On Completeness in Abstract Model Checking from the Viewpoint of Abstract Interpretation

Patrick COUSOT
École Normale Supérieure, 45 rue d'Ulm
75230 Paris cedex 05, France
mailto:Patrick.Cousot@ens.fr http://www.di.ens.fr/~cousot

Réunion Workshop on Implementation of Logics, November 11-12, 2000

Abstraction in Program Analysis & Model Checking

Abstract interpretation has been successfully applied in:

- static program analysis (by approximation of the semantics);
- model checking (state explosion & infinite state models).

Motivations & Results

Abstraction in Model Checking

Main abstractions in model checking:

- Implicit abstraction: to design the model of reference;
- Polyhedral abstraction (with widening): synchronous, real-time & hybrid system verification;
- Finitary abstraction (without widening): hardware & protocole verification [1];

[1] Abstracting concrete transition systems to abstract transition systems so as to reuse existing model checkers in the abstract.

Abstraction in Program Analysis & Model Checking

- The abstraction must always be sound;
- For completeness:
  - in static program analysis: not required (possible uncertainty);
  - in model checking: required [2] (formal verification method [3]).

[2] allowing only for yes/no answers, all uncertainty resulting only from getting out of computer resources.
[3] otherwise model-checking would be a mere debugging method or equivalent to program/model analysis.

Informal Objective of the Talk

- Understand the logical nature of the problem of finding an appropriate abstraction (for proving safety properties).

Discovery of Abstractions

- in static program analysis:
  - task of the program analyzer designer,
  - find a sound abstraction providing useful information for all programs,
  - essentially manual,
  - partially automated e.g. by combination & refinement of abstract domains;
- in model checking:
  - task of the user,
  - find a sound & complete abstraction required to verify one model,
  - looking for automation (e.g. starting from a trivial or user provided guess and refining by trial and error).

Formalization of the Problem

Fixpoint Checking

- Model-checking safety properties of transition systems:

$$\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S\ ?$$

- Program static analysis by abstract interpretation:

$$\gamma(\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot \alpha(I \vee F(\gamma(X)))) \leq S\ ?$$

Soundness / Completeness

Soundness: a positive abstract answer implies a positive concrete answer. So no error is possible when reasoning in the abstract;

Completeness: a positive concrete answer can always be found in the abstract;

Soundness

Soundness: a positive abstract answer implies a positive concrete answer. So no error is possible when reasoning in the abstract;

Soundness / (Partial) Completeness

Soundness: a positive abstract answer implies a positive concrete answer. So no error is possible when reasoning in the abstract;

Completeness: a positive concrete answer can always be found in the abstract;

Partial completeness: in case of termination of the abstract fixpoint checking algorithm, no positive answer can be missed.

Soundness / (Partial) Completeness

Soundness: a positive abstract answer implies a positive concrete answer. So no error is possible when reasoning in the abstract;

Completeness: a positive concrete answer can always be found in the abstract;

Partial completeness: in case of termination of the abstract fixpoint checking algorithm, no positive answer can be missed.

Termination/resource limitation is therefore considered a separate problem (widening/narrowing, etc.).

Objective of the Talk (Formally)

Constructively characterize the abstractions $\langle\alpha, \gamma\rangle$ for which abstract fixpoint algorithms are partially complete.

Practical Question

Is it possible to automatize the discovery of complete abstractions?

