \not\bowtie 0?]\!]\, (\mathrm{lfp}\, \lambda X. R \cup \mathsf{C}[\![P]\!]\, (\mathsf{C}[\![e \bowtie 0?]\!]\, X)) \\
\mathsf{C}[\![P; P']\!]\, R & \stackrel{\text{def}}{=} & \mathsf{C}[\![P']\!]\, (\mathsf{C}[\![P]\!]\, R)
\end{array}
$$

For array assignments, we define:

$$
\begin{array}{lcl}
\mathsf{C}[\![V \leftarrow A[e]]\!]\, R & \stackrel{\text{def}}{=} & \{\, (\rho[V \mapsto a(i)], a) \mid (\rho, a) \in R,\, i \in \mathsf{E}[\![e]\!]\, \rho,\, i \in [0, \rho(N) - 1] \,\} \\
\mathsf{C}[\![A[e] \leftarrow e']\!]\, R & \stackrel{\text{def}}{=} & \{\, (\rho, a[i \mapsto v]) \mid (\rho, a) \in R,\, i \in \mathsf{E}[\![e]\!]\, \rho,\, v \in \mathsf{E}[\![e']\!]\, \rho,\, i \in [0, \rho(N) - 1] \,\}
\end{array}
$$

Note that fetching an array index out of the array bounds stops the program.

Uniform abstraction. A first abstraction consists in replacing the array value map $[0, \rho(N) - 1] \to \mathbb{Z}$ with a single scalar integer representing all the possible array values; hence, we use:

$$\mathcal{E}^U \stackrel{\text{def}}{=} (\mathbb{V} \to \mathbb{Z}) \times \mathbb{Z} \ .$$

The uniform abstraction function $\alpha^U : \mathcal{P}(\mathcal{E}) \to \mathcal{P}(\mathcal{E}^U)$ is:

$$\alpha^U(R) \stackrel{\text{def}}{=} \{\, (\rho, a(i)) \mid (\rho, a) \in R,\, i \in [0, \rho(N) - 1] \,\} \ .$$

Question 1.

1. Give a concretization γ^U such that $\mathcal{P}(\mathcal{E}) \underset{\alpha^U}{\overset{\gamma^U}{\leftrightarrows}} \mathcal{P}(\mathcal{E}^U)$ is a Galois connection (prove the Galois connection property).
2. Show by an example that α^U can indeed result in a loss of precision.
3. Give a condition on R under which $\alpha^U(R)$ does *not* lose any precision.

Question 2.

1. Give the best uniform abstraction $\mathsf{C}^U[\![P]\!] : \mathcal{P}(\mathcal{E}^U) \to \mathcal{P}(\mathcal{E}^U)$ for the array access assignments $A[e] \leftarrow e'$ and $V \leftarrow A[e]$ (prove the optimality). You have to be careful about the case where an out-of-bound array access occurs. For instance, given $R^\sharp \stackrel{\text{def}}{=} \{\, ([V \mapsto 0, N \mapsto 2], 0) \,\} \subseteq \mathcal{E}^U$ denoting a 0-filled array of size 2, we have $\mathsf{C}^U[\![V \leftarrow A[2]]\!]\, R^\sharp = \emptyset$.
2. Give the best abstraction $\cup^U$ of the join $\cup$ (prove the optimality).
3. Are all these operators exact or not (give either a proof of exactness or a counter-example)?

Question 3. Consider the following program that fills an array with ones:

$$P_1 \stackrel{\text{def}}{=} \textbf{while } I < N \textbf{ do } A[I] \leftarrow 1; I \leftarrow I + 1 \textbf{ od} \tag{1}$$

starting in a concrete environment where the array and I are 0-initialized:

$$\mathcal{I} \stackrel{\text{def}}{=} \{\, ([I \mapsto 0, N \mapsto n], a) \mid n \geq 2,\, \forall i \in [0, n-1] : a(i) = 0 \,\} \subseteq \mathcal{E} \ . \tag{2}$$

1. Give the concrete environment $\mathcal{F} \stackrel{\text{def}}{=} \mathsf{C}[\![P_1]\!]\, \mathcal{I} \subseteq \mathcal{E}$ at the end of program.
2. Show that the abstractions $\mathcal{I}^U \stackrel{\text{def}}{=} \alpha^U(\mathcal{I})$ and $\mathcal{F}^U \stackrel{\text{def}}{=} \alpha^U(\mathcal{F})$ of $\mathcal{I}$ and $\mathcal{F}$ in $\mathcal{P}(\mathcal{E}^U)$ do not lose any precision.
3. Give the uniform abstract semantics $\mathsf{C}^U[\![P_1]\!]\, \mathcal{I}^U$ computed by induction in $\mathcal{P}(\mathcal{E}^U)$. Why is it coarser than $\mathcal{F}^U$?

