Abstract Interpretation: "Scene-Setting Talk"

Patrick Cousot

cims.nyu.edu/~pcousot

Dagstuhl Seminar 14352

Next Generation Static Software Analysis Tools

August 24th – August 29th 2014,

Motivation

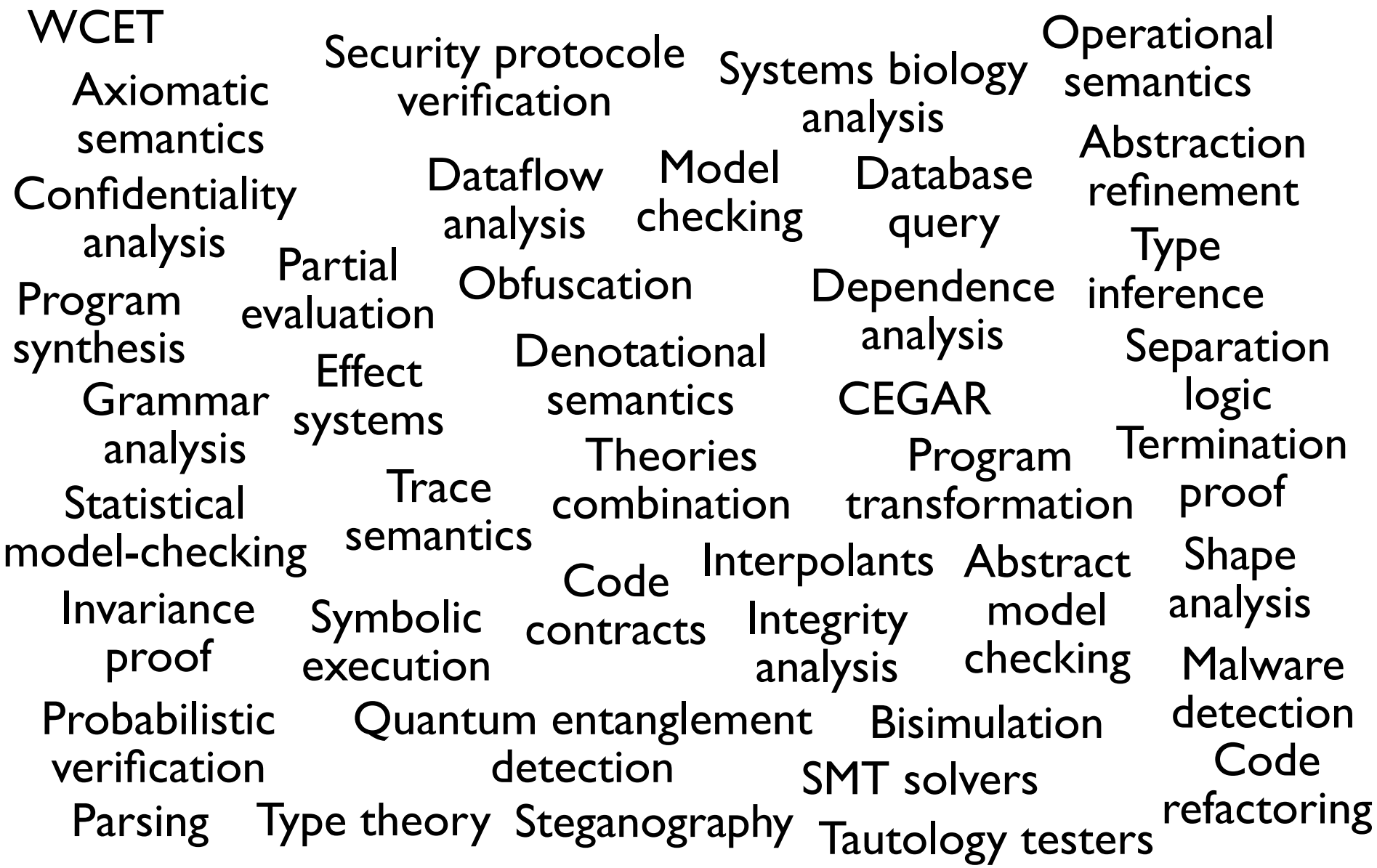
Fundamental motivations

Trends in formal methods research

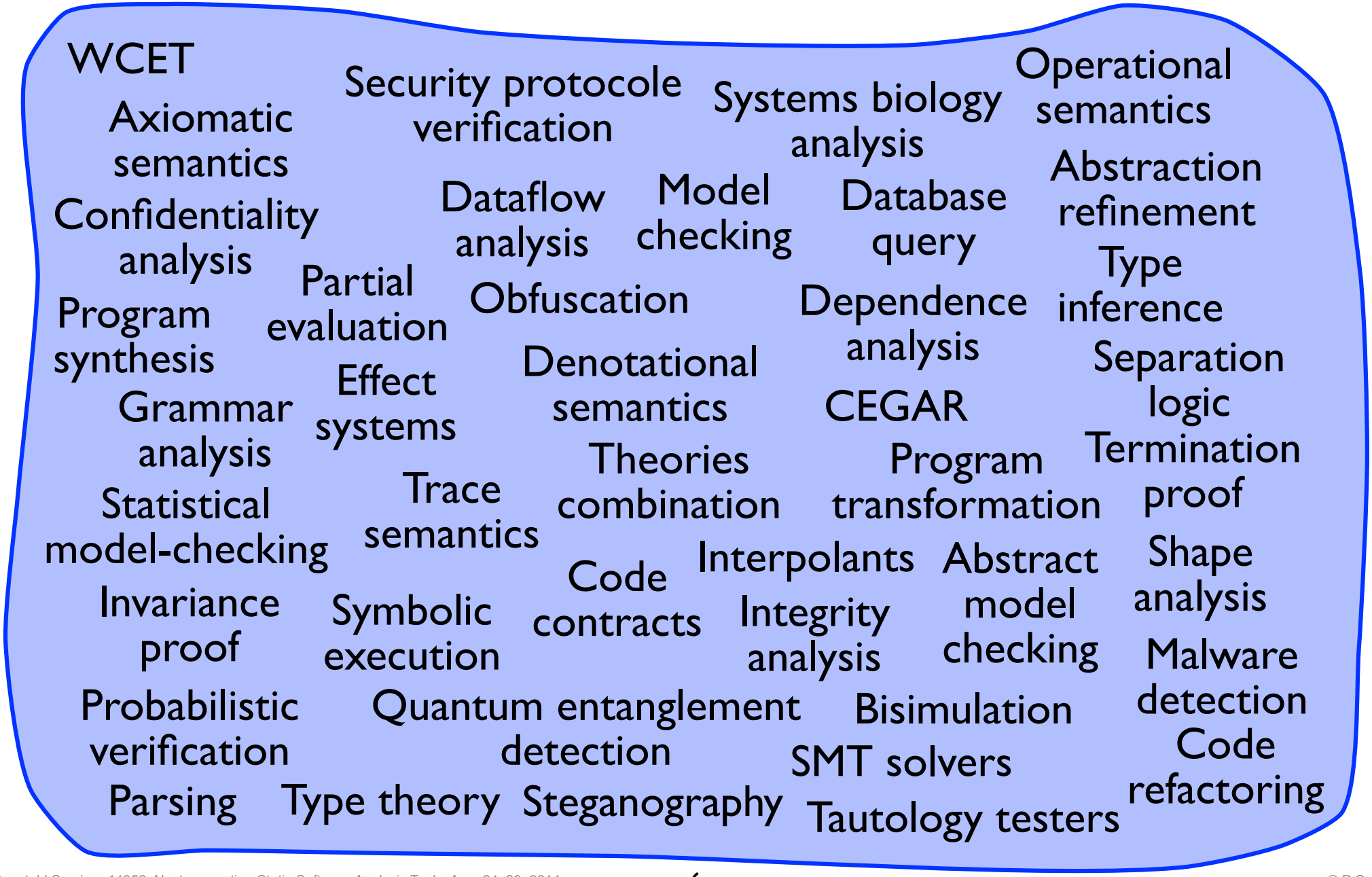
- dispersion and parcelization through a collection of local techniques for specific applications
- we should aim at unification and synthesis through universal principles
- that's the whole purpose of abstract interpretation
  - abstraction is a unifier of formal methods
  - with practical applications