Concrete Fixpoint Checking

Concrete Fixpoint Checking Problem

- Complete lattice $\langle L,\ \leq,\ 0,\ 1,\ \vee,\ \wedge\rangle$;
- Monotonic transformer $F \in L \stackrel{\text{mon}}{\longmapsto} L$;
- Specification $\langle I,\ S\rangle \in L^2$;

$$\mathsf{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S\ ?$$

Example

- Set of states: Σ;
- Initial states: $I \subseteq \Sigma$;
- Transition relation: $\tau \subseteq \Sigma \times \Sigma$;
- Transition system: $\langle \Sigma,\ \tau,\ I\rangle$;
- Complete lattice: $\langle \wp(\Sigma),\ \subseteq,\ \emptyset,\ \Sigma,\ \cup,\ \cap\rangle$;
- Right-image of $X \subseteq \Sigma$ by τ:

  $post[\tau](X) \stackrel{\triangle}{=} \{s' \mid \exists s \in X : \langle s,\ s'\rangle \in \tau\}$;
- Reflexive transitive closure of τ: $\tau^\star$

Example (contd.)

- Safety specification: $S \subseteq \Sigma$
- Reachable states from I:

$$post[\tau^\star](I) = \mathsf{lfp}^{\subseteq}\ \boldsymbol{\lambda} X \cdot I \cup post[\tau](X)\ ;$$

- Satisfaction of the safety specification ($post[\tau^\star](I) \subseteq S$):

$$\mathsf{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee post[\tau](X) \leq S\ ?$$

Concrete Fixpoint Checking Algorithm [4]

Algorithm 1

$X := I;\ Go := (X \leq S);$
while Go **do**
  $X' := I \vee F(X);$
  $Go := (X \neq X')\ \&\ (X' \leq S);$
  $X := X';$
od;
return $(X \leq S)$;

[4] P. Cousot & R. Cousot, POPL'77

Partial correctness of Alg. 1

Alg. 1 is partially correct: if it ever terminates then it returns $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$.

Concrete Invariants

- $A \in L$ an *invariant* for $\langle F, I, S\rangle$ if and only if $I \leq A$ & $F(A) \leq A$ & $A \leq S$;

  Note 1 (Floyd's proof method): $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$ if and only if there exists an invariant $A \in L$ for $\langle F, I, S\rangle$;

  Note 2: if Alg. 1 terminates successfully, then it has computed an invariant ($X = \mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X' \cdot I \vee F(X')$).

Galois connection

A *Galois connection*, written

$$\langle L, \leq\rangle \underset{f}{\overset{g}{\leftrightarrows}} \langle M, \sqsubseteq\rangle,$$

is such that:

- $\langle L, \leq\rangle$ and $\langle M, \sqsubseteq\rangle$ are posets;
- the maps $f \in L \mapsto M$ and $g \in M \mapsto L$ satisfy

$$\forall x \in L : \forall y \in M : f(x) \sqsubseteq y \text{ if and only if } x \leq g(y) .$$

Concrete Adjoinedness

In general, F has an *adjoint* $\widetilde{F}$ such that $\langle L, \leq\rangle \underset{F}{\overset{\widetilde{F}}{\leftrightarrows}} \langle L, \leq\rangle$.

Example of Concrete Adjoinedness

- τ^{-1} is the inverse of τ;
- $pre[\tau] \triangleq post[\tau^{-1}]$;
- Set complement $\neg X \triangleq \Sigma \setminus X$;
- $\widetilde{pre}[\tau](X) \triangleq \neg pre[\tau](\neg X)$;

$$\langle \wp(\Sigma),\ \subseteq \rangle \xleftarrow[post[\tau]]{\widetilde{pre}[\tau]} \langle \wp(\Sigma),\ \subseteq \rangle \ .$$

The Complete Lattice of Concrete Invariants

- The set $\mathcal{I}$ of invariants for $\langle F,\ I,\ S \rangle$ is a complete lattice $\langle \mathcal{I},\ \leq,\ \mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X),\ \mathit{gfp}^{\leq}\ \boldsymbol{\lambda} X \cdot S \wedge \widetilde{F}(X),\ \vee,\ \wedge \rangle$.

