Master Parisien de Recherche en Informatique
École normale supérieure

Année scolaire 2010/2011

Cours M.2-6
« Interprétation abstraite: applications à la vérification et à l'analyse statique »

Examen partiel et corrigé

Patrick Cousot

26 novembre 2010

Course and personal notes are the only allowed documents. It will not be answered to any question during the exam. If a question is ambiguous, imprecise or incorrect, it is part of the question to solve the ambiguity, imprecision or incorrectness by indicating all required hypotheses together with the solution, if any. All questions are independent and can be answered in any order.

Let us recall the following theorem:

Theorem 1 *If $\langle L, \leqslant, \bot\rangle$ is a cpo, $F \in L \to L$ is monotonically increasing, $\langle \overline{L}, \sqsubseteq\rangle$ is a poset, $\alpha \in L \to \overline{L}$ is continuous*[1,2]*, $\overline{F} \in \overline{L} \to \overline{L}$ commutes (resp. semi-commutes) with F that is $\alpha \circ F = \overline{F} \circ \alpha$*[3] *(resp. $\alpha \circ F \sqsubseteq \overline{F} \circ \alpha$) then $\alpha(\mathbf{lfp}^{\leqslant}_{\bot} F) = \mathbf{lfp}^{\sqsubseteq}_{\alpha(\bot)} \overline{F}$ (resp. $\alpha(\mathbf{lfp}^{\leqslant}_{\bot} F) \sqsubseteq \mathbf{lfp}^{\sqsubseteq}_{\alpha(\bot)} \overline{F}$).* ■

which may be useful in some questions.

Question 1

Sintzoff (1972) presents the rule of signs abstraction in the following way:

> "$a \times a + b \times b$ yields always the object "pos" when a and b are the objects "pos" or "neg", and when the valuation is defined as follows :
>
> $$\begin{array}{rcl@{\qquad}rcl}
\text{pos+pos} & = & \text{pos} & \text{pos} \times \text{pos} & = & \text{pos} \\
\text{pos+neg} & = & \text{pos,neg} & \text{pos} \times \text{neg} & = & \text{neg} \\
\text{neg+pos} & = & \text{pos,neg} & \text{neq} \times \text{pos} & = & \text{neg} \\
\text{neg+neg} & = & \text{neg} & \text{neg} \times \text{neg} & = & \text{pos} \\
\text{V(p+q)} & = & \text{V(p)+V(q)} & \text{V(p} \times \text{q)} & = & \text{V(p)} \times \text{V(q)} \\
\text{V(0)} & = & \text{V(1)} \;=\; \ldots \;=\; \text{pos} & & & \\
\text{V(-1)} & = & \text{V(-2)} \;=\; \ldots \;=\; \text{neg} & & &
\end{array}$$
>
> The valuation of $a \times a + b \times b$ yields "pos" by the following computation :

[1] α is *continuous* if and only if it preserves existing lubs of increasing chains.

[2] The continuity hypothesis for α can be restricted to the iterates $F^0 \triangleq \bot$, $F^{n+1} \triangleq F(F^n)$, $F^\omega \triangleq \bigsqcup_{n\geqslant} F^n$ of the least fixpoint of F.

[3] The commutation property $\alpha \circ F(x) = \overline{F} \circ \alpha(x)$ is only required for all $x \in L$ such that $\gamma \circ \alpha(x) \leqslant \mathbf{lfp}^{\leqslant} F$ or even just for the iterates of the least fixpoint of F.

$$\begin{array}{rcl@{\quad}rcl}
\mathrm{V}(a) & = & \text{pos,neg} & \mathrm{V}(b) & = & \text{pos,neg} \\
\mathrm{V}(a \times a) & = & \text{pos} \times \text{pos, neg} \times \text{neg} & \mathrm{V}(b \times b) & = & \text{pos} \times \text{pos, neg} \times \text{neg} \\
 & = & \text{pos,pos} \;=\; \text{pos} & & = & \text{pos,pos} \;=\; \text{pos}
\end{array}$$

$$\mathrm{V}(a \times a + b \times b) \;=\; \mathrm{V}(a \times a){+}\mathrm{V}(b \times b) \;=\; \text{pos+pos} \;=\; \text{pos''}$$

What is wrong about it?

Answer to question 1

We have “pos × neq = neq” with “V(0) = pos” and “V(-1) = V(-2) = … = neg” so V(0 × -1) = V(0) × V(-1) = pos × neq = neq, proving that $0 = 0 \times -1 < 0$!

Question 2

A multiplication $m \times n = r$ can be checked by summing the digits of integer m modulo 9, summing the digits of n modulo 9, and checking that their product modulo 9 is equal to the sum of the digits of the result r modulo 9. For example,

$$\begin{array}{rrcrcl}
 & 1234 & \rightarrow & 10 \bmod 9 & = & 1 \\
\times & 5678 & \rightarrow & 26 \bmod 9 & = & 8 \\
\hline
= & 7006652 & \rightarrow & 26 \bmod 9 & = & 8
\end{array}$$

succeeds, while

$$\begin{array}{rrcrclcl}
 & 1234 & \rightarrow & 10 \bmod 9 & = & 1 & & \\
\times & 5678 & \rightarrow & 26 \bmod 9 & = & 8 & & \\
\hline
= & 7006651 & \rightarrow & 27 \bmod 9 & = & 7 & \neq & 1 \times 8 \bmod 9
\end{array}$$

fails.

- Show that this *casting out nines* is an abstraction.
- Is it a proof[4] ?
- Can you cite a sound generalization of the idea used in program analysis?

Answer to question 2

- Casting out nines is a sound method of checking equations because of a property of modular arithmetic. Specifically, if n and n' (respectively, m and m') have the same remainder modulo 9, then so do $n \times m$ and $n' \times m'$. To compute n modulo 9, one observes that the sum of the digits of the decimal writing of an integer has the same remainder, modulo 9, as this integer. Of course all 9 digits in n and m can be cast out.
- This is an example of abstraction by over-approximation because the number is replaced by the set of all numbers which have the same remainder modulo 9. Of course a failure is a proof that $n \times m \neq r$.
- However a success is not a proof of correctness of the multiplication since changing m, n and r modulo 9 (e.g. by exchanging digits in the decimal representation or adding 9’s) yields the same false positive result.