This is what we concentrate on in this seminar

Formal methods for program verification

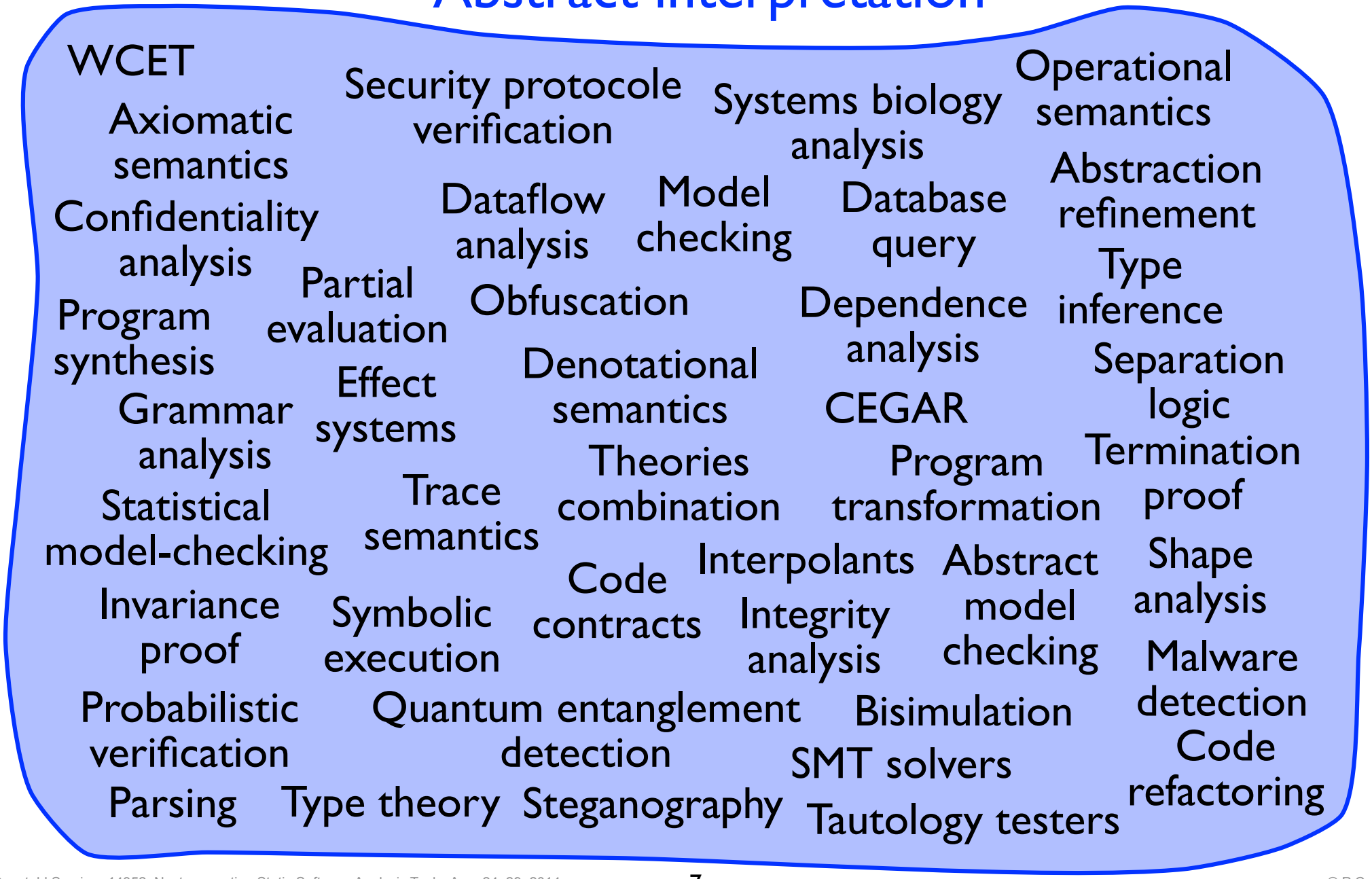


Formal methods for program verification



Formal methods for program verification

Abstract interpretation

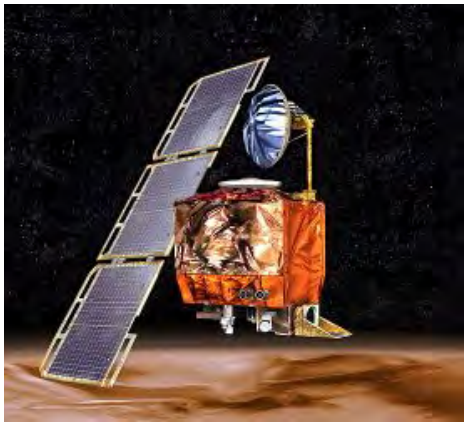


Practical motivations

All computer scientists have experienced bugs

Ariane 5.01 failure
(overflow)

Patriot failure
(float rounding)



Mars orbiter loss
(unit error)

Heartbleed
(buffer overrun)

- Checking the presence of bugs by debugging is great
- Proving their absence by static analysis is even better!
- Undecidability and complexity is the challenge for automation

Induction

Program verification by induction

- Program verification is by induction
  - Program steps (Turing, Floyd, Naur)
  - Structural (program syntactic structure, Strachey, Hoare)
  - Fixpoints (Scott)
  - Data structures (Burstall)
  - Segmentation hierarchies (*)

(*) Cousot, P., Cousot, R.: An abstract interpretation framework for termination. In: POPL. 245–258. ACM (2012)

Program verification by induction

- The only non-trivial base case is iteration/recursion (to be handled by induction):
  - concrete domain $\langle \mathscr{D}, \subseteq, \bot, \cup \rangle$ (poset)
  - transformer $F \in \mathscr{D} \mapsto \mathscr{D}$ (increasing)
  - specification $S \in \mathscr{D}$
  - proof $\mathbf{lfp}^{\subseteq} F \subseteq S$
- Example: Σ states, $\langle \wp(\Sigma), \subseteq, \varnothing, \cup \rangle$ complete lattice of properties, F: Floyd's verification conditions, S is an invariant, proof: reachable states satisfy the invariant S.

Abstraction

- The concrete domain $\mathscr{D}$ is in general *not machine representable*. Abstracted into
  - abstract domain $\langle \overline{\mathscr{D}}, \overline{\sqsubseteq}, \overline{\bot}, \overline{\sqcup} \rangle$ (poset)
  - concretization $\gamma \in \overline{\mathscr{D}} \mapsto \mathscr{D}$ (increasing)

- E.g.: Hoare logic uses $\langle$first-order predicates, $\Longrightarrow\rangle$

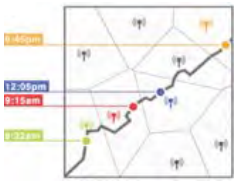
Homomorphic abstractions of a man / crowd

