Properties

MPRI 2–6: Abstract Interpretation,
application to verification and static analysis

Antoine Miné

year 2013–2014

course 03-A
4 October 2013

State properties

State properties

State property: $P \in \mathcal{P}(\Sigma)$.

Verification problem: $\mathcal{R}(\mathcal{I}) \subseteq P$.

(all the states reachable from $\mathcal{I}$ are in P)

Examples:

- absence of blocking: $P \stackrel{\text{def}}{=} \Sigma \setminus \mathcal{B}$,
- the variables remain in a safe range,
- dangerous program locations cannot be reached.

Invariance proof method

Invariance proof method: find an inductive invariant $I \subseteq \Sigma$

- $\mathcal{I} \subseteq I$
  (contains initial states)
- $\forall \sigma \in I: \sigma \to_\tau \sigma' \implies \sigma' \in I$
  (invariant by program transition)

that implies the desired property: $I \subseteq P$.

Link with the state semantics $\mathcal{R}(\mathcal{I})$:

Given $F_{\mathcal{R}}(S) \stackrel{\text{def}}{=} \mathcal{I} \cup \text{post}_\tau(S)$, we have $F_{\mathcal{R}}(I) \subseteq I$
$\implies$ I is a post-fixpoint of $F_{\mathcal{R}}$.

Recall that $\mathcal{R}(\mathcal{I}) = \text{lfp}\, F_{\mathcal{R}}$
$\implies$ $\mathcal{R}(\mathcal{I})$ is the tightest inductive invariant.

Hoare logic proof method

Idea:

- annotate program points with local sate invariants in $\mathcal{P}(\Sigma)$
- use logic rules to prove their correctness

$$\frac{}{\{P[e/X]\}\ X \leftarrow e\ \{P\}} \qquad \frac{\{P\}\ stat_1\ \{R\} \quad \{R\}\ stat_2\ \{Q\}}{\{P\}\ stat_1;\ stat_2\ \{Q\}}$$

$$\frac{\{P \wedge b\}\ stat\ \{Q\} \quad P \wedge \neg b \Rightarrow Q}{\{P\}\ \textbf{if}\ b\ \textbf{then}\ stat\ \{Q\}} \qquad \frac{\{P \wedge b\}\ stat\ \{P\}}{\{P\}\ \textbf{while}\ b\ \textbf{do}\ stat\ \{P \wedge \neg b\}}$$

$$\frac{\{P\}\ stat\ \{Q\} \quad P' \Rightarrow P \quad Q \Rightarrow Q'}{\{P'\}\ stat\ \{Q'\}}$$

Link with the state semantics $\mathcal{R}(\mathcal{I})$:

Equivalent to an invariant proof, partitioned by program location.
Any post-fixpoint of $\alpha_{\mathcal{L}} \circ F_{\mathcal{R}} \circ \gamma_{\mathcal{L}}$ gives valid Hoare triples.
$\alpha_{\mathcal{L}}(\mathcal{R}(\mathcal{I})) = \mathrm{lfp}(\alpha_{\mathcal{L}} \circ F_{\mathcal{R}} \circ \gamma_{\mathcal{L}})$ gives the tightest Hoare triples.

Weakest liberal precondition proof methods

Idea: Start with a postcondition $\mathcal{F} \in \mathcal{P}(\Sigma)$
and compute preconditions backwards $P \Rightarrow wlp(stat, Q)$

- $wlp(X \leftarrow e, Q) \stackrel{\text{def}}{=} Q[e/X]$
- $wlp((stat_1; stat_2), Q) \stackrel{\text{def}}{=} wlp(stat_1, wlp(stat_2, Q))$
- $wlp(\textbf{if } b \textbf{ then } stat, Q) \stackrel{\text{def}}{=} (b \Rightarrow wlp(stat, Q)) \wedge (\neg b \Rightarrow Q)$
- $wlp(\textbf{while } b \textbf{ do } stat, Q) \stackrel{\text{def}}{=}$
  $I \wedge ((I \wedge b) \Rightarrow wlp(stat, I)) \wedge ((I \wedge \neg b) \Rightarrow Q)$
  (where the loop invariant I is generally provided by the user)

($P \Rightarrow wlp(stat, Q)$ is equivalent to $\{P\}$ $stat$ $\{Q\}$)