The same reasoning is valid for other operations $+$, $-$, $/$, etc. A sound generalization of the idea isthe congruence abstraction used in program analysis.

[4] In French it is called a “proof by 9”.

Question 3

Define the *reflexive transitive closure* $r^\star$ of a relation $r \in \wp(S \times S)$ on a set S as $r^\star \triangleq \bigcup_{n \in \mathbb{N}} r^n$ where the *powers* r^n, $n \in \mathbb{N}$ are defined as $r^0 \triangleq \{\langle s,\ s\rangle \mid s \in S\}$ (which is the identity relation), $r^{n+1} \triangleq r \circ r^n$, and the *composition* of relations is $r \circ r' \triangleq \{\langle s,\ s''\rangle \in S \times S \mid \exists s' \in S : \langle s,\ s'\rangle \in r \wedge \langle s',\ s''\rangle \in r'\}$. Prove that the reflexive transitive closure $r^\star$ of the relation r is an abstraction of the partial trace semantics $\vec{r}$ of this relation r defined as

$$\begin{array}{rcl} \vec{r}^{\,n} & \triangleq & \{\pi \in S^n \mid \forall i \in [0, n-1] : \langle \pi_i,\ \pi_{i+1}\rangle \in r\}, \quad n > 0 \\ \vec{r} & \triangleq & \displaystyle\bigcup_{n=1}^{+\infty} \vec{r}^{\,n} \end{array}$$

Answer to question 3

Define

$$\alpha^\star(T) \quad \triangleq \quad \{\langle \pi_0,\ \pi_{n-1}\rangle \mid \pi \in T \wedge |\pi| = n\}$$

Let us show that $\alpha^\star(\vec{r}^{\,n}) = r^n$ be recurrence on n.

$$\begin{array}{rll} & \alpha^\star(\vec{r}^{\,1}) & \\ = & r^0 & \lbag \text{def. } \alpha^\star \text{ and } \vec{r}^{\,1} \rbag \end{array}$$

$$\begin{array}{rll} & \alpha^\star(\vec{r}^{\,n+1}) & \\ = & \{\langle \pi_0,\ \pi_n\rangle \mid \pi \in S^{n+1} \wedge \forall i \in [0, n] : \langle \pi_i,\ \pi_{i+1}\rangle \in r\} & \lbag \text{def. } \alpha^\star \text{ and } \vec{r}^{\,n+1} \rbag \\ = & \{\langle \sigma,\ \pi'_n\rangle \mid \langle \sigma,\ \pi'_0\rangle \in r \wedge \pi' \in S^n \wedge \forall i \in [0, n-1] : \langle \pi'_i,\ \pi'_{i+1}\rangle \in r\} & \\ & \qquad \lbag \text{letting } \pi = \sigma\pi' \rbag & \\ = & r \circ \{\langle \pi'_0,\ \pi'_n\rangle \mid \pi' \in S^n \wedge \forall i \in [0, n-1] : \langle \pi'_i,\ \pi'_{i+1}\rangle \in r\} & \lbag \text{def. } \circ \rbag \\ = & r \circ \alpha^\star(\vec{r}^{\,n+1}) & \lbag \text{def. } \alpha^\star \text{ and } \vec{r}^{\,1} \rbag \\ = & r \circ r^n & \lbag \text{ind. hyp.} \rbag \\ = & r^{n+1} & \lbag \text{def. } r^{n+1} \rbag \end{array}$$

It follows that

$$\begin{array}{rll} & \alpha^\star(\vec{r}) & \\ = & \alpha^\star(\displaystyle\bigcup_{n=1}^{+\infty} \vec{r}^{\,n}) & \lbag \text{def. } \vec{r} \rbag \\ = & \displaystyle\bigcup_{n=1}^{+\infty} \alpha^\star(\vec{r}^{\,n}) & \lbag \text{def. } \alpha^\star \rbag \\ = & \displaystyle\bigcup_{n=1}^{+\infty} r^n & \lbag \text{as shown above} \rbag \\ = & r^\star & \lbag \text{def. } r^\star \rbag \quad \blacksquare \end{array}$$

Question 4

Prove that

Theorem 2 *If $\langle \overline{L},\ \sqsubseteq,\ \top\rangle$ is a dcpo*[5]*, $\overline{F} \in \overline{L} \to \overline{L}$ is monotonically increasing, $\gamma \in \overline{L} \to L$ is co-continuous*[6]*, $F \in L \to L$ commutes with F that is $\gamma \circ \overline{F} = F \circ \gamma$ then $\gamma(\boldsymbol{gfp}^{\sqsubseteq}_{\top}\, \overline{F}) = \boldsymbol{gfp}^{\leqslant}_{\gamma(\top)}\, F$.* ■

Answer to question 4

By the dual of 1 (in particular since $\langle L,\ \leqslant\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \overline{L},\ \sqsubseteq\rangle$ implies $\langle \overline{L},\ \sqsupseteq\rangle \underset{\gamma}{\overset{\alpha}{\leftrightarrows}} \langle L,\ \geqslant\rangle$).

Question 5

Let $\langle L,\ \leqslant,\ \bot,\ \neg\rangle$ be a complete Boolean lattice (where $\neg$ is the unique complement). Prove that $\langle L,\ \leqslant\rangle \underset{\neg}{\overset{\neg}{\leftrightarrows}} \langle L,\ \geqslant\rangle$ is a Galois isomorphism.

Answer to question 5

For all $x, y \in L$, we have

$$
\begin{aligned}
& \neg x \geqslant y \\
\Leftrightarrow\ & \neg\neg x \leqslant \neg y \\
\Leftrightarrow\ & x \leqslant \neg y
\end{aligned}
$$

which is the definition of a Galois connection. $\neg$ is a isomorphism with inverse $\neg$ since $\neg\neg x = x$.