Height

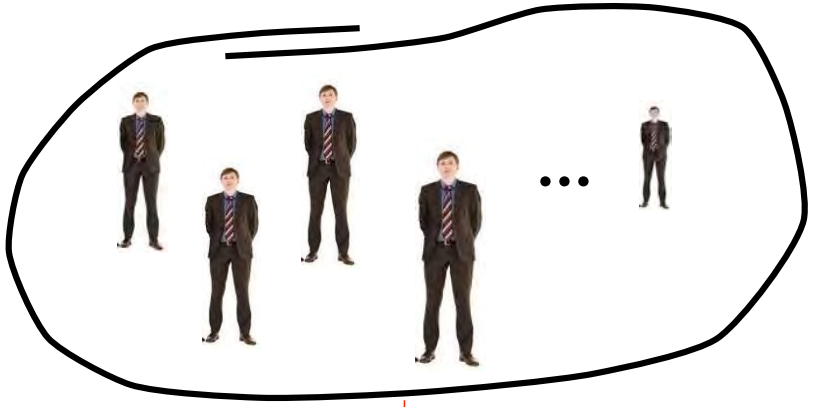
Fingerprint

Eye color

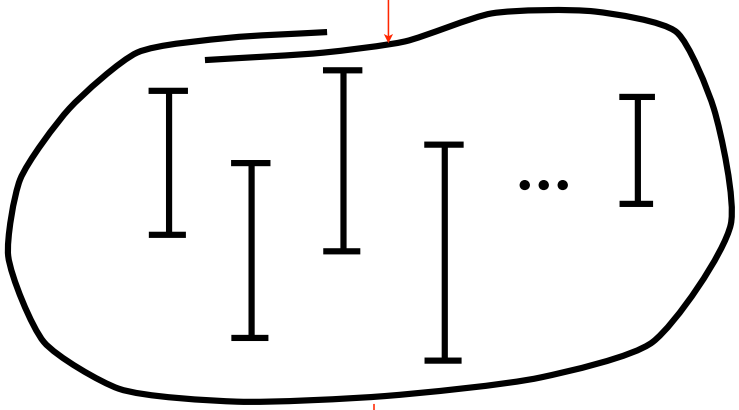
DNA



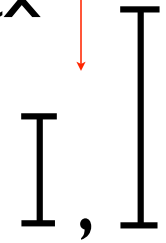
Phone metadata



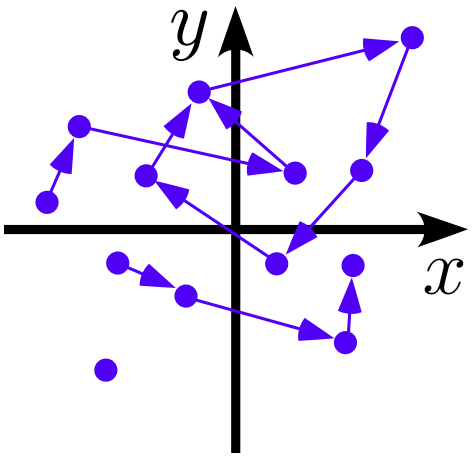
Individual heights



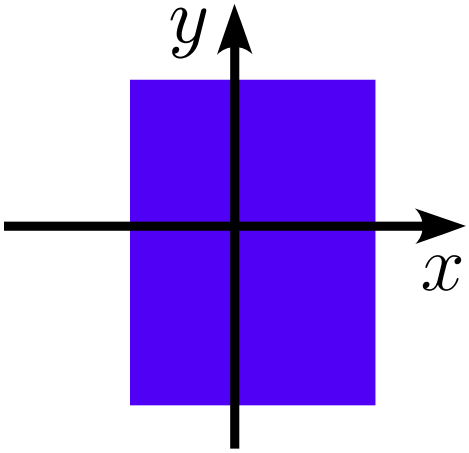
min, max



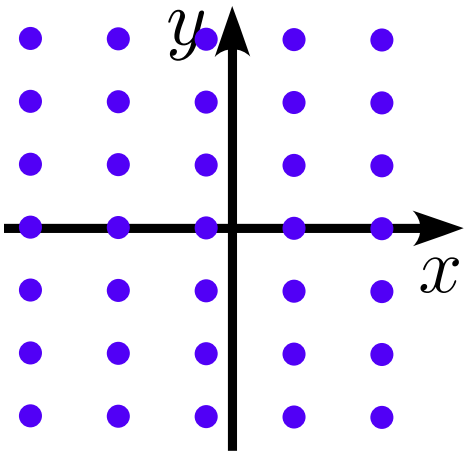
Numerical abstractions in Astrée



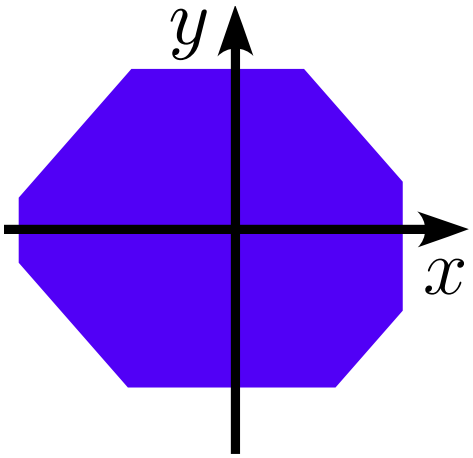
Collecting semantics:
partial traces



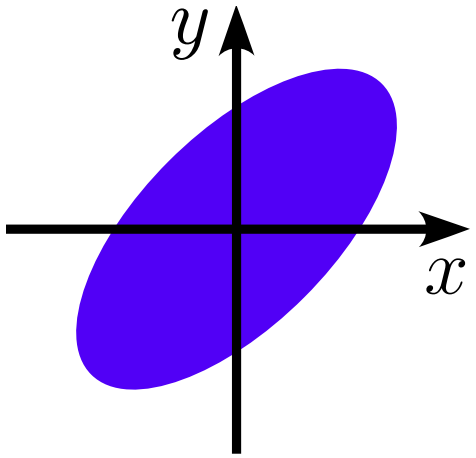
Intervals:
$\mathtt{x} \in [a, b]$



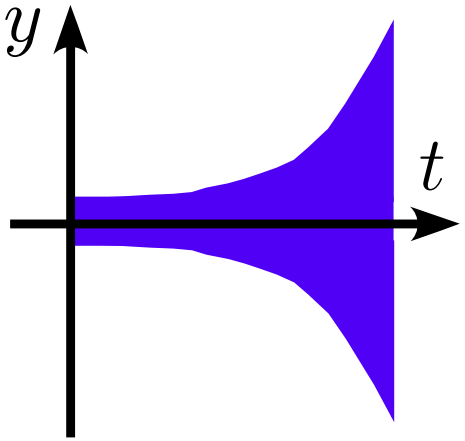
Simple congruences:
$\mathtt{x} \equiv a[b]$



Octagons:
$\pm\mathtt{x} \pm \mathtt{y} \leqslant a$



Ellipses:
$\mathtt{x}^2 + b\mathtt{y}^2 - a\mathtt{xy} \leqslant d$



Exponentials:
$-a^{bt} \leqslant \mathtt{y}(t) \leqslant a^{bt}$

Why abstraction may be approximate?

- Example

$\{ x = y \wedge 0 \leqslant x \leqslant 10 \}$

```
x := x - y;
```

$\{ x = 0 \wedge 0 \leqslant y \leqslant 10 \}$

Interval abstraction:

$\{ x \in [0, 10] \wedge y \in [0, 10] \}$

```
x := x - y;
```

$\{ x \in [-10, 10] \wedge y \in [0, 10] \}$

(but for constants, the interval abstraction can't express equality)

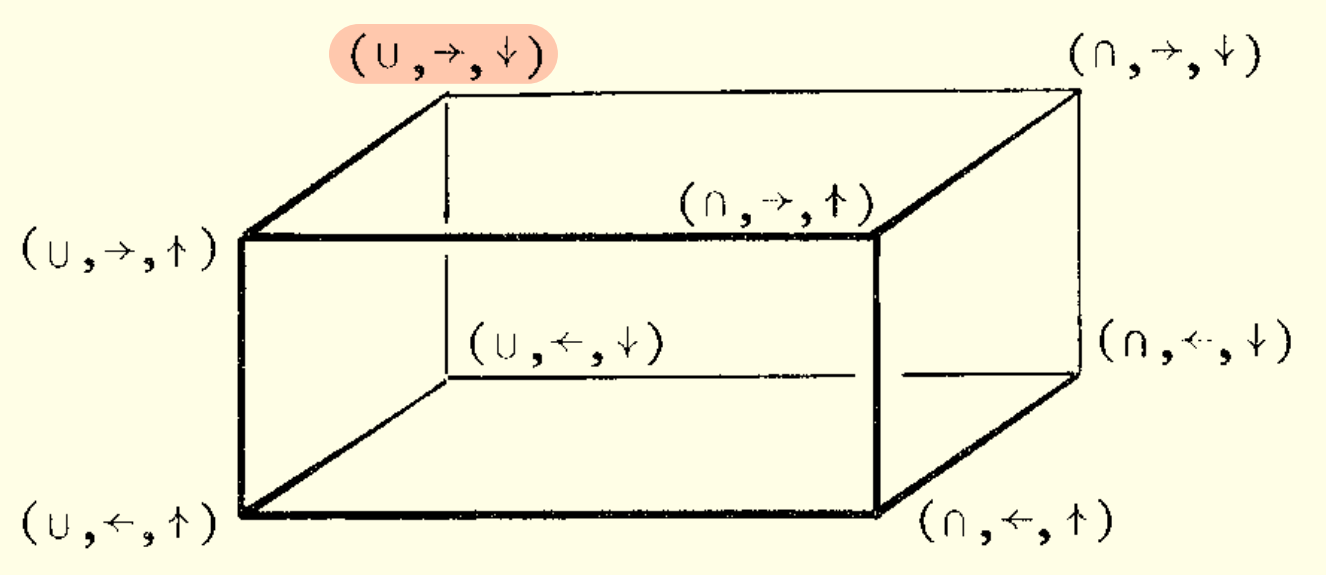
Why abstraction may be approximate?

- Hoare logic: loop invariants may not be expressible in the first-order logic
- Relative completeness only (under the expressiveness hypothesis that the loop invariants are expressible in the first-order logic)

Abstraction (cont'd)

- The concrete domain $\mathscr{D}$ is in general not machine representable. Abstracted into
  - abstract domain $\langle \overline{\mathscr{D}}, \sqsubseteq, \overline{\bot}, \overline{\sqcup} \rangle$ (poset)
  - concretization $\gamma \in \overline{\mathscr{D}} \mapsto \mathscr{D}$ (increasing)
  - abstract transformer $\overline{F} \in \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$
  - semi-commutation $F \circ \gamma \dot{\subseteq} \gamma \circ \overline{F}$
  - abstract specification $\overline{S} \in \overline{\mathscr{D}}$
  - inductive argument $\exists\, \overline{I} \in \overline{\mathscr{D}} : \; \overline{F}(\overline{I}) \sqsubseteq \overline{I}$
  - proof $\overline{I} \sqsubseteq \overline{S}$
  - soundness $\mathbf{lfp}^{\subseteq} F \subseteq \gamma(\overline{S})$

Duality

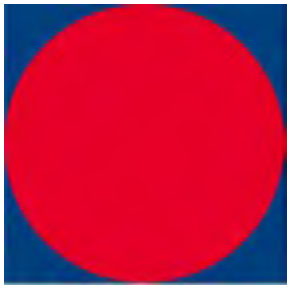


- Order duality: join (∪) or meet (∩)
- Inversion duality: forward (→) or backward (← = $(\rightarrow)^{-1}$)
- Fixpoint duality: least (↓) or greatest (↑)

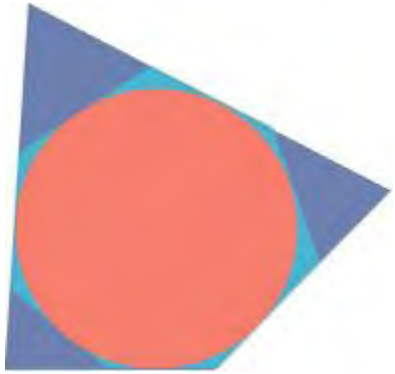
Patrick Cousot, Radhia Cousot: **Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints.** POPL 1977: 238-252

In general, no best abstraction

- Best abstraction of a disk by a rectangular parallelogram



- No best abstraction of a disk by a polyhedron (Euclid)



use only concretization [I]

[I] Patrick Cousot, Radhia Cousot: Abstract Interpretation Frameworks. J. Log. Comput. 2(4): 511-547 (1992)