Link with the state semantics $\mathcal{S}(\mathcal{Y})$:

(recall $\mathcal{S}(\mathcal{Y}) = \text{gfp } F_{\mathcal{S}}$ where $F_{\mathcal{S}}(S) \stackrel{\text{def}}{=} \mathcal{Y} \cap \widetilde{\text{pre}}_\tau(S)$)

Equivalent to sufficient preconditions, partitioned by location:
any pre-fixpoint of $\alpha_{\mathcal{L}} \circ F_{\mathcal{S}} \circ \gamma_{\mathcal{L}}$ gives valid liberal preconditions;
$\alpha_{\mathcal{L}}(\mathcal{S}(\mathcal{F})) = \text{gfp}(\alpha_{\mathcal{L}} \circ F_{\mathcal{R}} \circ \gamma_{\mathcal{L}})$ gives the weakest liberal preconditions while inferring loop invariants!

Trace properties

Trace properties

Trace property: $P \in \mathcal{P}(\Sigma^\infty)$

Verification problem: $\mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty) \subseteq P$

Examples:

- termination: $P \stackrel{\text{def}}{=} \Sigma^*$,
- non-termination: $P \stackrel{\text{def}}{=} \Sigma^\omega$,
- any state property $S \subseteq \Sigma$: $P \stackrel{\text{def}}{=} S^\infty$,
- maximal execution time: $P \stackrel{\text{def}}{=} \Sigma^{\leq k}$,
- minimal execution time: $P \stackrel{\text{def}}{=} \Sigma^{\geq k}$,
- ordering, e.g.: $P \stackrel{\text{def}}{=} (\Sigma \setminus \{b\})^* \cdot a \cdot \Sigma^* \cdot b \cdot \Sigma^\infty$.
  (a and b occur, and a occurs before b)

Safety properties

Idea: a safety property P models that "nothing bad ever occurs"

- P is provable by exhaustive testing;
  (observe the prefix trace semantics: $\mathcal{T}_p(\mathcal{I}) \subseteq P$)
- P is disprovable by finding a single finite execution not in P.

Examples:

- any state property: $P \stackrel{\text{def}}{=} S^\infty$ for $S \subseteq \Sigma$,
- ordering: $P \stackrel{\text{def}}{=} \Sigma^\infty \setminus ((\Sigma \setminus \{a\})^* \cdot b \cdot \Sigma^\infty)$,
  (no b can appear without an a before,
  but we can have only a, or neither a nor b)
  (not a state property)
- but termination $P \stackrel{\text{def}}{=} \Sigma^*$ is not a safety property.
  (disproving requires exhibiting an *infinite* execution)

Definition of safety properties

Reminder: finite prefix abstraction (simplified to allow ϵ)

$$(\mathcal{P}(\Sigma^\infty), \subseteq) \xleftarrow[\alpha_{*\preceq}]{\gamma_{*\preceq}} (\mathcal{P}(\Sigma^*), \subseteq)$$

- $\alpha_{*\preceq}(T) \stackrel{\text{def}}{=} \{\, t \in \Sigma^* \mid \exists u \in T \colon t \preceq u \,\}$
- $\gamma_{*\preceq}(T) \stackrel{\text{def}}{=} \{\, t \in \Sigma^\infty \mid \forall u \in \Sigma^* \colon u \preceq t \implies u \in T \,\}$

The associated upper closure $\rho_{*\preceq} \stackrel{\text{def}}{=} \gamma_{\preceq} \circ \alpha_{\preceq}$ is:
$\rho_{*\preceq} = \lim \circ \rho_p$ where:

- $\rho_p(T) \stackrel{\text{def}}{=} \{\, u \in \Sigma^\infty \mid \exists t \in T \colon u \preceq t \,\}$,
- $\lim(T) \stackrel{\text{def}}{=} T \cup \{\, t \in \Sigma^\omega \mid \forall u \in \Sigma^* \colon u \preceq t \implies u \in T \,\}$.

Definition: $P \in \mathcal{P}(\Sigma^\infty)$ is a safety property if $P = \rho_{*\preceq}(P)$.

Definition of safety properties (examples)

Definition: $P \subseteq \mathcal{P}(\Sigma^\infty)$ is a safety property if $P = \rho_{*\preceq}(P)$.