Non-relational abstractions. We now abstract $\mathcal{P}(\mathcal{E}^U)$ further using the interval domain. Given the set of intervals $\mathbb{I} \stackrel{\text{def}}{=} \{\, [a,b] \mid a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\}, a \leq b \,\}$, our abstract environments live in $\mathcal{D}^I \stackrel{\text{def}}{=} ((\mathbb{V} \cup \{\mathcal{A}\}) \to \mathbb{I}) \cup \{\bot\}$, i.e., we map each program variable as well as the extra variable $\mathcal{A}$ (denoting the contents of the array A) to an interval. The concretization $\gamma^{U,I}$ from $\mathcal{D}^I$ to $\mathcal{P}(\mathcal{E}^U)$ is:

$$\gamma^{U,I}(R^\sharp) \stackrel{\text{def}}{=} \{\, (\rho, a) \in \mathcal{E}^U \mid \forall V \in \mathbb{V} : \rho(V) \in R^\sharp(V),\, a \in R^\sharp(\mathcal{A}) \,\} \text{ if } R^\sharp \neq \bot,\ \emptyset \text{ otherwise } .$$

Question 4. Assuming that we already know (as seen in the course) interval abstractions $\mathsf{E}^I[\![\, e \,]\!]$, $\mathsf{C}^I[\![\, P \,]\!]$, and $\cup^I$ for non-array programs, propose interval abstractions of the uniform semantics for the array operations: $\mathsf{C}^{U,I}[\![\, A[e] \leftarrow e' \,]\!]$ and $\mathsf{C}^{U,I}[\![\, V \leftarrow A[e] \,]\!]$. Justify the soundness and possibly the optimality of your operators. Be careful in particular to handle precisely the case where an out-of-bound array access occurs.

Question 5. Consider the following program that stores the sequence $1, \ldots, N$ into the array:

$$P_2 \stackrel{\text{def}}{=} \textbf{while } I < N \textbf{ do } A[I] \leftarrow I + 1; I \leftarrow I + 1 \textbf{ od} \tag{3}$$

with the same initial state $\mathcal{I}$ as in (2): both A and I are 0-initialized.
Give the concrete semantics $\mathsf{C}[\![\, P_2 \,]\!]$ as well as the uniform semantics $\mathsf{C}^U[\![\, P_2 \,]\!]$ and its interval abstraction $\mathsf{C}^{U,I}[\![\, P_2 \,]\!]$.
Why are the abstract semantics imprecise?

Relational abstractions. In order to solve the precision issues in P_1, we now propose a new abstraction of $\mathcal{P}(\mathcal{E}^U)$ using relational domains (namely, polyhedra). In this part, we assimilate $\mathcal{E}^U = (\mathbb{V} \to \mathbb{Z}) \times \mathbb{Z}$ to $(\mathbb{V} \cup \{\mathcal{A}\}) \to \mathbb{Z}$. We will need to temporarily add more variables to $\mathbb{V} \cup \{\mathcal{A}\}$, and we define the following functions:

$$\begin{array}{lcl}
\mathsf{C}^U[\![\, \textbf{add } W \,]\!]\, R & \stackrel{\text{def}}{=} & \{\, \rho \oplus [W \mapsto v] \mid \rho \in R,\, v \in \mathbb{Z} \,\} \\
\mathsf{C}^U[\![\, \textbf{remove } W \,]\!]\, R & \stackrel{\text{def}}{=} & \{\, \rho \mid \exists v \in \mathbb{Z} : \rho \oplus [W \mapsto v] \in R \,\} \\
\mathsf{C}^U[\![\, \textbf{expand } V \mapsto W \,]\!]\, R & \stackrel{\text{def}}{=} & \{\, \rho \oplus [W \mapsto v] \mid \rho \in R,\, \rho[V \mapsto v] \in R \,\} \\
\mathsf{C}^U[\![\, \textbf{fold } V \hookleftarrow W \,]\!]\, R & \stackrel{\text{def}}{=} & \{\, \rho \mid \exists v \in \mathbb{Z} : \rho \oplus [W \mapsto v] \in R \,\} \cup \\
 & & \{\, \rho[V \mapsto v] \mid \rho \oplus [W \mapsto v] \in R \,\}
\end{array}$$

where $\rho \oplus [W \mapsto v]$ extends a map ρ where W is not defined to a map defined also on W. In **add** W, the new variable W is not initialized while, in **expand** $V \mapsto W$, it is initialized using the value of V in another environment that coincides on all other variables $V' \neq V$. In **remove** W, the value of W is forgotten while, in **fold** $V \hookleftarrow W$, it is stored as an alternate value for V.