Fixpoint Concrete Adjoinedness

$$\langle L,\ \leq \rangle \xleftarrow[\boldsymbol{\lambda} I \cdot \mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X)]{\boldsymbol{\lambda} S \cdot \mathit{gfp}^{\leq}\ \boldsymbol{\lambda} X \cdot S \wedge \widetilde{F}(X)} \langle L,\ \leq \rangle$$

Proof:

$$\begin{aligned}
& \mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S \\
\Longleftrightarrow\ & \exists A \in L : I \leq A \ \&\ F(A) \leq A \ \&\ A \leq S && (1) \\
\Longleftrightarrow\ & \exists A \in L : I \leq A \ \&\ A \leq \widetilde{F}(A) \ \&\ A \leq S \\
\Longleftrightarrow\ & I \leq \mathit{gfp}^{\leq}\ \boldsymbol{\lambda} X \cdot S \wedge \widetilde{F}(X) \ .
\end{aligned}$$

□

Dual Concrete Fixpoint Checking Algorithm [5]

Algorithm 2

$Y := S;\ Go := (I \leq Y);$
while Go **do**
  $Y' := S \wedge \widetilde{F}(Y);$
  $Go := (Y \neq Y')\ \&\ (I \leq Y');$
  $Y := Y';$
od;
return $(I \leq Y)$;

[5] P. Cousot, 1981; E.M. Clarke & E.A. Emerson, 1981; J.-P. Queille and J. Sifakis, 1982.

Partial correctness of Alg. 2

Alg. 2 is partially correct: if it ever terminates then it returns $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$.

On (Dual) Fixpoint Checking

$$\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$$

if and only if

$$I \leq \mathit{gfp}^{\leq}\ \boldsymbol{\lambda} X \cdot S \wedge \widetilde{F}(X).$$

if and only if

$$\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq \mathit{gfp}^{\leq}\ \boldsymbol{\lambda} X \cdot S \wedge \widetilde{F}(X)$$

The Adjoined Concrete Fixpoint Checking Algorithm

Algorithm 3

$X := I;\ Y := S;\ \mathit{Go} := (X \leq Y);$
while Go **do**
  $X' := I \vee F(X);\ Y' := S \wedge \widetilde{F}(Y);$
  $\mathit{Go} := (X \neq X')\ \&\ (Y \neq Y')\ \&\ (X' \leq Y');$
  $X := X';\ Y := Y';$
od;
return $(X \leq Y)$;

Partial correctness of Alg. 3

Alg. 3 is partially correct: if it ever terminates then it returns $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$.

Abstract Fixpoint Checking

Example: the Recurrent Abstraction in Abstract Model-Checking

- State abstraction: $h \in \Sigma \mapsto \overline{\Sigma}$;
- Property abstraction: $\alpha_h(X) \stackrel{\triangle}{=} \{h(x) \mid x \in X\} = post[h]$ [6];
- Property concretization: $\gamma_h(Y) \stackrel{\triangle}{=} \{x \mid h(x) \in Y\} = \widetilde{pre}[h]$;
- Galois connection:

$$\langle \wp(\Sigma),\ \subseteq \rangle \underset{\alpha_h}{\overset{\gamma_h}{\leftrightarrows}} \langle \wp(\overline{\Sigma}),\ \subseteq \rangle.$$

- Example (rule of signs): $\Sigma = \mathbb{Z}$ so choose $h(z)$ to be the sign of z.

[6] Considering the function h as a relation.

Abstract Interpretation

- Concrete complete lattice: $\langle L,\ \leq,\ 0,\ 1,\ \vee,\ \wedge \rangle$;
- Abstract complete lattice: $\langle M,\ \sqsubseteq,\ \bot,\ \top,\ \sqcap,\ \sqcup \rangle$;
- Abstraction/concretization pair $\langle \alpha,\ \gamma \rangle$;
- Galois connection:

$$\langle L,\ \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle M,\ \sqsubseteq \rangle.$$

Abstract Fixpoint Checking Algorithm [7]

Algorithm 4

$X := \alpha(I);\ Go := (\gamma(X) \leq S);$
while Go **do**
  $X' := \alpha(I \vee F(\gamma(X)));$
  $Go := (X \neq X')\ \&\ (\gamma(X') \leq S);$
  $X := X';$
od;
return if $(\gamma(X) \leq S)$ **then** *true* **else** *I don't know;*

[7] In P. Cousot & R. Cousot, POPL'77, $(\gamma(X) \leq S)$ is $X \sqsubseteq S'$ where $S' = \alpha(S)$.