Question 6

Using Th. 1 and Q. 5, prove the following theorem due to David Park (1969)

Theorem 3 *If $F \in L \to L$ is monotonically increasing on a complete Boolean lattice $\langle L,\ \leqslant,\ \bot,\ \neg\rangle$ then $\neg\, \boldsymbol{lfp}^{\leqslant}_{\bot}\, F = \boldsymbol{gfp}^{\leqslant}_{\neg\bot}\, \neg \circ F \circ \neg$.* ■

Answer to question 6

By Th. 1, for $\langle L,\ \leqslant\rangle \underset{\neg}{\overset{\neg}{\leftrightarrows}} \langle L,\ \geqslant\rangle$, $\neg \circ \neg$ is the identity and $\mathbf{lfp}^{\geqslant}\, \overline{F} = \mathbf{gfp}^{\leqslant}\, \overline{F}$.

Question 7

Prove the following theorem providing a condition for fixpoints of an increasing map on a complete Boolean lattice to be unique.

Theorem 4 (D. Park) *Let $f \in \mathcal{L} \xrightarrow{\nearrow} \mathcal{L}$ be an increasing map on the complete Boolean lattice $\langle L,\ \sqsubseteq,\ \bot,\ \top,\ \sqcup,\ \sqcap,\ \neg\rangle$. Then*

(1) $\boldsymbol{lfp}\, \widetilde{f} \sqcap \boldsymbol{lfp}\, f = \bot$

(2) $(\boldsymbol{lfp}\, \widetilde{f} \sqcup \boldsymbol{lfp}\, f = \top) \quad\Leftrightarrow\quad (\boldsymbol{lfp}\, f = \boldsymbol{gfp}\, f)$

where $\widetilde{f} \triangleq \neg \circ f \circ \neg$. ■

[5] A dual complete partial order (dcpo) has glbs of decreasing chains.

[6] γ is *co-continuous* if and only if it preserves existing glbs of decreasing chains.

Answer to question 7

$$\begin{array}{lll}
(1) & \mathbf{lfp}\, f \sqsubseteq \mathbf{gfp}\, f & ⟅\text{Tarski's theorem}⟆ \\
\Rightarrow & \neg\mathbf{gfp}\, f \sqsubseteq \neg\mathbf{lfp}\, f & ⟅\text{conjugate in complete Boolean lattice}⟆ \\
\Rightarrow & \neg\mathbf{gfp}\, f \sqcap \neg\mathbf{lfp}\, f \sqsubseteq \neg\mathbf{lfp}\, f \sqcap \mathbf{lfp}\, f & ⟅\text{def. glb in a lattice}⟆ \\
\Rightarrow & \neg\mathbf{gfp}\, f \sqcap \neg\mathbf{lfp}\, f \sqsubseteq \bot & ⟅\text{def. complement}⟆ \\
\Rightarrow & \neg\mathbf{gfp}\, f \sqcap \neg\mathbf{lfp}\, f = \bot & ⟅\bot\text{ is the infimum}⟆ \\
\Rightarrow & \mathbf{lfp}\, \widetilde{f} \sqcap \mathbf{lfp}\, f = \bot & ⟅\text{theorem } \mathbf{4}⟆
\end{array}$$

$$\begin{array}{lll}
(2, \Leftarrow) & \top & \\
= & \neg\mathbf{lfp}\, f \sqcup \mathbf{lfp}\, f & ⟅\text{def. complement}⟆ \\
= & \neg\mathbf{gfp}\, f \sqcup \mathbf{lfp}\, f & ⟅\text{since } \mathbf{lfp}\, f = \mathbf{gfp}\, f⟆ \\
= & \mathbf{lfp}\, \widetilde{f} \sqcup \mathbf{lfp}\, f & ⟅\text{theorem } \mathbf{4}⟆
\end{array}$$

$$\begin{array}{lll}
(2, \Rightarrow) & \mathbf{lfp}\, \widetilde{f} \sqcup \mathbf{lfp}\, f = \top & ⟅\text{hypothesis}⟆ \\
\Rightarrow & \mathbf{lfp}\, f = \neg\mathbf{lfp}\, \widetilde{f} & ⟅\text{def. complement}⟆ \\
\Rightarrow & \mathbf{lfp}\, f = \mathbf{gfp}\, f & ⟅\text{Th. 3}⟆ \quad \blacksquare
\end{array}$$

Question 8

The interval analysis of the following program

$$\boldsymbol{P} \quad \triangleq \quad {}^{1}\texttt{x := 100 ; while } {}^{2}\texttt{(x =/= 0) do } {}^{3}\texttt{x := (x - 1); od}^{4}.$$

has the following interval equations

$$\left\{\begin{array}{lll}
X_1 & = & [\texttt{min\_int},\ \texttt{max\_int}]\} \\
X_2 & = & [100,\ 100] \sqcup (\!|\ X_3 = \emptyset\ ?\ \emptyset\ ⁝\ \texttt{let}\,[a,\ b] = X_3\ \texttt{in} \\
 & & \qquad [\max(a-1,\ \texttt{min\_int}), \max(b-1, \texttt{min\_int})]\ |\!)\} \\
X_3 & = & (X_2 \sqcap [\texttt{min\_int},\ -1]) \sqcup (X_2 \sqcap [1,\ \texttt{max\_int}]) \\
X_4 & = & X_2 \sqcap [0,\ 0]
\end{array}\right.$$

The resolution of the equations by iteration with widening/narrowing yields a rather imprecise result.

```
% ocamlc interval.ml intervalWidening.ml intervalNarrowing.ml \
? invariant.ml invariantWidening.ml invariantNarrowing.ml \
? transformerBounded.ml iterator.ml \
? reachability_narrowing_bounded.ml
% time ./a.out
 1:(-1073741824,1073741823) 2:(-1073741824,100) 3:(-1073741824,100) 4:(0,0)
0.000u 0.000s 0:00.00 0.0%      0+0k 0+0io 0pf+0w
%
```

Propose a refinement of the interval widening to improve the precision of the analysis.

