APPLICATION TO PREDICATE ABSTRACTION

AND LOCAL COMPLETION (REFINEMENT)

P. Cousot

Patrick.Cousot@ens.fr http://www.di.ens.fr/~cousot

Biarritz IFIP-WG 2.3 meeting (2)

23 — 28 mars 2003, Hotel Miramar, Biarritz, France

3.2 Application to Predicate Abstraction

Indead an abstract interpretation of:

Reference

[2] S. Graf and H. Saïdi. Construction of abstract state graphs with PVS. In *Proc. 9th Int. Conf. CAV '97*,LNCS 1254, pp. 72–83. Springer, 1997.

Verification that Reachable States are Safe

- States: Σ
- Initial states: $I \subseteq \Sigma$
- Safe states: $S \subseteq \Sigma$
- Transition relation $t \subseteq \Sigma \times \Sigma$ (Small step operational semantics)
- Verification problem:

$$
\begin{aligned}
& \operatorname{post}[t^\star]I \subseteq S \\
\Leftrightarrow\ & \left(\operatorname{lfp}_{\emptyset}^{\subseteq} \lambda X \cdot I \cup \operatorname{post}[t]X\right) \subseteq S
\end{aligned}
$$

The Structure of Program States

- Program points/labels: $\mathcal{L}$ is finite
- Variables: $\mathbb{X}$ is finite (for a given program)
- Set of values: $\mathcal{V}$
- Memory states: $\mathcal{M} = \mathbb{X} \longmapsto \mathcal{V}$

Local Versus Global Assertions

- Isomorphism between global and local assertions:

$$\langle \wp(\mathcal{L} \times \mathcal{M}),\ \subseteq \rangle \underset{\alpha_\downarrow}{\overset{\gamma_\downarrow}{\leftleftarrows\!\!\!\rightrightarrows}} \langle \mathcal{L} \longmapsto \wp(\mathcal{M}),\ \dot{\subseteq} \rangle$$

where:

$$\begin{aligned} \alpha_\downarrow(P) &= \lambda \ell \cdot \{m \mid \langle \ell,\ m \rangle \in P\} \\ \gamma_\downarrow(Q) &= \{\langle \ell,\ m \rangle \mid \ell \in \mathcal{L} \wedge m \in Q_\ell\} \end{aligned}$$

and $\dot{\subseteq}$ is the pointwise ordering:
$Q \mathrel{\dot{\subseteq}} Q'$ if and only if $\forall \ell \in \mathcal{L} : Q_\ell \subseteq Q'_\ell$.

Syntactic Predicates

- Choose a set $\mathbb{P}$ of syntactic predicates such that:

$$\forall S \subseteq \mathbb{P} : \left(\bigwedge S \right) \in \mathbb{P}$$

- an interpretation $\mathcal{I} \in \mathbb{P} \longmapsto \wp(\mathcal{M})$ such that:

$$\forall S \subseteq \mathbb{P} : \mathcal{I}\left(\bigwedge S \right) = \bigcap_{p \in S} \mathcal{I}[\![p]\!]$$

- It follows that $\{\mathcal{I}[\![p]\!] \mid p \in \mathbb{P}\}$ is a Moore family.

Predicate Abstraction

A memory state property $Q \in \wp(\mathcal{M})$ is approximated by the subset of predicates p of $\mathbb{P}$ which holds when Q holds (formally $Q \subseteq \mathcal{I}[\![p]\!]$). This defines a Galois connection:

$$\langle \wp(\mathcal{M}),\ \subseteq \rangle \underset{\alpha_{\mathbb{P}}}{\overset{\gamma_{\mathbb{P}}}{\leftrightarrows}} \langle \wp(\mathbb{P}),\ \supseteq \rangle$$

where:

$$\alpha_{\mathbb{P}}(Q) \stackrel{\text{def}}{=} \{p \in \mathbb{P} \mid Q \subseteq \mathcal{I}[\![p]\!]\}$$

$$\gamma_{\mathbb{P}}(P) \stackrel{\text{def}}{=} \bigcap \{\mathcal{I}[\![p]\!] \mid p \in P\}$$