Partial correctness of Alg. 4

Alg. 4 is partially correct: if it terminates and returns "*true*" then $\mathsf{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$.

Dual Abstraction

$$\langle L,\ \geq\rangle \underset{\widetilde{\alpha}}{\overset{\widetilde{\gamma}}{\leftrightarrows}} \langle M,\ \sqsupseteq\rangle.$$

Example of Dual Abstraction

If

- $\langle L,\ \leq,\ 0,\ 1,\ \vee,\ \wedge,\ \neg\rangle$ is a complete boolean lattice;
- $\langle M,\ \sqsubseteq,\ \bot,\ \top,\ \sqcap,\ \sqcup,\ \backsim\rangle$ is a complete boolean lattice;
- $\langle L,\ \leq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle M,\ \sqsubseteq\rangle$;
- $\widetilde{\alpha} \stackrel{\triangle}{=} \backsim \circ\, \alpha \circ \neg$ and $\widetilde{\gamma} \stackrel{\triangle}{=} \neg \circ \gamma \circ \backsim$

then

$$\langle L,\ \geq\rangle \underset{\widetilde{\alpha}}{\overset{\widetilde{\gamma}}{\leftrightarrows}} \langle M,\ \sqsupseteq\rangle$$

Example of Dual Abstraction (Contd.)

For the recurrent abstraction in abstract model-checking $\alpha_h(X) \stackrel{\triangle}{=} \{h(x) \mid x \in X\} = post[h]$ we have:

- $\langle \wp(\Sigma),\ \subseteq\rangle \underset{post[h]}{\overset{\widetilde{pre}[h]}{\leftrightarrows}} \langle \wp(\Sigma),\ \subseteq\rangle$;
- $\widetilde{pre}[h](X) = \neg pre[h](\neg X)$ and $\widetilde{post}[h](X) = \neg post[h](\neg X)$, so:
- $\langle \wp(\Sigma),\ \supseteq\rangle \underset{\widetilde{post}[h]}{\overset{pre[h]}{\leftrightarrows}} \langle \wp(\Sigma),\ \supseteq\rangle$.

Abstract Adjoinedness

$\langle L, \leq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle M, \sqsubseteq\rangle$, $\langle L, \leq\rangle \underset{F}{\overset{\widetilde{F}}{\leftrightarrows}} \langle L, \leq\rangle$ and $\langle L, \geq\rangle \underset{\widetilde{\alpha}}{\overset{\widetilde{\gamma}}{\leftrightarrows}} \langle M, \sqsupseteq\rangle$ imply:

$$\langle M, \sqsubseteq\rangle \underset{\alpha\circ F\circ\widetilde{\gamma}}{\overset{\widetilde{\alpha}\circ\widetilde{F}\circ\gamma}{\leftrightarrows}} \langle M, \sqsubseteq\rangle.$$

The Dual Abstract Fixpoint Checking Algorithm

Algorithm 5

$Y := \widetilde{\alpha}(S)$; $Go := (I \leq \widetilde{\gamma}(Y))$;
while Go **do**
  $Y' := \widetilde{\alpha}(S \wedge \widetilde{F}(\widetilde{\gamma}(Y)))$;
  $Go := (Y \neq Y')\ \&\ (I \leq \widetilde{\gamma}(Y'))$;
  $Y := Y'$;
od;
return if $(I \leq \widetilde{\gamma}(Y))$ **then** *true* **else** *I don't know;*

Partial correctness of Alg. 5

Alg. 5 is partially correct: if it terminates and returns "*true*" then $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$.