Answer to question 8

The imprecision is due to the widening jumping over 0. The problem can be avoided by refining the widening with a threshold at 0.

$$
\begin{array}{rcl}
\emptyset \mathbin{\nabla} y & \triangleq & y \\
x \mathbin{\nabla} \emptyset & \triangleq & x \\
[a,\ b] \mathbin{\nabla} [c,\ d] & \triangleq & [\llparenthesis\, c < a \mathrel{?} \llparenthesis\, 0 \leqslant c \mathrel{?} 0 \mathrel{:} -\infty \,\rrparenthesis \mathrel{:} a \,\rrparenthesis, \\
 & & \llparenthesis\, d > b \mathrel{?} \llparenthesis\, b \geqslant 0 \mathrel{?} 0 \mathrel{:} +\infty \,\rrparenthesis \mathrel{:} b \,\rrparenthesis]
\end{array}
$$

Question 9

Consider the syntax of the `repeat` command is

$$
\begin{array}{rcll}
\boldsymbol{C} & \in & \mathbb{C}, & \text{commands} \\
\boldsymbol{C} & ::= & \ldots & \\
 & | & \texttt{repeat}\,\boldsymbol{C}\,\texttt{until}\,{}^{\ell}\boldsymbol{B} & \text{where } \ell \notin \mathbb{im}\llbracket \boldsymbol{C} \rrbracket \\
 & & & \text{and } \mathbb{im}\llbracket \texttt{repeat}\,\boldsymbol{C}\,\texttt{until}\,{}^{\ell}\boldsymbol{B} \rrbracket \triangleq \{\ell\} \cup \mathbb{im}\llbracket \boldsymbol{C} \rrbracket
\end{array}
$$

Execution of the `repeat` $\boldsymbol{C}$ `until` ${}^{\ell}\boldsymbol{B}$ command starts with that of the loop body $\boldsymbol{C}$

$$
\mathbb{i}\llbracket \texttt{repeat}\,\boldsymbol{C}\,\texttt{until}\,{}^{\ell}\boldsymbol{B} \rrbracket \quad \triangleq \quad \mathbb{i}\llbracket \boldsymbol{C} \rrbracket
$$

Execution of the loop body $\boldsymbol{C}$ ends at label ℓ just before evaluation of the condition $\boldsymbol{B}$

$$
\begin{array}{rcll}
\boldsymbol{C} & ::= & \ldots & \\
 & | & \texttt{repeat}\,\boldsymbol{C}_1\,\texttt{until}\,{}^{\ell}\boldsymbol{B} & \mathbb{f}\llbracket \texttt{repeat}\,\boldsymbol{C}\,\texttt{until}\,{}^{\ell}\boldsymbol{B} \rrbracket \triangleq \mathbb{f}\llbracket \boldsymbol{C} \rrbracket \\
 & & & \qquad\qquad\qquad \mathbb{f}\llbracket \boldsymbol{C}_1 \rrbracket \triangleq \ell
\end{array}
$$

Define the transitional semantics $\mathsf{T}\llbracket \texttt{repeat}\,\boldsymbol{C}\,\texttt{until}\,{}^{\ell}\boldsymbol{B} \rrbracket$ of the `repeat` command so that execution of the loop body $\boldsymbol{C}$ is repeated until the condition $\boldsymbol{B}$ is true.

Answer to question 9

$$
\begin{array}{l}
\mathsf{T}\llbracket \texttt{repeat}\,\boldsymbol{C}\,\texttt{until}\,{}^{\ell}\boldsymbol{B} \rrbracket \quad \triangleq \\
\qquad \{\langle \rho,\ \ell \rangle \longrightarrow \langle \rho,\ \mathbb{i}\llbracket \texttt{repeat}\,\boldsymbol{C}\,\texttt{until}\,{}^{\ell}\boldsymbol{B} \rrbracket \rangle \mid \mathfrak{false} \in \mathsf{B}\llbracket \boldsymbol{B} \rrbracket \rho\} \\
\quad \cup \{\langle \rho,\ \ell \rangle \longrightarrow \langle \rho,\ \mathbb{f}\llbracket[\texttt{repeat}\,\boldsymbol{C}\,\texttt{until}\,{}^{\ell}\boldsymbol{B} \rrbracket] \rangle \mid \mathfrak{true} \in \mathsf{B}\llbracket \boldsymbol{B} \rrbracket \rho\} \\
\quad \cup \mathsf{T}\llbracket \boldsymbol{C} \rrbracket
\end{array}
$$

Question 10

Let U be a universe and $F \in \wp(U) \xrightarrow{\nearrow} \wp(U)$ a $\subseteq$-increasing function on $\wp(U)$ defining $\mathbf{lfp}^{\emptyset}_{\subseteq} F \in \wp(U)$. What is the set R of inference rules such that the formal system $\langle U,\ R \rangle$ defines exactly the same set $\mathbf{lfp}^{\emptyset}_{\subseteq} F$?