Mathematicians proceed by induction

- The solution $\bar{I}$ to the constraints $\bar{F}(X) \sqsubseteq X$ on $\langle \bar{\mathscr{D}}, \sqsubseteq \rangle$ is computed iteratively:
  - $\bar{I}^0 = \bar{D}$ ($\bar{D}$ initial guess)
  - $\bar{I}^1 = \bar{F}(\bar{I}^0)$
  - $\bar{I}^n = \mathscr{I}(\bar{I}^0, \bar{F}, \bar{S}, \sqsubseteq, n)$ induction hypothesis
  - $\bar{I}^{n+1} = \bar{F}(\bar{I}^n) = \bar{F}(\mathscr{I}(\bar{I}^0, \bar{F}, \bar{S}, \sqsubseteq, n)) = \mathscr{I}(\bar{I}^0, \bar{F}, \bar{S}, \sqsubseteq, n+1)$

    induction
  - by recurrence: $\forall n \in \mathbb{N} : \bar{I}^n = \mathscr{I}(\bar{I}^0, \bar{F}, \bar{S}, \sqsubseteq, n))$
  - $\bar{I} = \lim_{n \to \infty} \mathscr{I}(\bar{I}^0, \bar{F}, \bar{S}, \sqsubseteq, n)$

Static analysis

- Static analysis must do some form of abstract induction (abstracting away the mathematical proof by recurrence)

Static analysis in Noetherian domains

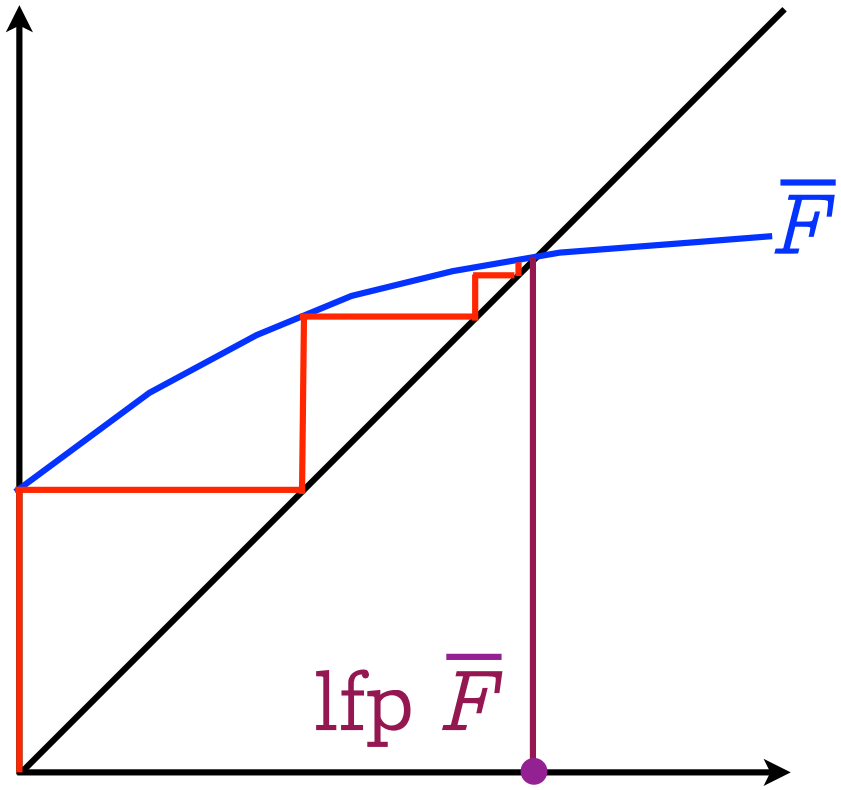
- The abstract domain is Noetherian: finite (which is equivalent to *predicate abstraction*(*)) or satisfies the ascending/descending chain condition
- The induction is predefined in the abstract domain (as defined by successive joins/meets which always converge in finitely many steps)
- Easy, inexpressive (the induction hypothesis is fully determined by the abstraction)

(*) Cousot, P.: Verification by abstract interpretation. In: Verification: Theory and Practice. LNCS 2772, 243–268. Springer (2003)

Static analysis in infinite domains

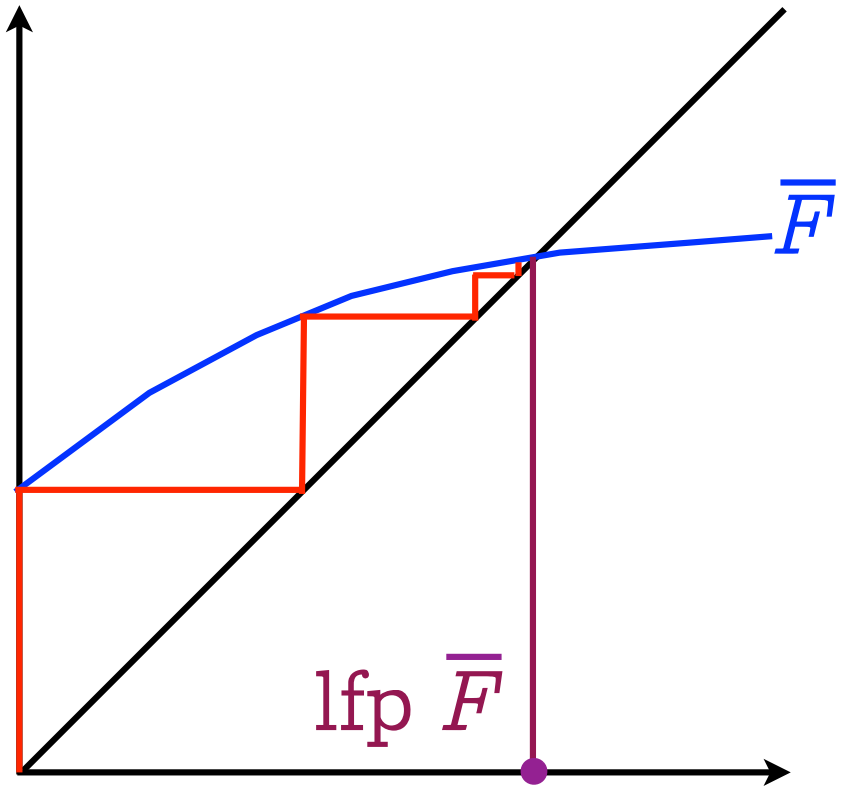
- Some form of abstract induction is needed
- Can be join/meet but iterates do not converge
- Must accelerate the convergence of the iterates by approximation

Convergence acceleration

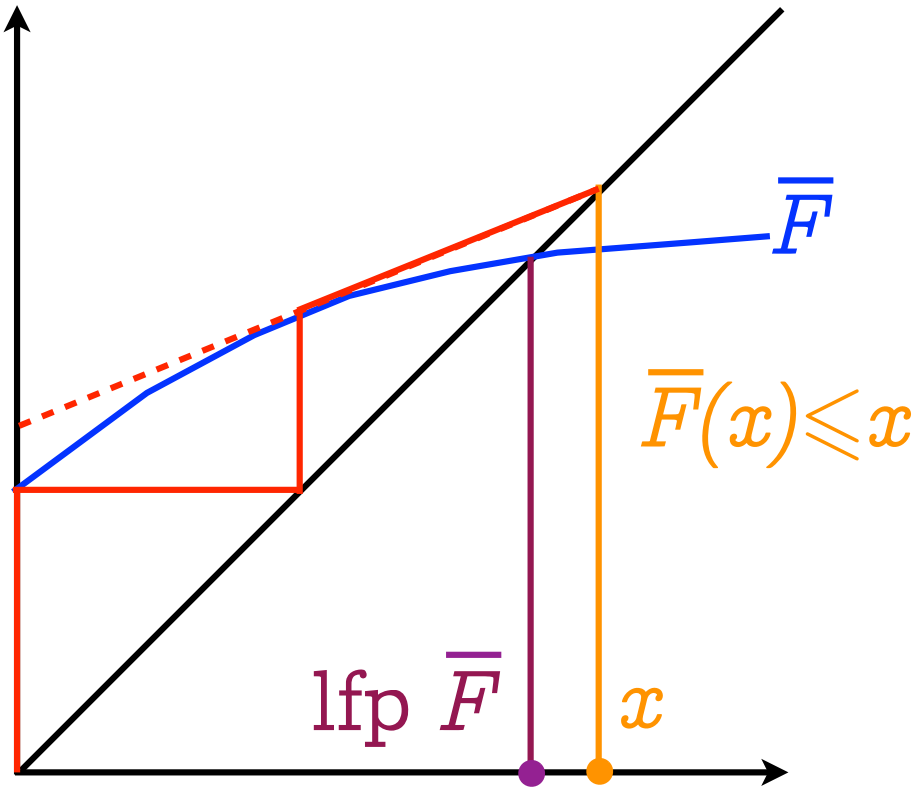


Infinite iteration

Convergence acceleration



Infinite iteration



Accelerated iteration with widening
(e.g. with a widening based on the derivative as in Newton-Raphson method(*))

(*) Javier Esparza, Stefan Kiefer, Michael Luttenberger: Newtonian program analysis. J. ACM 57(6): 33 (2010)