The Particular Case of Complement Abstraction

1. $\langle L, \leq, 0, 1, \vee, \wedge, \neg\rangle$ is a complete boolean lattice;
2. $\langle M, \sqsubseteq, \bot, \top, \sqcup, \sqcap, \backsim\rangle$ is a complete boolean lattice;
3. $\langle L, \leq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle M, \sqsubseteq\rangle$;
4. $\langle L, \leq\rangle \underset{F}{\overset{\widetilde{F}}{\leftrightarrows}} \langle L, \leq\rangle$;
5. $\widetilde{F} \triangleq \neg \circ F \circ \neg$, $\widetilde{\alpha} \triangleq \backsim \circ \alpha \circ \neg$ and $\widetilde{\gamma} \triangleq \neg \circ \gamma \circ \backsim$.

The Contrapositive Abstract Fixpoint Checking Algorithm

Alg. 5 becomes:

Algorithm 6

$Z := \alpha(\neg S);\ Go := (I \wedge \gamma(Z) = 0);$
while Go **do**
  $Z' := \alpha(\neg S \vee F(\gamma(Z)));$
  $Go := (Z \neq Z')\ \&\ (I \wedge \gamma(Z') = 0);$
  $Z := Z';$
od;
return if $(I \wedge \gamma(Z) = 0)$ **then** *true* **else** *I don't know;*

The Adjoined Abstract Fixpoint Checking Algorithm

Algorithm 7

$X := \alpha(I);\ Y := \widetilde{\alpha}(S);\ Go := (\gamma(X) \leq S)\ \&\ (I \leq \widetilde{\gamma}(Y));$
while Go **do**
  $X' := \alpha(I \vee F \circ \gamma(X));\ Y' := \widetilde{\alpha}(S \wedge \widetilde{F} \circ \widetilde{\gamma}(Y));$
  $Go := (X \neq X')\ \&\ (Y \neq Y')\ \&\ (\gamma(X') \leq S)\ \&\ (I \leq \widetilde{\gamma}(Y'));$
  $X := X';\ Y := Y';$
od;
return if $(\gamma(X) \leq S) \mid (I \leq \widetilde{\gamma}(Y))$ **then** *true*
else *I don't know;*

Partial correctness of Alg. 6

Alg. 6 is partially correct: if it terminates and returns "*true*" then $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$.

Partial correctness of Alg. 7

Alg. 7 is partially correct: if it terminates and returns "*true*" then $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$.

Further Requirements for Program Static Analysis

- In program static analysis, one *cannot* compute γ, $\widetilde{\gamma}$ and $\leq$ and sometimes neither I nor S may even be machine representable;
- So Alg. 7, which can be useful in model-checking, is of *limited interest* in program static analysis;
- Such problems do no appear in abstract model checking since the concrete model is almost always machine-representable (although sometimes too large).

Example: the Recurrent Abstraction in Abstract Model-Checking

Continuing with the abstraction of p. 31 with

$$\begin{array}{lll} & \alpha \triangleq post[h] & \gamma \triangleq \widetilde{pre}[h] \\ \text{and} & \widetilde{\alpha} \triangleq \widetilde{post}[h] & \widetilde{\gamma} \triangleq pre[h], \end{array}$$

we have:

1. $\forall X \in L : \gamma \circ \widetilde{\alpha}(X) \subseteq X$;
2. $\forall X \in L : X \subseteq \widetilde{\gamma} \circ \alpha(X)$.

Additional Hypotheses

In order to be able to check termination in the abstract, we assume:

1. $\forall X \in L : \gamma \circ \widetilde{\alpha}(X) \leq X$;
2. $\forall X \in L : X \leq \widetilde{\gamma} \circ \alpha(X)$.