Examples and counter-examples:

- state property $P \stackrel{\text{def}}{=} S^\infty$ for $S \subseteq \Sigma$:

  $\rho_p(S^\infty) = \lim(S^\infty) = S^\infty \implies$ safety;

- termination $P \stackrel{\text{def}}{=} \Sigma^*$:

  $\rho_p(\Sigma^*) = \Sigma^*$, but $\lim(\Sigma^*) = \Sigma^\infty \neq \Sigma^* \implies$ not safety;

- even number of steps $P \stackrel{\text{def}}{=} (\Sigma^2)^\infty$:

  $\rho_p((\Sigma^2)^\infty) = \Sigma^\infty \neq (\Sigma^2)^\infty \implies$ not safety.

Proving safety properties

Invariance proof method: find an inductive invariant I

- set of finite traces $I \subseteq \Sigma^*$
- $\mathcal{I} \subseteq I$
  (contains traces reduced to an initial state)
- $\forall \sigma_0, \ldots, \sigma_n \in I$: $\sigma_n \to_\tau \sigma_{n+1} \implies \sigma_0, \ldots, \sigma_n, \sigma_{n+1} \in I$
  (invariant by program transition)

and implies the desired property: $I \subseteq P$.

Link with the finite prefix trace semantics $\mathcal{T}_p(\mathcal{I})$:

An inductive invariant is a post-fixpoint of F_p: $F_p(I) \subseteq I$
where $F_p(T) \stackrel{\text{def}}{=} \mathcal{I} \cup T \frown \tau$.
$\mathcal{T}_p(\mathcal{I}) = \text{lfp } F_p$ is the tightest inductive invariant.

Correctness of the invariant method for safety

Soundness:

if P is a safety property and an inductive invariant I exists
then: $\mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty) \subseteq P$

proof:

Using the Galois connection between $\mathcal{M}_\infty$ and $\mathcal{T}$, we get:
$\mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty) \subseteq \rho_{*\preceq}(\mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty)) = \gamma_{*\preceq}(\alpha_{*\preceq}(\mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty))) =$
$\gamma_{*\preceq}(\alpha_{*\preceq}(\mathcal{M}_\infty) \cap (\mathcal{I} \cdot \Sigma^*)) = \gamma_{*\preceq}(\mathcal{T} \cap (\mathcal{I} \cdot \Sigma^*)) = \gamma_{*\preceq}(\mathcal{T}_p(\mathcal{I}))$.
Using the link between invariants and the finite prefix trace semantics, we have: $\mathcal{T}_p(\mathcal{I}) \subseteq I \subseteq P$.
As P is a safety property, $P = \gamma_{*\preceq}(P)$, so, $\gamma_{*\preceq}(\mathcal{T}_p(\mathcal{I})) \subseteq \gamma_{*\preceq}(P) = P$,
and so, $\mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty) \subseteq P$.

Completeness: an inductive invariant always exists

proof: $\mathcal{T}_p(\mathcal{I})$ provides an inductive invariant.

Disproving safety properties

Proof method:

A safety property P can be disproved by constructing a finite prefix of execution that does not satisfy the property:

$$\mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty) \not\subseteq P \implies \exists t \in \mathcal{T}_p(\mathcal{I}) \colon t \notin P$$

proof:

By contradiction, assume that no such trace exists, i.e., $\mathcal{T}_p(\mathcal{I}) \subseteq P$.
We proved in the previous slide that this implies $\mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty) \subseteq P$.

Examples:

- disproving a state property $P \stackrel{\text{def}}{=} S^\infty$:
  $\Rightarrow$ find a partial execution containing a state in $\Sigma \setminus S$;
- disproving an order property $P \stackrel{\text{def}}{=} \Sigma^\infty \setminus ((\Sigma \setminus \{a\})^* \cdot b \cdot \Sigma^\infty)$
  $\Rightarrow$ find a partial execution where b appears and not a.

Liveness properties

Idea: liveness property $P \in \mathcal{P}(\Sigma^\infty)$

Liveness properties model that "something good eventually occurs"

- P cannot be proved by testing
  (if nothing good happens in a prefix execution,
  it can still happen in the rest of the execution)
- disproving P requires exhibiting an infinite execution not in P

Examples:

- termination: $P \stackrel{\text{def}}{=} \Sigma^*$,
- inevitability: $P \stackrel{\text{def}}{=} \Sigma^* \cdot a \cdot \Sigma^\infty$,
  (a eventually occurs in all executions)
- state properties are not liveness properties.