Question 6.

1. Prove that $\mathsf{C}^U[\![\, V \leftarrow A[e] \,]\!]$ can be soundly replaced with:

   $$\mathsf{C}^U[\![\, \textbf{expand } \mathcal{A} \mapsto \mathcal{B}; V \leftarrow \mathcal{B}; \textbf{remove } \mathcal{B} \,]\!]$$

   where $\mathcal{B} \notin \mathbb{V} \cup \{\mathcal{A}\}$ is a fresh temporary variable.

2. Is this approximation exact?

3. Show on a counter-example that $\mathsf{C}^U[\![\, \textbf{expand } \mathcal{A} \mapsto \mathcal{B} \,]\!] = \mathsf{C}^U[\![\, \textbf{add } \mathcal{B}; \mathcal{B} \leftarrow \mathcal{A} \,]\!]$ does not always hold, and moreover that $\mathsf{C}^U[\![\, V \leftarrow \mathcal{A} \,]\!]$ is *not* a sound abstraction of $\mathsf{C}^U[\![\, V \leftarrow A[e] \,]\!]$.

Question 7. Similarly, propose a sound expression of $\mathsf{C}^U[\![A[e] \leftarrow e']\!]$ using only the operators **add** W, **fold** $V \hookleftarrow W$, as well as regular assignments (prove the soundness and discuss the exactness).

Question 8. We now consider abstracting the uniform semantics in the polyhedra abstract domain. Based on the polyhedra operations seen in the course, propose polyhedral abstractions $\mathsf{C}^{U,P}[\![\textbf{add } W]\!]$, $\mathsf{C}^{U,P}[\![\textbf{remove } W]\!]$, $\mathsf{C}^{U,P}[\![\textbf{expand } V \mapsto W]\!]$, $\mathsf{C}^{U,P}[\![\textbf{fold } V \hookleftarrow W]\!]$ (justify the soundness and discuss the optimality of your abstract operations).

Question 9. Give the polyhedra semantics $\mathsf{C}^{U,P}[\![P_1]\!]$ and $\mathsf{C}^{U,P}[\![P_2]\!]$ of the programs P_1 from (1) and P_2 from (3).

Parametric predicate abstractions. In this last part, instead of representing the array contents with a single variable $\mathcal{A}$, we use a *logic predicate* that states some properties about the array contents. To handle P_1 from (1), we use the predicate $one(L, H)$ denoting the partial initialization of A to 1: $\forall i \in [L, H] : a(i) = 1$. Now, environments live in

$$\mathcal{E}^{one} \stackrel{\text{def}}{=} (\mathbb{V} \cup \{L, H\}) \to \mathbb{Z}$$

and assign values to program variables in $\mathbb{V}$ and predicate variables in $\{L, H\}$. We have the following concretization $\gamma^{one} : \mathcal{P}(\mathcal{E}^{one}) \to \mathcal{P}(\mathcal{E})$:

$$\gamma^{one}(R^\sharp) \stackrel{\text{def}}{=} \{ (\rho, a) \in \mathcal{E} \mid \exists l, h \in \mathbb{Z} : \rho \oplus [L \mapsto l, H \mapsto h] \in R^\sharp, \forall i \in [l, h] : a(i) = 1 \} \ .$$

Question 10.

1. Give abstractions in $\mathcal{P}(\mathcal{E}^{one})$ of the following operators used in the semantics of P_1: $\mathsf{C}^{one}[\![A[I] \leftarrow 1]\!]$, $\mathsf{C}^{one}[\![I \leftarrow I + 1]\!]$, $\mathsf{C}^{one}[\![I < N?]\!]$, and the join $\cup^{one}$ (prove their soundness and discuss their optimality).

2. Give the abstract semantics $\mathsf{C}^{one}[\![P_1]\!]\mathcal{I}^{one}$ of P_1 (1) in $\mathcal{P}(\mathcal{E}^{one})$, with initial set $\mathcal{I}^{one} \stackrel{\text{def}}{=} \{ [I \mapsto 0, N \mapsto n, L \mapsto 0, H \mapsto -1] \mid n \geq 2 \}$ (setting $L > H$ models the fact that A does not contain any 1 when the program starts).

Question 11. We now abstract $\mathcal{P}(\mathcal{E}^{one})$ further using the polyhedra domain over $\mathbb{V} \cup \{L, H\}$.

1. Give polyhedral abstractions $\mathsf{C}^{P,one}[\![A[I] \leftarrow I]\!]$, $\mathsf{C}^{P,one}[\![I \leftarrow I+1]\!]$, $\mathsf{C}^{P,one}[\![I < N?]\!]$, and $\cup^{P,one}$ of the operators defined in the previous question.