Pointwise Extension to All Program Points

By pointwise extension, we have for all program points:

$$\langle \mathcal{L} \longmapsto \wp(\mathcal{M}),\ \dot{\subseteq} \rangle \ \underset{\dot{\alpha}_{\mathbb{P}}}{\overset{\dot{\gamma}_{\mathbb{P}}}{\leftrightarrows}}\ \langle \mathcal{L} \longmapsto \wp(\mathbb{P}),\ \dot{\supseteq} \rangle$$

where:

$$\dot{\alpha}_{\mathbb{P}}(Q) = \lambda \ell \cdot \alpha_{\mathbb{P}}(Q_\ell)$$

$$\dot{\gamma}_{\mathbb{P}}(P) = \lambda \ell \cdot \gamma_{\mathbb{P}}(P_\ell)$$

$$P \mathrel{\dot{\supseteq}} P' = \forall \ell \in \mathcal{L} : P_\ell \supseteq P'_\ell$$

Boolean Encoding

- $\mathbb{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_k\}$ is finite
- $\mathbb{B} = \{\mathtt{tt}, \mathtt{ff}\}$ is the set of booleans with $\mathtt{ff} \Rightarrow \mathtt{ff} \Rightarrow \mathtt{tt} \Rightarrow \mathtt{tt}$
- We can use a boolean encoding of subsets of $\mathbb{P}$:

$$\langle \wp(\mathbb{P}),\ \supseteq\rangle \underset{\alpha_b}{\overset{\gamma_b}{\leftrightarrows}} \langle \prod_{i=1}^{k} \mathbb{B},\ \dot{\Leftarrow}\rangle$$

where:

$$\alpha_b(P) = \prod_{i=1}^{k} (\mathfrak{p}_i \in P)$$

$$\gamma_b(Q) = \{\mathfrak{p}_i \mid 1 \leq i \leq k \wedge Q_i\}$$

$$Q \mathrel{\dot{\Leftarrow}} Q' = \forall i : 1 \leq i \leq k : Q_i \Leftarrow Q'_i$$

Pointwise Extension to All program Points

By pointwise extension, we have for all program points:

$$\langle \mathcal{L} \longmapsto \wp(\mathbb{P}),\ \dot{\supseteq}\rangle \xleftrightarrows[\dot{\alpha}_b]{\dot{\gamma}_b} \langle \mathcal{L} \longmapsto \prod_{i=1}^{k} \mathbb{B},\ \dot{\Leftarrow}\rangle$$

where:

$$\dot{\alpha}_b(P) = \lambda \ell \cdot \alpha_b(P_\ell)$$

$$\dot{\gamma}_b(Q) = \lambda \ell \cdot \gamma_b(Q_\ell)$$

$$Q \dot{\Leftarrow} Q' = \forall \ell \in \mathcal{L} : Q_\ell \Leftarrow Q'_\ell$$

Composition: Pointwise Boolean Encoded Predicate Abstraction

By composition, we get:

$$\langle \wp(\mathcal{L} \times \mathcal{M}),\ \subseteq\rangle \xleftarrow[\alpha]{\gamma} \langle \mathcal{L} \longmapsto \prod_{i=1}^{k} \mathbb{B},\ \dot{\Leftarrow}\rangle$$

where:

$$\alpha(P) = \dot{\alpha}_b \circ \dot{\alpha}_{\mathbb{P}} \circ \alpha_{\downarrow}(P)$$

$$\gamma(Q) = \gamma_{\downarrow} \circ \dot{\gamma}_{\mathbb{P}} \circ \dot{\gamma}_b(Q)$$

Abstract Predicate Transformer (Sketchy)

$$\alpha_{\mathbb{P}} \circ \mathsf{post}[\![\mathtt{X}:=\mathtt{E}]\!] \circ \gamma_{\mathbb{P}}(\{q_1, \ldots, q_n\}) \quad \text{where } \{q_1, \ldots, q_n\} \subseteq \{\mathfrak{p}_1, \ldots, \mathfrak{p}_k\}$$

$$= \alpha_{\mathbb{P}} \circ \mathsf{post}[\![\mathtt{X}:=\mathtt{E}]\!](\bigcap_{i=1}^{n} \mathcal{I}[\![q_i]\!]) \qquad \text{def. } \gamma_{\mathbb{P}}$$

$$= \alpha_{\mathbb{P}}(\{\rho[\mathtt{X}/[\![\mathtt{E}]\!]\rho] \mid \rho \in \bigcap_{i=1}^{n} \mathcal{I}[\![q_i]\!]\}) \qquad \text{def. } \mathsf{post}[\![\mathtt{X}:=\mathtt{E}]\!]$$

$$= \alpha_{\mathbb{P}}(\bigcap_{i=1}^{n} \{\rho[\mathtt{X}/[\![\mathtt{E}]\!]\rho] \mid \rho \in \mathcal{I}[\![q_i]\!]\}) \qquad \text{def. } \cap$$