Answer to question 10

Choose $R = \{\frac{P}{c} \mid P \in \wp(U) \wedge c \in F(P)\}$. The consequence operator is then F so $\langle U,\, R\rangle$ defines $\mathbf{lfp}^{\subseteq}_{\emptyset}\, F$. Let $\overline{F}$ be the consequence operator for R. We have

Proof

$$\begin{array}{rll}
& \overline{F}(X) & \\
= & \{c \mid \exists \dfrac{P}{c} \in R : P \subseteq X\} & \lbag\text{def. consequence operator } \overline{F} \text{ for } R.\rbag \\
= & \{c \mid \exists \dfrac{P}{c} \in \{\dfrac{P}{c} \mid P \in \wp(U) \wedge c \in F(P)\} : P \subseteq X\} & \lbag\text{def. } R\rbag \\
= & \{c \mid c \in F(P) \wedge P \subseteq X\} & \lbag\text{def. } \in\rbag \\
= & \bigcup\{F(P) \mid P \subseteq X\} & \lbag\text{def. } \cup\rbag \\
= & F(X) \quad \lbag\text{since } P \subseteq X \text{ implies } F(P) \subseteq F(X) \text{ since } F \text{ is increasing so } \bigcup\{F(P) \mid P \subseteq X\} \subseteq F(X) & \\
& \text{and inversely } X \subseteq X \text{ by reflexivity so } F(X) \in \{F(P) \mid P \subseteq X\} \text{ so } F(X) \subseteq \bigcup\{F(P) \mid P \subseteq X\}\rbag &
\end{array}$$

■

Question 11

Let us consider the following program $\boldsymbol{P}$.

```
$^1$x := ? ;
while $^2$(1 < x) do
  $^3$x := x - 2
 od$^4$.
```

Formally define the program property that once initialized the variable x keeps the same parity.

Answer to question 11

We state that on any prefix execution trace π the difference of any two values of x is even (but when control is at program point 1).

$$\{S \in \wp(\mathcal{S}^+) \mid \forall \pi \in S : \forall i, j \in \mathfrak{dom}(\pi) : (\mathfrak{l}(\pi_i) \neq 1 \wedge \mathfrak{l}(\pi_j) \neq 1) \Rightarrow \\ (\exists k \in \mathbb{Z} : \pi_i(\mathrm{x}) - \pi_j(\mathrm{x}) = 2k)\}$$

Question 12

Define the abstraction and concretization for the "bounding abstraction" of the trace semantics which cuts traces at a given depth n, so that the abstract semantics has all its traces of length at most n. This abstraction is left implicit in bounded model-checking.

Answer to question 12

The bounding abstraction α^{b_n}, $n > 0$ cut traces at length n while preserving those of shorter length.

$$\begin{array}{rcl} \alpha^{b_n} & \in & \wp(\mathcal{S}^+) \mapsto \wp(\mathcal{S}^{\leqslant n}), \qquad \mathcal{S}^{\leqslant n} \triangleq \bigcup_{k\leqslant n} \mathcal{S}^k \\ \alpha^{b_n}(T) & \triangleq & \{b_n(\pi) \mid \pi \in T\} \\ b_n(\pi) & \triangleq & \pi \quad \text{when} \quad \pi \in \mathcal{S}^k \wedge k \leqslant n \\ b_n(\pi) & \triangleq & \pi_0 \ldots \pi_{n-1} \quad \text{when} \quad \pi \in \mathcal{S}^k \wedge k > n \end{array}$$

The concretization extend traces beyond length n by any possible behavior (so that nothing is known on traces after n steps).

$$\begin{array}{rcl} \gamma^{b_n} & \in & \wp(\mathcal{S}^{\leqslant n}) \mapsto \wp(\mathcal{S}^+) \\ \gamma^{b_n}(B) & \triangleq & \{\pi \in \mathcal{S}^+ \mid b_n(\pi) \in B\} \end{array}$$

so that $\langle \wp(\mathcal{S}^+),\ \subseteq\rangle \overset{\gamma^{b_n}}{\underset{\alpha^{b_n}}{\leftrightarrows}} \langle \wp(\mathcal{S}^{\leqslant n}),\ \subseteq\rangle$

$$\begin{array}{ll} \quad \alpha^{b_n}(T) \subseteq B & \\ \Leftrightarrow \{b_n(\pi) \mid \pi \in T\} \subseteq B & \wr\text{def. } \alpha^g\wr \\ \Leftrightarrow \forall \pi \in T : b_n(\pi) \in B & \wr\text{def. } \subseteq\wr \\ \Leftrightarrow T \subseteq \{\pi \in \mathcal{S}^+ \mid b_n(\pi) \in B\} & \wr\text{def. } \subseteq\wr \\ \Leftrightarrow T \subseteq \gamma^{b_n}(B) & \wr\text{def. } \gamma^{b_n}\wr \end{array}$$

Question 13

Consider the following *transition abstraction* from sets of traces to a transition relation.

$$\begin{array}{rcl} \alpha^\tau & \in & \wp(\mathcal{S}^+) \mapsto \wp(\mathcal{S} \times \mathcal{S}) \\ \alpha^\tau(T) & \triangleq & \{\langle \pi_i,\ \pi_{i+1}\rangle \mid \exists n \geqslant 1 : \pi \in T \cap \mathcal{S}^n \wedge 0 \leqslant i < n-1\} \end{array}$$

1. Provide an example proving that this abstraction can loose information on the set of traces.
2. Provide a characterization of those sets of traces for which the abstraction loose no information.

Answer to question 13

The concretization is

$$\begin{array}{rcl} \gamma^\tau & \in & \wp(\mathcal{S} \times \mathcal{S}) \mapsto \wp(\mathcal{S}^+) \\ \gamma^\tau(T) & \triangleq & \{\pi \in \mathcal{S}^n \mid n \geqslant 1 \wedge \forall i \in [0, n-2] : \langle \pi_i,\ \pi_{i+1}\rangle \in T\}. \end{array}$$

1. Given $\mathcal{S} = \{a, b\}$ and $S = \{ab, ba\}$, we have $\alpha^\tau(S) = \{\langle a,\ b\rangle, \langle b,\ a\rangle\}$ to that $\gamma^\tau(\{\langle a,\ b\rangle, \langle b,\ a\rangle\})$ is $(a(ba)*(\epsilon|b))|(b(ab)*(\epsilon|a))$ using a regular expression notation.
2. The sets of traces for which the abstraction loose no information are those that are prefix-close and history-insensitive.

Question 14

Prove that given $h \in \mathfrak{X} \in\mapsto \wp(\mathfrak{Y})$, defining $\alpha^h(P) \triangleq \bigcup\{h(x) \mid x \in P\}$ yields a Galois connection $\langle \wp(\mathfrak{X}),\ \subseteq\rangle \overset{\gamma^h}{\underset{\alpha^h}{\leftrightarrows}} \langle \wp(\mathfrak{Y}),\ \subseteq\rangle$ (this is the abstraction commonly used in model-checking).