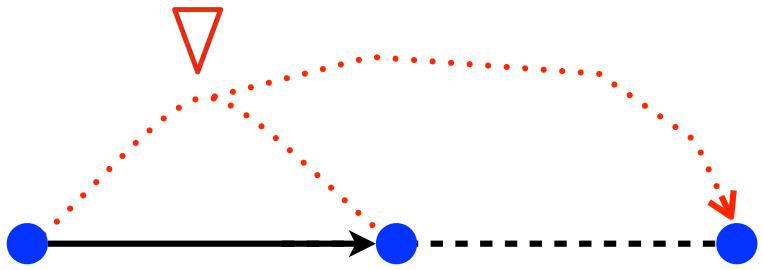
Operators for abstract induction

-

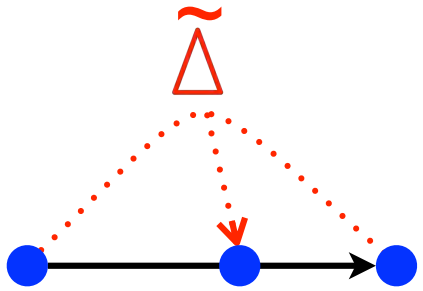
	Convergence above the limit	Convergence below the limit
Increasing iteration	Widening ∇	Dual-narrowing $\widetilde{\Delta}$
Decreasing iteration	Narrowing Δ	Dual widening $\widetilde{\nabla}$

Extrapolators (∇, $\widetilde{\nabla}$) and interpolators (Δ, $\widetilde{\Delta}$)

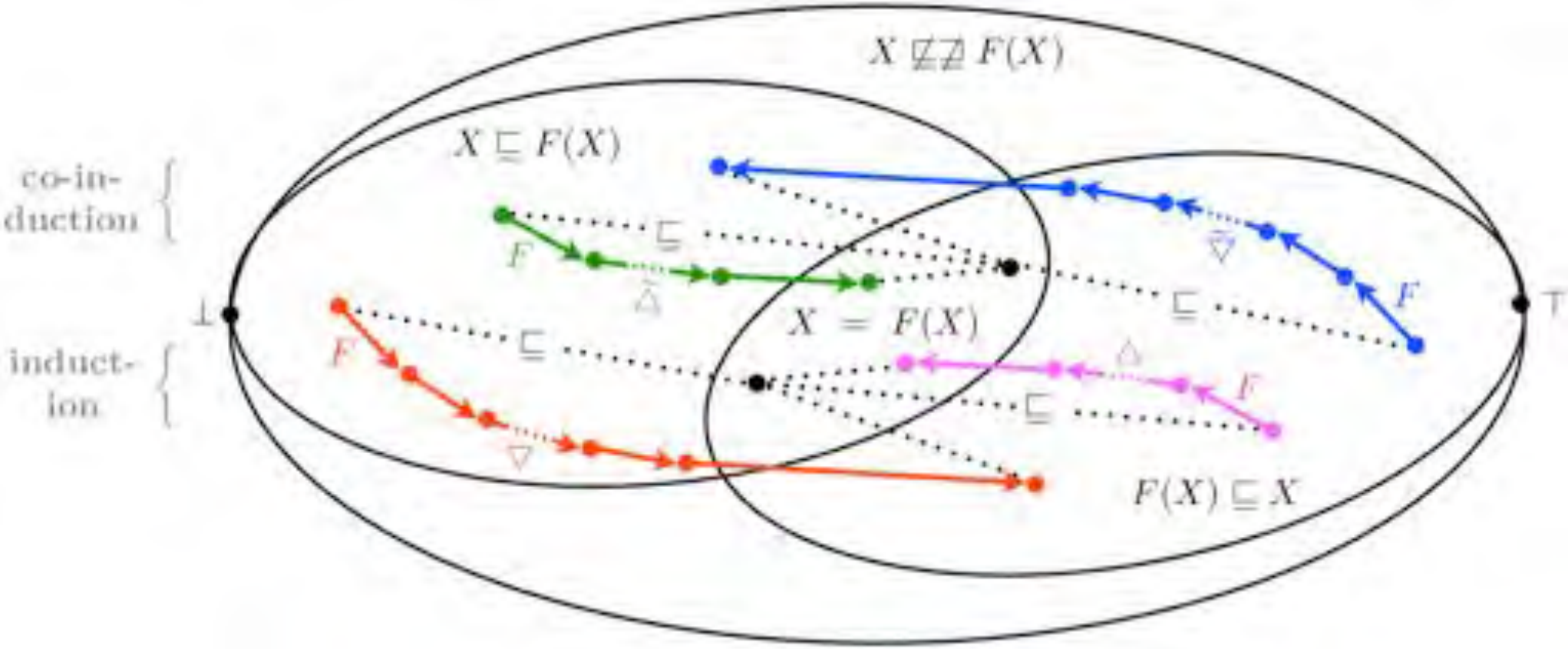
- Extrapolators:



- Interpolators:



Extrapolators and interpolators



Abstract induction versus convergence acceleration

- Abstract induction approximate the iterates
- Convergence acceleration enforces termination
- Two separate issues (*)

(*) Cousot, P.: Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique de programmes. Thèse d'État ès sciences mathématiques, Université Joseph Fourier, Grenoble, France (21 Mar. 1978)

Finite versus infinite abstractions

[In]finite abstractions

- Given a program $\mathtt{P}$ and a program property S which holds (*i*.e. lfp $F[\![\mathtt{P}]\!] \subseteq S$) there exists a most abstract abstraction in a finite domain $\mathscr{A}[\![\mathtt{P}]\!]$ to prove it (*)

- Example:

`x=0; while x<1 do x++` $\longrightarrow$ {⊥, [0,0], [0,1],[-∞,∞]}

`x=0; while x<2 do x++` $\longrightarrow$ {⊥, [0,0], [0,1], [0,2],[-∞,∞]}

…

`x=0; while x<`n `do x++` $\longrightarrow$ {⊥, [0,0], [0,1], [0,2], [0,3], …, [0,n],[-∞,∞]}

…

(*) Patrick Cousot: Partial Completeness of Abstract Fixpoint Checking. SARA 2000: 1-25

[In]finite abstractions

- No such domain exists for infinitely many programs
  - $\bigcup_{\mathtt{P} \in \mathbb{L}} \mathscr{A}[\![\mathtt{P}]\!]$ is infinite

    Example: $\{\bot, [0,0], [0,1], [0,2], [0,3], \ldots, [0,n], [0,n+1], \ldots, [-\infty,\infty]\}$

  - $\lambda \mathtt{P} \in \mathbb{L} . \mathscr{A}[\![\mathtt{P}]\!]$ is not computable (for undecidable properties)

$\Longrightarrow$ finite abstractions will fail infinitely often while infinite abstractions will succeed!

Terminating widenings are not monotone

Iteration with widening

- Iterates $\langle \overline{X}^k, k \in \mathbb{N} \rangle$ of $\overline{F}$ extensive on $\langle \overline{\mathscr{D}}, \overline{\sqsubseteq} \rangle$ with terminating widening $\overline{\nabla} \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$:
  - $\overline{X}^0 \triangleq \overline{D}$ where $\overline{D} \in \overline{\mathscr{D}}$ initial approximation
  - $\overline{X}^{k+1} \triangleq \overline{X}^k \,\overline{\nabla}\, \overline{F}(\overline{X}^k), k \in \mathbb{N}$

- The iterates are increasing ($\overline{F}$ extensive) and converge in finitely many steps ($\overline{\nabla}$ terminating)

Example of widenings

- Primitive widening [1,2]

```
(x ∇̄ y) = cas x ∈ V_a, y ∈ V_a dans
    | ⊥, ? => y ;
    | ?, ⊥ => x ;
    | [n1,m1],[n2,m2] =>
        [si n2 < n1 alors -∞ sinon n1 fsi ;
          si m2 > m1 alors +∞ sinon m1 fsi] ;
    fincas ;
```

$$[a_1, b_1] \bar{\nabla} [a_2, b_2] =$$
$$[\underline{\text{if}}\ a_2 < a_1\ \underline{\text{then}}\ -\infty\ \underline{\text{else}}\ a_1\ \underline{\text{fi}},$$
$$\underline{\text{if}}\ b_2 > b_1\ \underline{\text{then}}\ +\infty\ \underline{\text{else}}\ b_1\ \underline{\text{fi}}]$$

- Widening with thresholds [3]

$$\forall x \in \bar{L}_2, \bot\, \nabla_2(j)\, x = x\, \nabla_2(j)\, \bot = x$$
$$[l_1, u_1]\, \nabla_2(j)\, [l_2, u_2]$$
$$= [\textit{if}\ 0 \le l_2 < l_1\ \textit{then}\ 0\ \textit{elsif}\ l_2 < l_1\ \textit{then}\ -b - 1\ \textit{else}\ l_1\ \textit{fi},$$
$$\textit{if}\ u_1 < u_2 \le 0\ \textit{then}\ 0\ \textit{elsif}\ u_1 < u_2\ \textit{then}\ b\ \textit{else}\ u_1\ \textit{fi}]$$

[1] Patrick Cousot, Radhia Cousot: Vérification statique de la cohérence dynamique des programmes, Rapport du contrat IRIA-SESORI No 75-032, 23 septembre 1975.
[2] Patrick Cousot, Radhia Cousot: Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. POPL 1977: 238-252
[3] Patrick Cousot, Semantic foundations of program analysis, Ch. 10 of Program flow analysis: theory and practice, N. Jones & S. Muchnich (eds), Prentice Hall, 1981.