$$= \alpha_{\mathbb{P}}(\bigcap_{i=1}^{n} \mathcal{I}[\![q_i[\mathtt{X}/\mathtt{E}]]\!]) \qquad \text{def. substitution}$$

$$= \{\mathfrak{p}_j \mid \mathcal{I}[\![q_i[\mathtt{X}/\mathtt{E}] \Rightarrow \mathfrak{p}_j]\!]\} \qquad \text{def. } \alpha_{\mathbb{P}}$$

$$\Rightarrow \{\mathfrak{p}_j \mid \mathsf{theorem\_prover}[\![q_i[\mathtt{X}/\mathtt{E}] \Rightarrow \mathfrak{p}_j]\!]\}$$

since $\mathsf{theorem\_prover}[\![q_i[\mathtt{X}/\mathtt{E}] \Rightarrow \mathfrak{p}_j]\!]$ implies $\mathcal{I}[\![q_i[\mathtt{X}/\mathtt{E}] \Rightarrow \mathfrak{p}_j]\!]$

2.2.3 Local Completion

See Sec. 9.2 of [POPL '79].

Reference

[POPL '79] P. Cousot & R. Cousot. Systematic design of program analysis frameworks. In *6^th POPL*, pages 269–282, San Antonio, TX, 1979. ACM Press. 31

Non Distributivity [POPL '79]

- An abstraction ρ is $\cup$-complete or distributive, whenever the union of abstract properties is abstract:

$$\forall S \subseteq \wp(\Sigma) : \bigcup_{P \in S} \rho(P) = \rho(\bigcup_{P \in S} \rho(P))$$

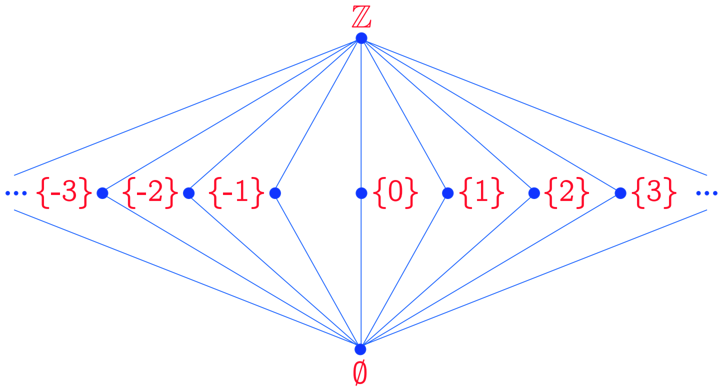
- Hence, the abstract union of abstract properties looses no information with respect to their concrete one;
- Otherwise it is $\cup$-incomplete or non-distributive.

Reference

[POPL '79] P. Cousot & R. Cousot. Systematic design of program analysis frameworks. In *6^{th} POPL*, pages 269–282, San Antonio, TX, 1979. ACM Press. 32

EXAMPLE OF NON DISTRIBUTIVITY [POPL '79]

- Kildall's constant propagation $\langle \{\emptyset, \mathbb{Z}\} \cup \{\{i\} \mid i \in \mathbb{Z}\}, \subseteq \rangle$



is <u>not</u> distributive:

$$\rho(\{1\}) \cup \rho(\{2\}) = \{1, 2\} \neq \mathbb{Z} = \rho(\rho(\{1\}) \cup \rho(\{2\})) \ .$$

Reference

[POPL '79] P. Cousot & R. Cousot. Systematic design of program analysis frameworks. In *6*th *POPL*, pages 269–282, San Antonio, TX, 1979. ACM Press. 33

Disjunctive Completion [POPL '79]

- The $\cup$-completion or disjunctive completion $\mathfrak{C}^{\cup}(\overline{\mathcal{A}})$ of an abstract domain $\overline{\mathcal{A}}$ is the smallest distributive abstract domain containing $\overline{\mathcal{A}}$;
- The disjunctive completion adds all missing joins to the abstract domain:

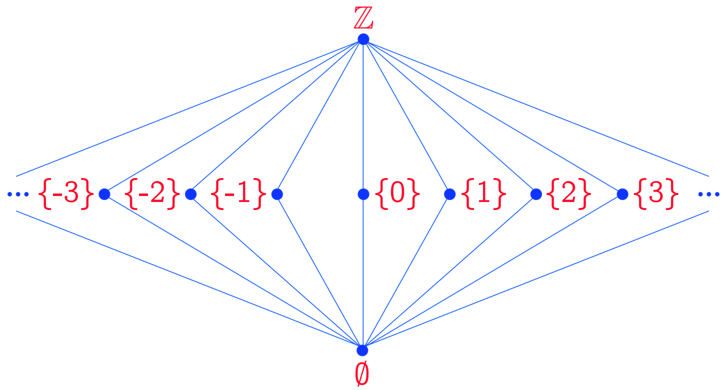
$$\mathfrak{C}^{\cup}(\overline{\mathcal{A}}) = \mathrm{lfp}^{\bar{\mathcal{A}}}_{\subseteq} \lambda A \cdot \mathcal{M}\big(A \cup \{\bigcup_{P \in S} \rho_A(P) \mid \rho_A(\bigcup_{P \in S} \rho_A(P)) \neq \bigcup_{P \in S} \rho_A(P)\}\big)$$

Reference

[POPL '79] P. Cousot & R. Cousot. Systematic design of program analysis frameworks. In *6th POPL*, pages 269–282, San Antonio, TX, 1979. ACM Press. 34

Example of Disjunctive Completion [POPL '79]

- Kildall's constant propagation $\langle \{\emptyset, \mathbb{Z}\} \cup \{\{i\} \mid i \in \mathbb{Z}\}, \subseteq\rangle$



  is not distributive;

- The disjunctive completion is $\langle \wp(\mathbb{Z}), \subseteq\rangle$ (i.e. identity abstraction!).

Reference

[POPL '79] P. Cousot & R. Cousot. Systematic design of program analysis frameworks. In 6^{th} *POPL*, pages 269–282, San Antonio, TX, 1979. ACM Press. 35

Local Image Completeness [POPL '79]

- Given $f \in \wp(\Sigma) \longmapsto \wp(\Sigma)$, the abstraction ρ is f-complete iff the f-transformation of abstract properties is abstract:

$$\forall P \in \wp(\Sigma) : \rho \circ f \circ \rho(P) = f \circ \rho(P)$$

- Hence, the abstract transformation of an abstract property looses no information with respect to the concrete one;
- Otherwise ρ is f-incomplete.

Reference

[POPL '79] P. Cousot & R. Cousot. Systematic design of program analysis frameworks. In *6th POPL*, pages 269–282, San Antonio, TX, 1979. ACM Press. 36

Local Image Completion [5]

- The f-completion $\mathfrak{C}^f(\overline{\mathcal{A}})$ of an abstract domain $\overline{\mathcal{A}}$ is the smallest f-complete abstract domain containing $\overline{\mathcal{A}}$;
- The local image completion adds all missing abstract elements to the abstract domain:

$$\mathfrak{C}^f(\overline{\mathcal{A}}) = \operatorname{lfp}_{\subseteq}^{\bar{\mathcal{A}}} \lambda A \bullet \mathcal{M}\big(A \cup \{f \circ \rho_A(P) \mid \rho_A \circ f \circ \rho_A(P) \neq f \circ \rho_A(P)\}\big) \tag{1}$$

[5] See other completion methods in:

P. Cousot. Partial Completeness of Abstract Fixpoint Checking, invited paper. In *4th Int. Symp. SARA '2000*, LNAI 1864, Springer, pp. 1–25, 2000.

R. Giacobazzi, F. Ranzato, and F. Scozzari. Making abstract interpretations complete. *J. ACM*, 47(2):361–416, 2000.

Fixpoint Completion

- We want to prove $\operatorname{lfp} F \subseteq \gamma(I)$ i.e. $\alpha(\operatorname{lfp} F) \sqsubseteq^\sharp I$
- The abstraction is in general incomplete so $\operatorname{lfp} F^\sharp \not\sqsubseteq^\sharp I$
- Hence we look for the most abstract abstraction $\bar{\alpha}$ which is more precise than α and is fixpoint complete:

$$\bar{\alpha}(\operatorname{lfp} F) = \operatorname{lfp} \bar{F}^\sharp \qquad \text{where} \qquad \bar{F}^\sharp = \bar{\alpha} \circ F \circ \bar{\gamma}$$

- This is sound since $\operatorname{lfp} \bar{F}^\sharp \sqsubseteq^\sharp I$ implies $\alpha(\operatorname{lfp} F) \sqsubseteq^\sharp I$ that is $\operatorname{lfp} F \subseteq \gamma(I)$
- This is complete since $\operatorname{lfp} F \subseteq \bar{\gamma}(I) = \gamma(I)$ so $\bar{\alpha}(\operatorname{lfp} F) \sqsubseteq^\sharp I$ i.e. $\operatorname{lfp} \bar{F}^\sharp \sqsubseteq^\sharp I$ is now provable in the abstract.