The Adjoined Abstract Fixpoint Abstract Checking Algorithm

Algorithm 8

$X := \alpha(I);\ Y := \widetilde{\alpha}(S);\ Go := (X \sqsubseteq Y);$
while Go **do**
  $X' := \alpha(I) \sqcup \alpha \circ F \circ \gamma(X);\ Y' := \widetilde{\alpha}(S) \sqcap \widetilde{\alpha} \circ \widetilde{F} \circ \widetilde{\gamma}(Y);$
  $Go := (X \neq X')\ \&\ (Y \neq Y')\ \&\ (X' \sqsubseteq Y');$
  $X := X';\ Y := Y';$
od;
return if $X \sqsubseteq Y$ **then** *true* **else** *I don't know;*

Partial correctness of Alg. 8

Alg. 8 is partially correct: if it ever terminates and returns "*true*" then $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$.

Partially Complete Abstraction (definition) [8]

Definition 9 The abstraction $\langle \alpha,\ \gamma \rangle$ is *partially complete* if, whenever Alg. 4 terminates and $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$ then the returned result is "*true*".

[8] Observe that this notion of *partial completeness* is different from the notions of *fixpoint completeness* ($\alpha(\mathit{lfp}^{\leq}\ G) = \mathit{lfp}^{\sqsubseteq}\ \alpha \circ G \circ \gamma$) and the stronger one of *local completeness* ($\alpha \circ G = \alpha \circ G \circ \gamma \circ \alpha$) considered in P. Cousot & R. Cousot, POPL'79.

Partially complete abstraction

Characterization of Partially Complete Abstractions for Algorithm 4

Theorem 10 The abstraction $\langle \alpha,\ \gamma \rangle$ is partially complete for Alg. 4 if and only if $\alpha(L)$ contains an abstract value A such that $\gamma(A)$ is an invariant for $\langle F,\ I,\ S \rangle$.

Characterization of Partially Complete Abstractions for Algorithm 4

Theorem 10 The abstraction $\langle \alpha, \gamma \rangle$ is partially complete for Alg. 4 if and only if $\alpha(L)$ contains an abstract value A such that $\gamma(A)$ is an invariant for $\langle F, I, S \rangle$.

Intuition: *finding a partially complete abstraction is logically equivalent to making an invariance proof.*

The Most Abstract Partially Complete Abstraction (Definition)

Definition 11 The *most abstract partially complete abstraction* $\langle \overline{\alpha}, \overline{\gamma} \rangle$, if it exists, is defined such that:

1. The *abstract domain* $\overline{M} = \overline{\alpha}(L)$ has the smallest possible cardinality;
2. If another abstraction $\langle \alpha', \gamma' \rangle$ is a partially complete abstraction with the same cardinality, then there exists a bijection β such that $\forall x \in \overline{M} : \gamma'(\beta(x)) \leq \overline{\gamma}(x)$ [9].

[9] Otherwise stated, the abstract values in $\overline{\alpha}(L)$ are more approximate than the corresponding elements in $\alpha'(L)$.

Characterization of the Most Abstract Complete Abstraction

Theorem 12 The most abstract partially complete abstraction for Alg. 4 is such that:

- if $S = 1$ then $\overline{M} = \{\top\}$ where $\overline{\alpha} \triangleq \boldsymbol{\lambda} X \cdot \top$ and $\overline{\gamma} \triangleq \boldsymbol{\lambda} Y \cdot 1$;
- if $S \neq 1$ then $\overline{M} = \{\bot, \top\}$ where $\bot \sqsubseteq \bot \sqsubset \top \sqsubseteq \top$ with $\langle \overline{\alpha}, \overline{\gamma} \rangle$ such that:

$$
\begin{aligned}
\overline{\alpha}(X) &\triangleq \text{if } X \leq \mathit{gfp}^{\leq}\ \boldsymbol{\lambda} X \cdot S \wedge \widetilde{F}(X) \text{ then } \bot \text{ else } \top \\
\overline{\gamma}(\bot) &\triangleq \mathit{gfp}^{\leq}\ \boldsymbol{\lambda} X \cdot S \wedge \widetilde{F}(X) \qquad (2) \\
\overline{\gamma}(\top) &\triangleq 1
\end{aligned}
$$