Prove that any Galois connection $\langle \wp(\mathfrak{X}),\ \subseteq\rangle \overset{\gamma}{\underset{\alpha}{\leftrightarrows}} \langle \wp(\mathfrak{Y}),\ \subseteq\rangle$ can be put in that form for an appropriate choice of h such that $\alpha = \alpha^h$ and $\gamma = \gamma^h$.

Provide an example of abstraction $\langle \wp(\mathfrak{X}),\ \subseteq\rangle \overset{\gamma}{\underset{\alpha}{\leftrightarrows}} \langle A,\ \sqsubseteq\rangle$ that cannot be put in that form.

Answer to question 14

For all $P \in \wp(\mathfrak{X})$ and $Q \in \wp(\mathfrak{Y})$,

$$\begin{aligned}
& \alpha^h(P) \subseteq Q \\
\Leftrightarrow\ & \bigcup\{h(x) \mid x \in P\} \subseteq Q && \lbag\text{def. } \alpha^h(P) \triangleq \bigcup\{h(x) \mid x \in P\}\rbag \\
\Leftrightarrow\ & \forall x \in P : h(x) \subseteq Q && \lbag\text{def. } \bigcup\rbag \\
\Leftrightarrow\ & P \subseteq \{x \in \mathfrak{X} \mid h(x) \subseteq Q\} && \lbag\text{def. } \subseteq\rbag \\
\Leftrightarrow\ & P \subseteq \gamma^h(Q)
\end{aligned}$$

by defining $\gamma^h(Q) \triangleq \{x \in \mathfrak{X} \mid h(x) \subseteq Q\}$. Given $\langle\wp(\mathfrak{X}), \subseteq\rangle \mathrel{\underset{\alpha}{\overset{\gamma}{\leftrightarrows}}} \langle\wp(\mathfrak{Y}), \subseteq\rangle$, define $h \in \mathfrak{X} \in\mapsto \wp(\mathfrak{Y})$ by $h(x) = \alpha(\{x\})$. We have $\langle\wp(\mathfrak{X}), \subseteq\rangle \mathrel{\underset{\alpha^h}{\overset{\gamma^h}{\leftrightarrows}}} \langle\wp(\mathfrak{Y}), \subseteq\rangle$ as shown above. Moreover for $P \in \wp(\mathfrak{X})$,

$$\begin{aligned}
& \alpha^h(P) \\
=\ & \bigcup\{h(x) \mid x \in P\} && \lbag\text{def. } \alpha^h(P) \triangleq \bigcup\{h(x) \mid x \in P\}\rbag \\
=\ & \bigcup\{\alpha(\{x\}) \mid x \in P\} && \lbag\text{def. } h(x) = \alpha(\{x\})\rbag \\
=\ & \alpha(\bigcup\{\{x\} \mid x \in P\}) && \lbag\alpha \text{ preserves lubs}\rbag \\
=\ & \alpha(\{x \mid x \in P\}) && \lbag\text{def. } \bigcup\rbag \\
=\ & \alpha(P) && \lbag\text{def. } \in\rbag
\end{aligned}$$

and for $Q \in \wp(\mathfrak{Y})$,

$$\begin{aligned}
& \gamma^h(Q) \\
=\ & \{x \in \mathfrak{X} \mid h(x) \subseteq Q\} && \lbag\text{def. } \gamma^h(Q) \triangleq \{x \in \mathfrak{X} \mid h(x) \subseteq Q\}\rbag \\
=\ & \{x \in \mathfrak{X} \mid \alpha(\{x\}) \subseteq Q\} && \lbag\text{def. } h(x) = \alpha(\{x\})\rbag \\
=\ & \{x \in \mathfrak{X} \mid \{x\} \subseteq \gamma(Q)\} && \lbag\text{Galois connection inversion}\rbag \\
=\ & \{x \in \mathfrak{X} \mid x \in \gamma(Q)\} && \lbag\text{def. } \subseteq\rbag \\
=\ & \gamma(Q) && \lbag\text{def. } \in \text{ and } \gamma(Q) \in \wp(\mathfrak{X})\rbag
\end{aligned}$$

The interval abstraction is obviously not of that form.

Question 15

Show that the program property “to be deterministic” (i.e. to have only one possible execution trace, either finite or infinite) is neither a safety nor a liveness property.

Answer to question 15

The property of a program $\mathbf{P}$ with states $\mathcal{S}[\![\mathbf{P}]\!]$ “to be deterministic” is $\{\{\pi\} \mid \pi \in \mathcal{S}[\![\mathbf{P}]\!]^{+\infty}\}$. This is not a trace property hence neither a safety nor a liveness property.

Question 16

Is the infinite union of safety properties a safety property?

Answer to question 16

The infinite union of safety properties is not a safety property. For example, terminating in exactly n steps is a safety property. To check it at runtime just count the number of steps and produce an alarm after n steps. Their infinite union is termination, which is not a safety property.

Question 17

The abstract best transformer for the interval abstraction has been shown to be

$$[a_1, b_1] \overline{-} [a_2, b_2] \quad = \quad [a_1 - b_2, b_1 - a_2] \tag{1}$$

Assume that variable `x` has a value $x \in [-100, 100]$. For the value of the expression `x - x`, calculate $x \overline{-} x$ as indicated in (1). Can you do better? Why does $\overline{-}$ is so-called the best abstraction of $-$ on powersets?

Answer to question 17

We have $x \overline{-} x = [-200, 200]$ whereas once could imagine $[0, 0]$ which looks better. Nevertheless $\overline{-}$ is the best abstraction after the Cartesian abstraction, which ignores that the two parameters correspond to the same program variable and so returns the same result for $x \overline{-} x$ and $x \overline{-} y$, $x = [-100, 100]$, $y = [-100, 100]$.