Local Image and Domain Completeness

- When $F^\sharp = \bar{\alpha} \circ F \circ \bar{\gamma}$ and $\bar{\rho} = \bar{\gamma} \circ \bar{\alpha}$, the abstract commutation condition $\bar{\alpha} \circ F = F^\sharp \circ \bar{\alpha}$ amounts to *local domain completeness* $\bar{\rho} \circ F = \bar{\rho} \circ F \circ \bar{\rho}$;
- This is different from *local image completeness* $F \circ \bar{\rho} = \bar{\rho} \circ F \circ \bar{\rho}$ for which we provided a completion construction (1) [7];
- A common particular case is when F has an adjoint $\widetilde{F}$ such that $\langle P, \subseteq\rangle \underset{F}{\overset{\widetilde{F}}{\leftrightarrows}} \langle Q, \sqsubseteq\rangle$ in which case adjoined local image completeness $\widetilde{F} \circ \bar{\rho} = \bar{\rho} \circ \widetilde{F} \circ \bar{\rho}$ implies local domain completeness $\bar{\rho} \circ F = \bar{\rho} \circ F \circ \bar{\rho}$.

[7] *Local domain completion* is also possible but more complicated, see R. Giacobazzi, F. Ranzato, and F. Scozzari. Making abstract interpretations complete. *J. ACM*, 47(2):361–416, 2000.

Exact Fixpoint Abstraction by Adjoint Local Image Completion

When F has an adjoint $\widetilde{F}$, a *sufficient condition* to ensure exact fixpoint abstraction $\bar{\alpha}(\operatorname{lfp} F) = \operatorname{lfp} \bar{F}^\sharp$ where $F^\sharp = \bar{\alpha} \circ F \circ \bar{\gamma}$ is:

- Local dual image completeness that is $\widetilde{F} \circ \bar{\gamma} = \bar{\gamma} \circ \bar{\widetilde{F}}^\sharp$ (i.e. $\widetilde{F} \circ \bar{\rho} = \bar{\rho} \circ \widetilde{F} \circ \bar{\rho}$ where $\bar{\rho} = \bar{\gamma} \circ \bar{\alpha}$);
- This can be achieved by refining the original abstract domain $\bar{\rho}$ by the local image fixpoint completion construction (1) [8], [9];
- This implies local domain completeness $\bar{\rho} \circ F = \bar{\rho} \circ F \circ \bar{\rho}$ (i.e. $F \circ \bar{\rho} = \bar{\rho} \circ F \circ \bar{\rho}$);
- This in turn implies exact/precise fixpoint abstraction $\bar{\alpha}(\operatorname{lfp} F) = \operatorname{lfp} \bar{F}^\sharp$ in the refined domain.

---

[8] The local dual image completion can be restricted to the fixpoint iterates.

[9] In general, the completed domain does not satisfy the ascending chain condition (see the previous constant propagation example).

Predicate Abstraction Completion

Principle of refinement for $\dot{\alpha}_{\mathbb{P}}\left(\mathrm{lfp}_{\emptyset}^{\subseteq} \lambda X \cdot I \cup \mathrm{post}[t]X\right)$:

- Start from $\mathbb{P} = \mathbb{P}_0$; (e.g. $\mathbb{P}_0\{\mathrm{true}\}$)
- Iteratively repeat

  Check $\left(\mathrm{lfp}_{\emptyset}^{\subseteq} \lambda X \cdot I \cup \mathrm{post}[t]X\right) \subseteq S$ by pred. abs. $\mathbb{P}_n$

  If failed, do local domain completion of $\mathbb{P}_n$ into $\mathbb{P}_{n+1}$ for adjoint $\widetilde{\mathrm{pre}}[t]$

  until verification done [1];

A few convincing practical experiences e.g. [3]

Reference

[3] T. Ball, R. Majumdar, T.D. Millstein, and S.K. Rajamani. Automatic predicate abstraction of C programs. In *Proc. ACM SIGPLAN 2001 Conf. PLDI. ACM SIGPLAN Not. 36(5)*, pages 203–213. ACM Press, June 2001. 19

[1] convergence has to be enforced by widenings since the problem is undecidable e.g. $n < N$ or “I don’t know”.