Definition of liveness properties

Definition: $P \in \mathcal{P}(\Sigma^\infty)$ is a liveness property if $\rho_{*\preceq}(P) = \Sigma^\infty$.

Examples and counter-examples:

- termination $P \stackrel{\text{def}}{=} \Sigma^*$:

  $\rho_p(\Sigma^*) = \Sigma^*$ and $\lim(\Sigma^*) = \Sigma^\infty \implies$ liveness;
- inevitability: $P \stackrel{\text{def}}{=} \Sigma^* \cdot a \cdot \Sigma^\infty$

  $\rho_p(P) = P \cup \Sigma^*$ and $\lim(P \cup \Sigma^*) = \Sigma^\infty \implies$ liveness;
- state property $P \stackrel{\text{def}}{=} S^\infty$ for $S \subseteq \Sigma$:

  $\rho_p(S^\infty) = \lim(S^\infty) = S^\infty \neq \Sigma^\infty$ if $S \neq \Sigma \implies$ not liveness;
- maximal execution time $P \stackrel{\text{def}}{=} \Sigma^{\leq k}$:

  $\rho_p(\Sigma^{\leq k}) = \lim(\Sigma^{\leq k}) = \Sigma^{\leq k} \neq \Sigma^\infty \implies$ not liveness;
- the only property which is both safety and liveness is Σ^∞.

Proving liveness properties

Variance proof method: (informal definition)

Find a decreasing quantity until something good happens.

Example: termination proof

- find $f : \Sigma \to \mathcal{S}$ where $(\mathcal{S}, \sqsubseteq)$ is well-ordered;

  (f is called a "ranking function")
- $\sigma \in \mathcal{B} \implies f = \min \mathcal{S}$;
- $\sigma \to_\tau \sigma' \implies f(\sigma') \sqsubset f(\sigma)$.

(f counts the number of steps remaining before termination)

Disproving liveness properties

Property:

If P is a liveness property, then $\forall t \in \Sigma^*: \exists u \in P: t \preceq u$.

proof:

By definition of liveness, $\rho_{*\preceq}(P) = \Sigma^\infty$, so $t \in \rho_{*\preceq}(P) = \lim(\alpha_p(P))$.
As $t \in \Sigma^*$ and lim only adds infinite traces, $t \in \alpha_p(P)$.
By definition of α_p, $\exists u \in P: t \preceq u$.

Consequence:

- liveness cannot be disproved by testing.

Trace topology

Topology on X, defined by

- a family $\mathcal{C} \subseteq \mathcal{P}(X)$ of closed sets
  - $c, c' \in \mathcal{C} \implies c \cup c' \in \mathcal{C}$ (closed by finite unions)
  - $C \subseteq \mathcal{C} \implies \cap \{ c \mid c \in C \} \in \mathcal{C}$ (closed by intersections)
- open sets $\mathcal{O}$ are derived from closed sets:
  $\mathcal{O} \stackrel{\text{def}}{=} \{ X \setminus c \mid c \in \mathcal{C} \}$
  (closed by unions and finite intersections)
  (we can alternatively define a topology by $\mathcal{O}$, and derive $\mathcal{C}$ from $\mathcal{O}$)

Definition: we define a topology on traces by setting:

- $X \stackrel{\text{def}}{=} \Sigma^\infty$
- $\mathcal{C} \stackrel{\text{def}}{=} \{ P \in \mathcal{P}(\Sigma^\infty) \mid P \text{ is a safety property} \}$

Closure and density

Topological closure: $\rho : \mathcal{P}(X) \to \mathcal{P}(X)$

- $\rho(x) \stackrel{\text{def}}{=} \cap \{ c \in \mathcal{C} \mid x \subseteq c \}$;
  (ρ is an upper closure operator in $(\mathcal{P}(X), \subseteq)$)
  $(\rho(x) = x \iff x \in \mathcal{C})$
- on our trace topology, $\rho = \rho_{*\preceq}$.

Dense sets:

- $x \subseteq X$ is dense if $\rho(x) = X$;
- on our trace topology, dense sets are liveness properties.