Example of widenings (cont'd)

- Parameterized widenings:

$$\nabla(\delta, T(\delta), \overline{F}, \langle \overline{X}^{\beta},\ \beta \leqslant \delta \rangle, \langle \overline{F}(\overline{X}^{\beta}),\ \beta \leqslant \delta \rangle)$$

- Example: bounded model-checking

An abrupt such example is n-bounded abstract model checking [3] using an iteration $\overline{X}^{k+1} \triangleq \overline{X}^{k} \nabla_{(k)} \overline{F}(\overline{X}^{k})$ with parameterized widening $\overline{X} \nabla_{(k)} \overline{Y} \triangleq \llbracket k \leqslant n \ ? \ \overline{Y} \ : \ \overline{\top} \rrbracket$ where $\overline{\top}$ is the abstract supremum: $\forall X \in \mathscr{D} : P \subseteq \gamma(\overline{\top})$ (everything is unknown beyond n iterations).

Required properties of terminating widenings

- Widening over-approximates the iterates:

$$(\overline{\nabla}.\text{a}) \quad \forall \overline{X}, \overline{Y} \in \overline{\mathscr{D}} : \overline{Y} \sqsubseteq \overline{X} \,\overline{\nabla}\, \overline{Y}.$$

- Iteration converges when solution found:

$$(\overline{\nabla}.\text{b}) \quad \forall \overline{X}, \overline{Y} \in \overline{\mathscr{D}} : (\overline{Y} \sqsubseteq \overline{X}) \implies (\overline{X} \,\overline{\nabla}\, \overline{Y} = \overline{X}).$$

- Convergence is enforced by *terminating* widening:

$(\overline{\nabla}.\text{c})$

$\overline{\nabla}$ is *terminating* that is for any increasing chain $\langle \overline{X}^k \in \overline{\mathscr{D}}, k \in \mathbb{N} \rangle$ and arbitrary sequence $\langle \overline{Y}^k \in \overline{\mathscr{D}}, k \in \mathbb{N} \rangle$ such that $\forall k \in \mathbb{N} : \overline{X}^k \sqsubseteq \overline{Y}^k$, the sequence $\langle \overline{X}^k \overline{\nabla} \overline{Y}^k, k \in \mathbb{N} \rangle$ is ultimately stationary (*i.e.* $\exists n \in \mathbb{N} : \forall k \geqslant n : \overline{X}^k \,\overline{\nabla}\, \overline{Y}^k = \overline{X}^n$).

Terminating [dual-]widenings are not monotone

Theorem 5 (Non-monotonicity of terminating [dual] widening). *Let $\langle \overline{\mathscr{D}}, \sqsubseteq \rangle$ be a poset and $\nabla \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$ be a widening satisfying* (∇.a), (∇.b), *and* (∇.c). *Then ∇ cannot be increasing in its first parameter. The dual holds for $\widetilde{\nabla}$.*

Proof. By reflexivity, $\overline{Y} \sqsubseteq \overline{Y}$ so (∇.b) implies $\overline{Y} \nabla \overline{Y} = \overline{Y}$. By reductio ad absurdum, if ∇ is increasing in its first parameter then $\overline{X} \sqsubseteq \overline{Y}$ implies $\overline{X} \nabla \overline{Y} \sqsubseteq \overline{Y} \nabla \overline{Y} = \overline{Y} \sqsubseteq \overline{X} \nabla \overline{Y}$ by (∇.a) which implies that $\overline{X} \nabla \overline{Y} = \overline{Y}$ by antisymmetry. By (∇.c), $\forall k \geqslant n$, $\overline{X}^{n+k} = \overline{X}^k \nabla \overline{Y}^k = \overline{X}^k = \overline{X}^n$. By hypothesis $\overline{X}^k \sqsubseteq \overline{Y}^k$ so $\overline{X}^k \nabla \overline{Y}^k = \overline{Y}^k$ which implies $\forall k \geqslant n : \overline{Y}^k = \overline{X}^n$, in contradiction with the fact that $\langle \overline{Y}^k, k \in \mathbb{N} \rangle$ is an arbitrary sequence of elements of $\overline{\mathscr{D}}$, hence in general not ultimately stationary. □

Counter-example 5 (Top widening). The top widening $X \nabla_\top Y \triangleq \top$ is terminating and increasing in its first parameter but does not satisfy condition (∇.b) since if a solution is found by iteration with widening, the top widening will degrade it to $\top$. □

Consequences

- The transformers of structural static analyzers (which contain widenings) are not increasing
- Example

```
while (TRUE) {if (x == 0) {x = 1} else {x = 2}}
```

$$\overline{F}_{\texttt{while}}(I) = \mathbf{lfp}^{\sqsubseteq}\ \boldsymbol{\lambda}\, X \cdot I \sqcup \overline{F}_{\texttt{if}}(X)$$

$$\overline{F}_{\texttt{if}}(X) = (\!|\, 0 \in X \ ?\ [1,1]\ :\ \emptyset \,|\!) \sqcup (\!|\, \exists x \in X : x \neq 0 \ ?\ [2,2]\ :\ \emptyset \,|\!)$$

$$\overline{F}_{\texttt{while}}([0,0]) = [0,+\infty]$$

$$\overline{F}_{\texttt{while}}([0,2]) = [0,2]$$

- Consequence: abstract fixpoints may not exist !

Solution

- The concrete (so-called collecting) semantics is for increasing concrete transformers in posets
- The abstract transformers may not be increasing (no fixpoints)
- Rely on abstract iterations (not abstract fixpoints)
- Rely on appoximations (not existing lubs/glbs)
- As a separate issue, enforce convergence (approximation of infinite iterations)

Concrete iteration

Structure of the domain of properties

- The domain of concrete properties $\langle \mathscr{D}, \subseteq \rangle$ is assumed to be a poset
- No need for CPOs, complete lattices, *etc* since we are only interested in the chains of iterates of a concrete transformer $F \in \mathscr{D} \mapsto \mathscr{D}$
- Even iterates of (an abstract) $F \in \mathscr{D} \mapsto \mathscr{D}$ may have no lubs/glbs
- Abstract/concrete is relative so we need to make weak hypotheses on the concrete.

Chains

Lemma 1 (Increasing sequences in posets are ultimately stationary). *Any $\preceq$-increasing transfinite sequence $\langle X^\delta, \delta \in \mathbb{O}\rangle$ of elements of a poset $\langle \mathscr{P}, \preceq\rangle$ is ultimately stationary* (i.e. $\exists \epsilon \in \mathbb{O} : \forall \delta \geqslant \epsilon : X^\delta = X^\epsilon$. *The smallest such ϵ is the rank of the sequence.*). □

Iterates

Definition 2 (Least/upper bounded iterates). *Let* $F \in \mathscr{D} \mapsto \mathscr{D}$ *be an transformer on a poset* $\langle \mathscr{D}, \subseteq \rangle$ *and* $D \in \mathscr{D}$. *By* least/upper bounded iterates of F from D *we mean a transfinite sequence* $\langle X^\delta, \delta \in \mathbb{O} \rangle$ *of elements of* $\mathscr{D}$ *such that* $X^0 \triangleq D$, $X^{\delta+1} \triangleq F(X^\delta)$, *and for limit ordinals* λ, $\forall \delta < \lambda : X^\delta \subseteq X^\lambda$ *for upper bounded iterates and* X^λ *is the least element with that property for least bounded iterates* $(\forall \delta < \lambda : X^\delta \subseteq X^\lambda \wedge \forall Y : \forall \delta < \lambda : X^\delta \subseteq Y \implies X^\lambda \subseteq Y)$. □

Limits

Lemma 3 (Increasing fixpoint iterates). *Let* $\langle X^\delta, \delta \in \mathbb{O}\rangle$ *be the iterates of an transformer* $F \in \mathscr{D} \mapsto \mathscr{D}$ *on a poset* $\langle \mathscr{D}, \subseteq\rangle$ *from* $D \in \mathscr{D}$.