Question 18

The partial trace semantics of a transition system $\langle \mathcal{S},\ \mathcal{I},\ \mathcal{F},\ \mathsf{T}\rangle$ is $\mathbf{P}^t \triangleq \{\pi \in \mathcal{S}^n \mid n \geqslant 1 \wedge \forall i \in [0, n-2] : \langle \pi_i,\ \pi_{i+1}\rangle \in \mathsf{T}\} = \mathbf{lfp}^{\subseteq}\, \mathsf{F}^t$ where $\mathsf{F}^t(X) \triangleq \mathcal{S}^1 \cup \{\pi\sigma\sigma' \mid \pi\sigma \in X \wedge \langle \sigma,\ \sigma'\rangle \in \mathsf{T}\}$. Ihe prefix trace semantics is defined as the restriction of the prefix trace semantics to traces starting with an initial state. $\mathbf{P}^{ti} \triangleq \{\pi \in \mathcal{S}^n \mid n \geqslant 1 \wedge \pi_0 \in \mathcal{I} \wedge \forall i \in [0, n-2] : \langle \pi_i,\ \pi_{i+1}\rangle \in \mathsf{T}\}$. This prefix trace semantics has the following fixpoint characterization $\mathbf{P}^{ti} = \mathbf{lfp}^{\subseteq}\, \mathsf{F}^{ti}$ where $\mathsf{F}^{ti}(X) \triangleq \{\pi \in \mathcal{S}^1 \mid \pi_0 \in \mathcal{I}\} \cup \{\pi\sigma\sigma' \mid \pi\sigma \in X \wedge \langle \sigma,\ \sigma'\rangle \in \mathsf{T}\}$.

Find an abstraction α such that $\mathbf{P}^{ti} = \alpha(\mathbf{P}^t)$. Then derive the fixpoint characterization of $\mathbf{P}^{ti}$ from that of $\mathbf{P}^t$ using Th. 1.

Answer to question 18

The abstraction is $\langle \wp(\mathcal{S}^+),\ \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(\mathcal{S}^+),\ \subseteq\rangle$ where $\alpha(X) = \{\pi \in X \mid \pi_0 \in \mathcal{I}\}$. F^{ti} is derived from F^t using the commutation condition of Th. 1.

$$\begin{aligned}
& \alpha \circ \mathsf{F}^t(X) \\
= \ & \{\pi \in (\mathcal{S}^1 \cup \{\pi\sigma\sigma' \mid \pi\sigma \in X \wedge \langle \sigma,\ \sigma'\rangle \in \mathsf{T}\}) \mid \pi_0 \in \mathcal{I}\} && \wr\text{def. function composition } \circ,\ \alpha \text{ and } \mathsf{F}^t\int \\
= \ & \{\pi \in \mathcal{S}^1 \mid \pi_0 \in \mathcal{I}\} \cup \{\pi\sigma\sigma' \mid \pi\sigma \in X \wedge \pi_0 \in \mathcal{I} \wedge \langle \sigma,\ \sigma'\rangle \in \mathsf{T}\} && \wr\text{def. } \in\int \\
= \ & \{\pi \in \mathcal{S}^1 \mid \pi_0 \in \mathcal{I}\} \cup \{\pi\sigma\sigma' \mid \pi\sigma \in \{\pi\sigma \in X \mid \pi_0 \in \mathcal{I}\} \wedge \langle \sigma,\ \sigma'\rangle \in \mathsf{T}\} && \wr\text{def. } \in\int \\
= \ & \{\pi \in \mathcal{S}^1 \mid \pi_0 \in \mathcal{I}\} \cup \{\pi\sigma\sigma' \mid \pi\sigma \in \alpha(X) \wedge \langle \sigma,\ \sigma'\rangle \in \mathsf{T}\} && \wr\text{def. } \alpha\int \\
= \ & \mathbf{P}^{ti}(\alpha(X)) && \wr\text{def. } \mathbf{P}^{ti}\int
\end{aligned}$$

So by Th. 1, $\mathbf{P}^{ti} = \alpha(\mathbf{P}^t) = \alpha(\mathbf{lfp}^{\subseteq}_{\emptyset}\, \mathsf{F}^t) = \mathbf{lfp}^{\subseteq}_{\alpha(\emptyset)}\, \mathsf{F}^{ti} = \mathbf{lfp}^{\subseteq}_{\emptyset}\, \mathsf{F}^{ti}$.

Question 19

Consider the polyhedral abstraction where the abstract properties P are the conjunction of linear inequalities $\bigwedge_{i=1}^{m}\sum_{j=1}^{n} a_i^j \leqslant b_i$ written

$$
\begin{array}{rcl}
P & = & Ax \leqslant b \\
 & = & \{A_i x + b_i \mid i \in [1, m]\} \\
 & = & \{a_i^1 x_1 + \ldots + a_i^j x_j + \ldots + a_i^n x_n \leqslant b_i \mid i \in [1, m]\} \\
 & = & \{\displaystyle\sum_{j=1}^{n} a_i^j \leqslant b_i \mid i \in [1, m]\}
\end{array}
$$

We say that polyhedron P *entails* a constraint $\varphi = \sum_i a_i x_i \leqslant b_i$ when $\bigwedge P \Rightarrow \varphi$ that is $\{x \in \mathbb{Q}^n \mid Ax \leqslant b\} \subseteq \{x \in \mathbb{Q}^n \mid \sum_i a_i x_i \leqslant b_i\}$, written $P \models \varphi$.

The concretization is

$$\gamma^P(P) \quad \triangleq \quad \gamma^P(\langle A,\ b\rangle) \quad \triangleq \quad \{x \in \mathbb{Q}^n \mid Ax \leqslant b\}$$

that is the set of all possible values of the program numerical variables that satisfy all constraints in P.

Prove that $\forall \varphi_1 \in P_1 : \gamma^P(P_1) = \gamma^P((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\})$ if and only if $P_1 \models \varphi_2$ and $((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}) \models \varphi_1$.