Decomposition theorem

Theorem: decomposition on a topological space

Any set $x \subseteq X$ is the intersection of a closed set and a dense set.

proof:

We have $x = \rho(x) \cap (x \cup (X \setminus \rho(x)))$. Indeed:
$\rho(x) \cap (x \cup (X \setminus \rho(x))) = (\rho(x) \cap x) \cup (\rho(x) \cap (X \setminus \rho(x))) = \rho(x) \cap x = x$
as $x \subseteq \rho(x)$.

- $\rho(x)$ is closed
- $x \cup (X \setminus \rho(x))$ is dense because:

$$\begin{aligned} \rho(x \cup (X \setminus \rho(x))) &\supseteq \rho(x) \cup \rho(X \setminus \rho(x)) \\ &\supseteq \rho(x) \cup (X \setminus \rho(x)) \\ &= X \end{aligned}$$

Consequence: on trace properties

Every trace property is the conjunction of
a safety property and a liveness property.
(proving a trace property can be decomposed into
a soundness proof and a liveness proof)

Program properties

Properties

We generalize the notion of properties and program verification.

General setting:

- programs: $prog \in Prog$
- semantics: $[\![\cdot]\!] : Prog \to \mathcal{D}$ in some semantic domain $\mathcal{D}$
- property: the set of allowed program semantics $P \in \mathcal{P}(\mathcal{D})$

  $\subseteq$ gives an information order on properties

  $P \subseteq P'$ means that P' is weaker than P (allows more semantics)
- verification problem: $[\![prog]\!] \in P$

Collecting semantics

Collecting semantics: $Col : Prog \to \mathcal{P}(\mathcal{D})$

- $Col(prog) \stackrel{\text{def}}{=} \{ [\![prog]\!] \}$
- $Col(prog)$ is the strongest property of a program in $\mathcal{P}(\mathcal{D})$
  (relative to the choice of the semantic domain $\mathcal{D}$ and function $[\![\cdot]\!]$)
- we can interpret program verification as property inclusion:
  $Col(prog) \subseteq P$
  P is weaker than $Col(prog)$ in the information order of properties
- generally, the collecting semantics cannot be computed;
  we settle for a weaker property $S^\sharp$ that
  - is sound: $Col(prog) \subseteq S^\sharp$
  - implies the desired property: $S^\sharp \subseteq P$

Retrieving state and trace properties

Reachability state semantics:

- $\mathcal{D} \stackrel{\text{def}}{=} \mathcal{P}(\Sigma)$
- $[\![\cdot]\!] \stackrel{\text{def}}{=} \mathcal{R}(\mathcal{I})$

Trace semantics:

- $\mathcal{D} \stackrel{\text{def}}{=} \mathcal{P}(\Sigma^\infty)$
- $[\![\cdot]\!] \stackrel{\text{def}}{=} \mathcal{M}_\infty \cap (\mathcal{I} \cdot \Sigma^\infty)$

State and trace properties: interpreted in $\mathcal{P}(\mathcal{D})$

$\rho_\downarrow(x)$ for some $x \in \mathcal{D}$
where $\rho_\downarrow(x) \stackrel{\text{def}}{=} \{\, y \in \mathcal{D} \,|\, y \subseteq x \,\} \in \mathcal{P}(\mathcal{D})$

(proof: $A \subseteq B \iff A \in \rho_\downarrow(B)$)

Non-trace properties

Note: expressing properties in $\mathcal{P}(\mathcal{D})$
is more general than expressing properties in $\mathcal{D}$

Example: non-interference for variable X

$$P \stackrel{\text{def}}{=} \{\, T \in \mathcal{P}(\Sigma^*) \mid \forall \sigma_0, \ldots, \sigma_n \in T : \forall \sigma'_0 : \sigma_0 \equiv \sigma'_0 \implies \exists \sigma'_0, \ldots, \sigma'_m \in T : \sigma'_m \equiv \sigma_m \,\}$$

where $(\ell, \rho) \equiv (\ell', \rho') \iff \ell = \ell' \wedge \forall V \neq X : \rho(V) = \rho'(V)$

(changing the initial value of X does not affect the set of final environments up to the value of X)

There is no $Q \subseteq \Sigma^\infty$ such that $P = \rho_\downarrow(Q)$.
$\implies$ non-interference is not a trace property in $\mathcal{P}(\Sigma^\infty)$.