2. Give the polyhedra semantics $\mathsf{C}^{P,one}[\![P_1]\!]$ of the program P_1 from (1).

Question 12. Consider now the following program that computes the maximum of the array:

$$\begin{aligned} P_3 \stackrel{\text{def}}{=} \quad & V \leftarrow A[0]; I \leftarrow 1; \\ & \textbf{while } I < N \textbf{ do } X \leftarrow A[I]; \textbf{if } X > V \textbf{ then } V \leftarrow X \textbf{ fi}; I \leftarrow I + 1 \textbf{ od} \end{aligned}$$

Propose a predicate domain in the spirit of $\mathcal{P}(\mathcal{E}^{one})$, but able to prove that, at the end of P_3, V equals the maximum of the array.
Give the elements of the domain, their concretization, and sound abstract transfer functions sufficiently precise to prove the property on P_3.

Part II: Exercise

Let $(X, \sqsubseteq, \sqcup, \sqcap, \bot, \top)$ be a complete lattice. We say that the subset $M \subseteq X$ is a lower Moore family if and only if $\forall x \in X : \{\, y \in M \mid x \sqsubseteq y \,\}$ is not empty and has a least element in M.

1. Prove that: M is a lower Moore family if and only if it contains X's greatest element $\top$ and it is closed by the meet $\sqcap$ of X (i.e., $\forall S \subseteq M : \sqcap S \in M$).

2. Recall that $\rho \in X \to X$ is an upper closure operator if and only if it is monotonic $\forall x \sqsubseteq y : \rho(X) \sqsubseteq \rho(Y)$, extensive $\forall x : x \sqsubseteq \rho(x)$, and idempotent $\forall x : \rho(\rho(x)) = \rho(x)$.
   Prove that: a set $M \subseteq X$ is a lower Moore family if and only if there exists an upper closure operator ρ such that $M = \{\, \rho(x) \mid x \in X \,\}$.

3. Consider the powerset concrete lattice $\mathcal{P}(\mathbb{Z})$ partially ordered by set inclusion $\subseteq$ and the abstract set $X^\sharp \stackrel{\text{def}}{=} \{\, \emptyset, \mathbb{Z}, \{\, x \mid x \geq 0 \,\}, \{\, x \mid x \leq 0 \,\} \,\}$, also ordered by $\subseteq$.
   Show that $X^\sharp$ is *not* a Moore family.
   What does it tell us about the existence of best approximations of the concrete elements from $\mathcal{P}(\mathbb{Z})$ in the abstract domain $X^\sharp$?

4. Show that this can be corrected by *adding* some elements to $X^\sharp$ (give an example).
   Show that this can be also corrected by *removing* some elements from $X^\sharp$ (give an example).

Part III: Exercise

Consider the classic interval widening:

$$[a,b] \triangledown [c,d] \stackrel{\text{def}}{=} \left[\begin{cases} a & \text{if } a \leq c \\ -\infty & \text{if } a > c \end{cases}, \begin{cases} b & \text{if } b \geq d \\ +\infty & \text{if } b < d \end{cases} \right]$$

Note that $\triangledown$ is *stable*, i.e., $y \sqsubseteq x \implies x \triangledown y = x$. The widening will stop iterating as soon as a post-fixpoint is reached.

1. Is $\triangledown$ monotonic in its first argument ($[a,b] \sqsubseteq [a',b'] \implies [a,b] \triangledown [c,d] \sqsubseteq [a',b'] \triangledown [c,d]$)? In its second argument? (give a proof or a counter-example)

2. Consider the loop:

   $$P \stackrel{\text{def}}{=} \textbf{while } X \geq 0 \textbf{ do if } X < 10 \textbf{ then } X \leftarrow X + 1 \textbf{ else } X \leftarrow X - 1 \textbf{ fi od}$$

   starting in the environment where $X = 0$, analyzed in the interval domain with widening.
   Show the abstract iteration when iterating from the interval $[0, 0]$.
   Show that starting the iteration with a larger interval can lead to a more precise result (give an example).

3. Consider a domain $(\mathcal{D}, \sqsubseteq)$ equipped with a widening $\triangledown$ which is sound: $\forall x, y : x \sqsubseteq x \triangledown y$, $y \sqsubseteq x \triangledown y$, stable $\forall x, y : y \sqsubseteq x \implies x \triangledown y = x$, and monotonic in its first argument $\forall x, x', y : x \sqsubseteq x' \implies x \triangledown y \sqsubseteq x' \triangledown y$.
   Prove that iterations with widening stabilize in finite time only if $\mathcal{D}$ has no strictly infinite chain, i.e., a monotonic widening is not helpful to enforce termination.