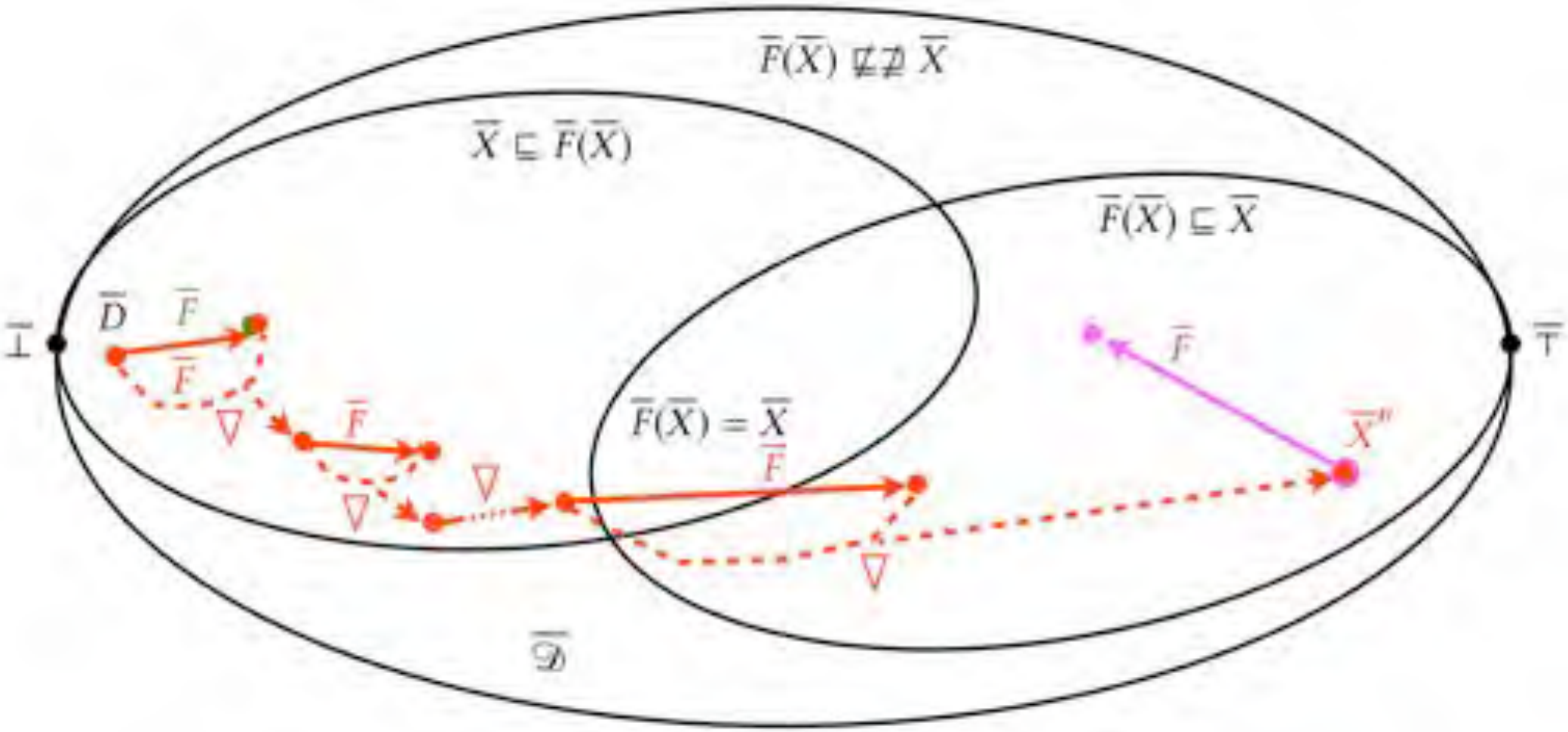
(a) *If* F *is extensive* (i.e. $\forall X \in \mathscr{D} : X \subseteq F(X)$) *and the iterates are upper bounded then they are increasing and* F *has a fixpoint* $\subseteq$*-greater than of equal to* D.

(b) *If* F *is increasing,* D *a prefix-point of* F (i.e. $D \subseteq F(D)$), *and the iterates are upper bounded (resp. least upper bounded) then they are increasing and* F *has a fixpoint* $\subseteq$*-greater than of equal to* D *(resp. least fixpoint* $\boldsymbol{lfp}^{\subseteq}_{D}\ F$*).*

(c) *In case* (b) *of least upper bounded iterates,* $\forall Y \in \mathscr{D} : (D \subseteq Y \wedge F(Y) \subseteq Y) \implies (\boldsymbol{lfp}^{\subseteq}_{D}\ F \subseteq Y)$. □

Increasing iteration with non-monotonic transformer/widening

Increasing iteration with widening



Hypotheses on widenings

Hypotheses 7 (Sound widening for concretization γ).

(a) • for $\nabla \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$, $\quad \forall \overline{P}, \overline{Q} \in \overline{\mathscr{D}} : \gamma(\overline{P}) \subseteq \gamma(\overline{P} \nabla \overline{Q}) \wedge \gamma(\overline{Q}) \sqsubseteq \gamma(\overline{P} \nabla \overline{Q})$

(a′) $\quad \forall \overline{P}, \overline{Q} \in \overline{\mathscr{D}} : \overline{P} \sqsubseteq (\overline{P} \nabla \overline{Q}) \wedge \overline{Q} \sqsubseteq (\overline{P} \nabla \overline{Q})$

(b) • for $\nabla \in \wp(\overline{\mathscr{D}}) \mapsto \overline{\mathscr{D}}$, $\quad \forall \mathfrak{X} \in \wp(\overline{\mathscr{D}}) : \forall \overline{P} \in \mathfrak{X} : \gamma(\overline{P}) \subseteq \gamma(\nabla \mathfrak{X})$

Example of widenings (cont'd)

- Bounded widening: $\forall \overline{P}, \overline{Q} : \overline{P} \nabla_{\overline{S}} \overline{Q} \sqsubseteq \overline{S}$

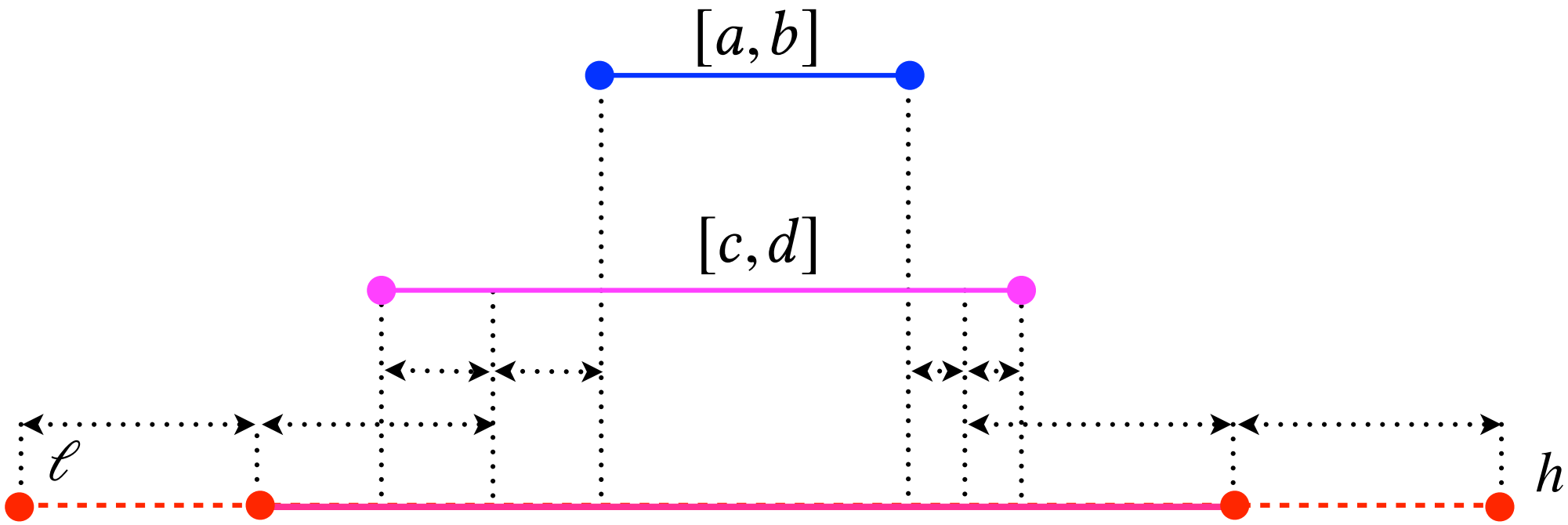
- Ex: bounded widening (in [`min_int`, `max_int`]):

[a, b] ∇ [c, d] $\triangleq$

[if c<a then `min_int` else a, if d>a then `max_int` else b]

Example of widenings (cont'd)

- Bounded widening (in $[\ell, h]$):



$$[a,b] \nabla_{[\ell,h]} [c,d] \triangleq [\frac{c+a-2\ell}{2}, \frac{b+d+2h}{2}]$$

Soundness of widening iterations

Theorem 8 (Over-approximation of increasing abstract iterates by widening). *Let* $\langle X^\delta, \delta \in \mathbb{O}\rangle$ *be the least upper bound iterates of the increasing transformer* $F \in \mathcal{D} \mapsto \mathcal{D}$ *on a concrete poset* $\langle \mathcal{D}, \subseteq\rangle$ *from* $D \in \mathcal{D}$ *such that* $D \subseteq F(D)$. *By* **Lem. 3** (b), $\langle X^\delta, \delta \in \mathbb{O}\rangle$ *is therefore increasing and ultimately stationary at* $X^\epsilon = \mathit{lfp}^{\subseteq}_D F$.

Let the abstract domain $\langle \overline{\mathcal{D}}, \sqsubseteq\rangle$ *be a poset, the concretization* $\gamma \in \overline{\mathcal{D}} \mapsto \mathcal{D}$ *be increasing, the abstract transformer be* $\overline{F} \in \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$, $\nabla \in \overline{\mathcal{D}} \times \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$ *be a widening satisfying* **Hyp. 7** (a) *and* $\nabla \in \wp(\overline{\mathcal{D}}) \mapsto \overline{\mathcal{D}}$ *satisfies* **Hyp. 7** (b) *for all* $\mathcal{X} = \{\overline{X}^\delta \mid \delta < \lambda \wedge \lambda \in \mathbb{O}$ *is a limit ordinal*$\}$, *the abstract iterates be the transfinite sequence* $\langle \overline{X}^\delta \in \overline{\mathcal{D}}, \delta \in \mathbb{O}\rangle$ *such that* $\overline{X}^{\delta+1} \triangleq \overline{X}^\delta \nabla \overline{F}(\overline{X}^\delta)$ *and* $\overline{X}^\lambda \triangleq \nabla_{\beta<\lambda} \overline{X}^\beta$ *for limit ordinals* λ. *Then*

(a) *The concretization* $\langle \gamma(\overline{X}^\delta), \delta \in \mathbb{O}\rangle$ *of the abstract iterates* $\langle \overline{X}^\delta, \delta \in \mathbb{O}\rangle$ *is increasing and ultimately stationary with limit* $\gamma(\overline{X}^\varepsilon)$.