The Least Abstract Partially Complete Abstraction (Definition)

Definition 13 Dually, the *least abstract partially complete abstraction* $\langle \overline{\alpha}, \overline{\gamma} \rangle$, if it exists, is defined such that:

1. The *abstract domain* $\overline{M} = \overline{\alpha}(L)$ has the smallest possible cardinality;
2. If another abstraction $\langle \alpha', \gamma' \rangle$ is a partially complete abstraction with the same cardinality, then there exists a bijection β such that $\forall x \in \overline{M} : \overline{\gamma}(x) \leq \gamma'(\beta(x))$ [10].

[10] Otherwise stated, the abstract values in $\overline{\alpha}(L)$ are less approximate than the corresponding elements in $\alpha'(L)$.

Characterization of the Least Abstract Complete Abstraction

Theorem 14 Dually, the least abstract partially complete abstraction for Alg. 4 is such that:

- if $I = 1$ then $\underline{M} = \{\top\}$ where $\underline{\alpha} \stackrel{\triangle}{=} \boldsymbol{\lambda} X \cdot \top$ and $\underline{\gamma} \stackrel{\triangle}{=} \boldsymbol{\lambda} Y \cdot 1$;
- if $I \neq 1$ then $\underline{M} = \{\bot, \top\}$ where $\bot \sqsubseteq \bot \sqsubset \top \sqsubseteq \top$ with $\langle \underline{\alpha},\ \underline{\gamma} \rangle$ such that:

$$\begin{aligned}\underline{\alpha}(X) &\stackrel{\triangle}{=} \text{if } X \leq \mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \text{ then } \bot \text{ else } \top \\ \underline{\gamma}(\bot) &\stackrel{\triangle}{=} \mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \qquad (3) \\ \underline{\gamma}(\top) &\stackrel{\triangle}{=} 1\end{aligned}$$

The Complete Lattice of Minimal Complete Abstractions for Alg. 4

Theorem 16

- The relation $\langle \{\bot, \top\},\ \sqsubseteq,\ \alpha,\ \gamma \rangle \preceq \langle \{\bot', \top'\},\ \sqsubseteq',\ \alpha',\ \gamma' \rangle$ if and only if $\gamma(\bot) \leq \gamma'(\bot')$ is a pre-ordering on $\mathcal{A}$.
- Let $\langle \{\bot, \top\},\ \sqsubseteq,\ \alpha,\ \gamma \rangle \cong \langle \{\bot', \top'\},\ \sqsubseteq',\ \alpha',\ \gamma' \rangle$ if and only if $\gamma(\bot) = \gamma'(\bot')$ be the corresponding equivalence.
- The quotient $\mathcal{A}_{/\cong}$ is a complete lattice [11] for $\preceq$ with infimum class representative $\langle \underline{M},\ \underline{\sqsubseteq},\ \underline{\alpha},\ \underline{\gamma} \rangle$ and supremum $\langle \overline{M},\ \overline{\sqsubseteq},\ \overline{\alpha},\ \overline{\gamma} \rangle$.

[11] Observe however that it is not a sublattice of the lattice of abstract interpretations of P. Cousot & R. Cousot, POPL'77, POPL'79 with reduced product as glb.

The Minimal Partially Complete Abstractions for Algorithm 4

Theorem 15

- The set $\mathcal{A}$ of partially complete abstractions of minimal cardinality for Alg. 4 is the set of all abstract domains $\langle M,\ \sqsubseteq,\ \alpha,\ \gamma \rangle$ such that $M = \{\bot, \top\}$ with $\bot \sqsubseteq \bot \sqsubseteq \top \sqsubseteq \top$, $\langle L,\ \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle M,\ \sqsubseteq \rangle$, $\gamma(\bot) \in \mathcal{I}$ and $\bot = \top$ if and only if $\gamma(\top) \in \mathcal{I}$.