Answer to question 19

If $\varphi = \sum_i a_i x_i \leqslant b_i$, define

$$\gamma^P(\varphi) \quad \triangleq \quad \{x \in \mathbb{Q}^n \mid \sum_i a_i x_i \leqslant b_i\}$$

If follows that if P is given by $Ax \leqslant b$ then

$$
\begin{array}{lll}
 & \gamma^P(P) & \\
= & \{x \in \mathbb{Q}^n \mid Ax \leqslant b\} & \wr\text{def. } \gamma^P(P)\int \\
= & \{x \in \mathbb{Q}^n \mid \displaystyle\bigwedge_{i=1}^{n} A_i x \leqslant b_i\} & \wr\text{conjunctive interpretation of } Ax \leqslant b\int \\
= & \bigcap\{x \in \mathbb{Q}^n \mid i \in [1, n] \wedge A_i x =\leqslant b_i\} & \wr\text{def. } \cap\int \\
 & \bigcap\{\gamma^P(A_i x \leqslant b_i) \mid i \in [1, n] \wedge A_i x \leqslant b_i\} & \wr\text{def. } \gamma^P \text{ for a single constraint}\int \\
= & \bigcap\{\gamma^P(\varphi) \mid \varphi \in P\} & \wr\text{def. } P = \{A_i x \leqslant b_i \mid i \in [1, n]\}\int
\end{array}
$$

Assume $\varphi_1 \in P_1$.

$$
\begin{array}{lll}
— & \gamma^P(P_1) = \gamma^P((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}) & \\
\Leftrightarrow & \gamma^P(P_1) = \displaystyle\bigcap_{\varphi \in (P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}} \gamma^P(\varphi) & \wr\text{as shown above for } \gamma^P\int \\
\Leftrightarrow & \gamma^P(P_1) = \displaystyle\bigcap_{\varphi \in P_1 \setminus \{\varphi_1\}} \gamma^P(\varphi) \cap \gamma^P(\varphi_2) & \wr\text{def. } \cap\int \\
\Rightarrow & \gamma^P(P_1) \subseteq \gamma^P(\varphi_2) & \wr\text{by reflexivity and def. glb}\int \\
\Leftrightarrow & P_1 \models \varphi_2 & \wr\text{def. } \models \text{ and } \gamma^P \text{ above}\int
\end{array}
$$

— Moreover,

$$\begin{aligned}
& \gamma^P(P_1) = \gamma^P((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}) \\
\Leftrightarrow\ & \bigcap_{\varphi \in P_1} \gamma^P(\varphi) = \gamma^P((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}) && \wr\text{as shown above for } \gamma^P \int \\
\Leftrightarrow\ & \bigcap_{\varphi \in P_1} \gamma^P(\varphi) \cap \gamma^P(\varphi_1) = \gamma^P((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}) && \wr \varphi_1 \in P_1 \text{ by hypothesis} \int \\
\Rightarrow\ & \gamma^P((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}) \subseteq \gamma^P(\varphi_1) && \wr\text{by reflexivity and def. glb} \int \\
\Leftrightarrow\ & (P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\} && \wr\text{def. } \models \text{ and } \gamma^P \text{ above} \int
\end{aligned}$$

— Reciprocally, assume $(P_1 \models \varphi_2) \wedge ((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}) \models \varphi_1$ or equivalently, by definition of $\models$ and γ^P above, that $(\gamma^P(P_1) \subseteq \gamma^P(\varphi_2)) \wedge (\gamma^P((P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}) \subseteq \gamma^P(\varphi_1))$. Then,

$$\begin{aligned}
& \gamma^P(P_1) = \bigcap_{\varphi \in P_1} \gamma^P(\varphi) && \wr\text{as shown above for } \gamma^P \int \\
= \ & \bigcap_{\varphi \in P_1 \setminus \{\varphi_1\}} \gamma^P(\varphi) \cap \gamma^P(\varphi_1) && \wr\text{since } \varphi_1 \in P_1 \int \\
= \ & \bigcap_{\varphi \in P_1 \setminus \{\varphi_1\}} \gamma^P(\varphi) \cap \gamma^P(\varphi_1) \cap \gamma^P(\varphi_2) \\
& && \wr\text{since } \textstyle\bigcap_{\varphi \in P_1 \setminus \{\varphi_1\}} \gamma^P(\varphi) \cap \gamma^P(\varphi_1) = \gamma^P(P_1) \subseteq \gamma^P(\varphi_2) \int \\
\Leftrightarrow\ & \bigcap_{\varphi \in (P_1 \setminus \{\varphi_1\}) \cup \{\varphi_2\}} \gamma^P(\varphi) \cap \gamma^P(\varphi_1) && \wr\text{def. } \textstyle\bigcap \int \quad \blacksquare
\end{aligned}$$

Question 20

Prove the following statements to be wrong: "*the widening approach to program static analysis is useless since it is always possible to perform an iterative static analysis using a finite abstract domain*"Ãě[7] and "*widenings can always be designed by further abstraction in an abstract domain satisfying the ascending chain condition*"[8].

Answer to question 20

This is due to the confusion between the static analysis of a given specific program $\boldsymbol{P}$ and the static analysis of all programs $\boldsymbol{P} \in \mathbb{W}$ of a language with infinitely many different programs as shown by the interval analysis of the following program

$$\boldsymbol{P}(n) \quad \triangleq \quad {}^{1}\texttt{x := 1 ; while } {}^{2}(\texttt{x <= } n) \texttt{ do } {}^{3}\texttt{x := (x + 1); od}^{4}.$$

for all possible values of n.

[7] R.B. Kieburtz and M. Napierala. Abstract semantics. In S. Abramsky and C. Hankin, eds., *Abstract Interpretation of Declarative Languages*, chapter 7, pp. 143–180. Ellis Horwood, Chichester, U.K., 1987.

[8] C. Hankin, S. Hunt: Approximate Fixed Points in Abstract Interpretation. In *Sci. Comput. Program.* 22(3):283–306 (1994)