Intuition for Minimal Partially Complete Abstractions

- There is a one to one correspondance between partially complete abstractions of minimal cardinality for Alg. 4 and the set of invariants for proving $\mathit{lfp}^{\leq}\ \boldsymbol{\lambda} X \cdot I \vee F(X) \leq S$;
- Similar results hold for the other Algs. 6, 7 & 8.

Conclusion

On the Automatic Inference of Partially Complete Abstractions

- The automatic inference/refinement of abstractions is an active subject of research [12];
- Automating the abstraction is logically equivalent to discovering an invariant and checking a proof obligation (Th. 10);

[12] Graf & Loiseaux, CAV'93; Loiseaux, Graf, Sifakis, Bouajjani & Bensalem FMSD(6:1)'95, Graf & Saïdi, CAV'97; Bensalem, Lakhnech & Owre CAV'98; Colon & Uribe, CAV'98; Abdulla, Annichini, Bensalem, Bouajjani, Habermehl & Lakhnech, CAV'99; Das, Dill & Park, CAV'99; Saïdi & Shankar, CAV'99; Saïdi, SAS'00; Baumgartner, Tripp, Aziz, Singhal & Andersen, CAV'00; Clarke, Grumberg, Jha, Lu & Veith, CAV'00; etc.

On the Automatic Inference of Partially Complete Abstractions

- The automatic inference/refinement of abstractions is an active subject of research [12];

[12] Graf & Loiseaux, CAV'93; Loiseaux, Graf, Sifakis, Bouajjani & Bensalem FMSD(6:1)'95, Graf & Saïdi, CAV'97; Bensalem, Lakhnech & Owre CAV'98; Colon & Uribe, CAV'98; Abdulla, Annichini, Bensalem, Bouajjani, Habermehl & Lakhnech, CAV'99; Das, Dill & Park, CAV'99; Saïdi & Shankar, CAV'99; Saïdi, SAS'00; Baumgartner, Tripp, Aziz, Singhal & Andersen, CAV'00; Clarke, Grumberg, Jha, Lu & Veith, CAV'00; etc.

On the Automatic Inference of Partially Complete Abstractions

- The automatic inference/refinement of abstractions is an active subject of research [12];
- Automating the abstraction is logically equivalent to discovering an invariant and checking a proof obligation (Th. 10);
- After immoderate hopes in the seventies, there was no breakthrough for the last 20 years in automatic program proving;

[12] Graf & Loiseaux, CAV'93; Loiseaux, Graf, Sifakis, Bouajjani & Bensalem FMSD(6:1)'95, Graf & Saïdi, CAV'97; Bensalem, Lakhnech & Owre CAV'98; Colon & Uribe, CAV'98; Abdulla, Annichini, Bensalem, Bouajjani, Habermehl & Lakhnech, CAV'99; Das, Dill & Park, CAV'99; Saïdi & Shankar, CAV'99; Saïdi, SAS'00; Baumgartner, Tripp, Aziz, Singhal & Andersen, CAV'00; Clarke, Grumberg, Jha, Lu & Veith, CAV'00; etc.

On the Automatic Inference of Partially Complete Abstractions (contd.)

Will the empirical methods (presently) used in abstract model-checking be able to automatize the discovery of partially complete abstractions? [13]

13 May be not so abstract model-checking will eventually boils down to incomplete abstract interpretations as used in program analysis or program debugging using a simultaneous simulation of program executions (although the current per-example reasoning can go on for ever).

THE END, THANK YOU.

Reference: P. Cousot. *Partial Completeness of Abstract Fixpoint Checking*. Proc. 4th Int. Symp. SARA'2000, LNAI 1864, pp. 1–25, Springer-Verlag, Jul. 2000.

THE END