Moreover, if $D \subseteq \gamma(\overline{X}^0)$ *and the semi-commutation condition* $\forall \delta \in \mathbb{O} : F \circ \gamma(\overline{X}^\delta) \subseteq \gamma \circ \overline{F}(\overline{X}^\delta)$ *holds, then*

(b) $\forall \delta \in \mathbb{O} : X^\delta \subseteq \gamma(\overline{X}^\delta)$ *(so, in particular* $X^\epsilon \subseteq \gamma(\overline{X}^\varepsilon)$*);*

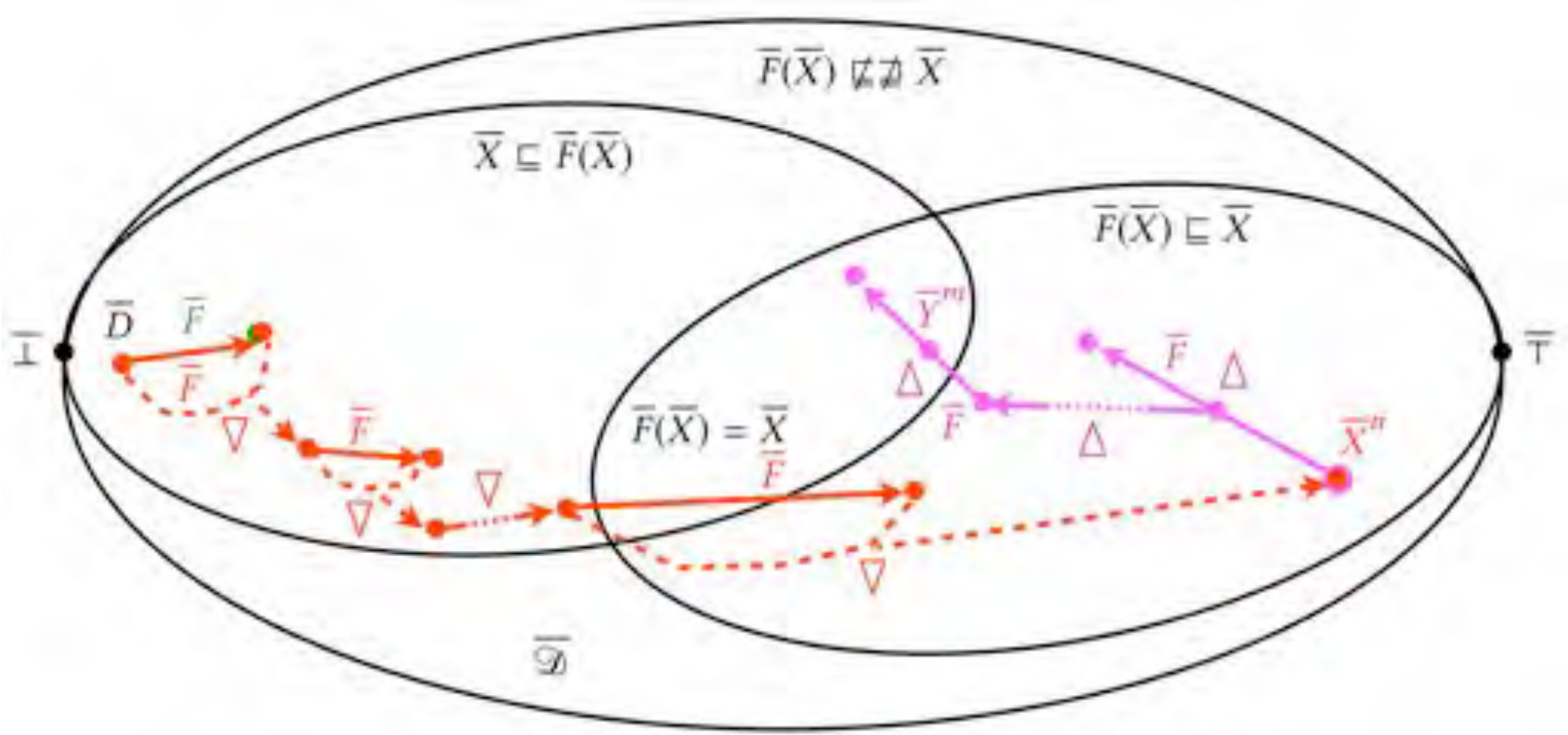
(c) $\forall \delta \in \mathbb{O} : \overline{F}(\overline{X}^\delta) \sqsubseteq \overline{X}^\delta \implies X^\epsilon \subseteq \gamma(\overline{X}^\delta)$.

(d) *Moreover, if* ∇ *satisfies* **Hyp. 7** (a′) *then* $\overline{F}(\overline{X}^\varepsilon) \sqsubseteq \overline{X}^\varepsilon$. □

Condition **Th. 8**.(c) provides a sufficient condition for stopping the abstract iteration.

Decreasing iteration with non-monotonic transformer/narrowing

Decreasing iteration with narrowing



Hypotheses on narrowings

Hypotheses 12 (Sound narrowing for concretization γ).

- for $\Delta \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$,

(a) $\quad \forall \overline{P}, \overline{Q} \in \overline{\mathscr{D}} : (\gamma(\overline{Q}) \subseteq \gamma(\overline{P})) \implies (\gamma(\overline{Q}) \subseteq \gamma(\overline{P} \,\Delta\, \overline{Q}) \subseteq \gamma(\overline{P}))$

(a′) $\quad \forall \overline{P}, \overline{Q} \in \overline{\mathscr{D}} : (\gamma(\overline{Q}) \subseteq \gamma(\overline{P})) \implies (\overline{Q} \sqsubseteq (\overline{P} \,\Delta\, \overline{Q}) \sqsubseteq \overline{P})$

- for $\Delta \in \wp(\overline{\mathscr{D}}) \mapsto \overline{\mathscr{D}}$,

(b) $\quad \forall P \in \mathscr{D} : \forall \mathscr{X} \in \wp(\overline{\mathscr{D}}) : (\forall \overline{Q} \in \mathscr{X} : P \subseteq \gamma(\overline{Q})) \implies (P \subseteq \gamma(\Delta \mathscr{X}) \subseteq \gamma(\overline{Q}))$ □

Example of narrowing

- [2]

$$[a_1,b_1] \bar{\Delta} [a_2,b_2] =$$

$$[\underline{\text{if}}\ a_1 = -\infty\ \underline{\text{then}}\ a_2\ \underline{\text{else}}\ \text{MIN}\ (a_1,a_2),$$

$$\underline{\text{if}}\ b_1 = +\infty\ \underline{\text{then}}\ b_2\ \underline{\text{else}}\ \text{MAX}\ (b_1,b_2)]$$

[2] Patrick Cousot, Radhia Cousot: Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. POPL 1977: 238-252

Soundness of narrowing iterations

Theorem 15 (Over-approximation of decreasing iterates with narrowing). *By the dual of* **Def. 2**, *let* $\langle X^\delta,\ \delta \in \mathbb{O}\rangle$ *be the greatest lower bound iterates of the increasing transformer* $F \in \mathcal{D} \mapsto \mathcal{D}$ *on a concrete poset* $\langle \mathcal{D}, \subseteq\rangle$ *from* $D \in \mathcal{D}$ *such that* $F(D) \subseteq D$. *By the dual of* **Lem. 3** (b), $\langle X^\delta,\ \delta \in \mathbb{O}\rangle$ *is therefore decreasing and ultimately stationary at* $X^\epsilon = \boldsymbol{gfp}^{\subseteq}_D F$.

Let the abstract domain $\langle \overline{\mathcal{D}}, \sqsubseteq\rangle$ *be a poset, the concretization* $\gamma \in \overline{\mathcal{D}} \mapsto \mathcal{D}$ *be increasing, the abstract transformer be* $\overline{F} \in \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$, $\overline{\triangle} \in \overline{\mathcal{D}} \times \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$ *be a narrowing satisfying* **Hyp. 13** (a) (*or* **Hyp. 13** (a′)) *and* $\overline{\triangle} \in \wp(\overline{\mathcal{D}}) \mapsto \overline{\mathcal{D}}$ *satisfies* **Hyp. 13** (b) *for* $\mathfrak{X} = \{\overline{X}^\delta \mid \delta < \lambda \wedge \lambda \in \mathbb{O}$ *is a limit ordinal*$\}$, *where the abstract iterates are the transfinite sequence* $\langle \overline{X}^\delta \in \overline{\mathcal{D}}, \delta \in \mathbb{O}\rangle$ *such that* $D \subseteq \gamma(\overline{X}^0)$, $\overline{X}^{\delta+1} \triangleq \overline{X}^\delta \,\overline{\triangle}\, \overline{F}(\overline{X}^\delta)$, $\overline{X}^\lambda \triangleq \overline{\bigtriangleup}_{\beta<\lambda} \overline{X}^\beta$ *for limit ordinals* λ, *do satisfy the semi-commutation condition* $\forall \delta \in \mathbb{O} : F \circ \gamma(\overline{X}^\delta) \subseteq \gamma \circ \overline{F}(\overline{X}^\delta)$.

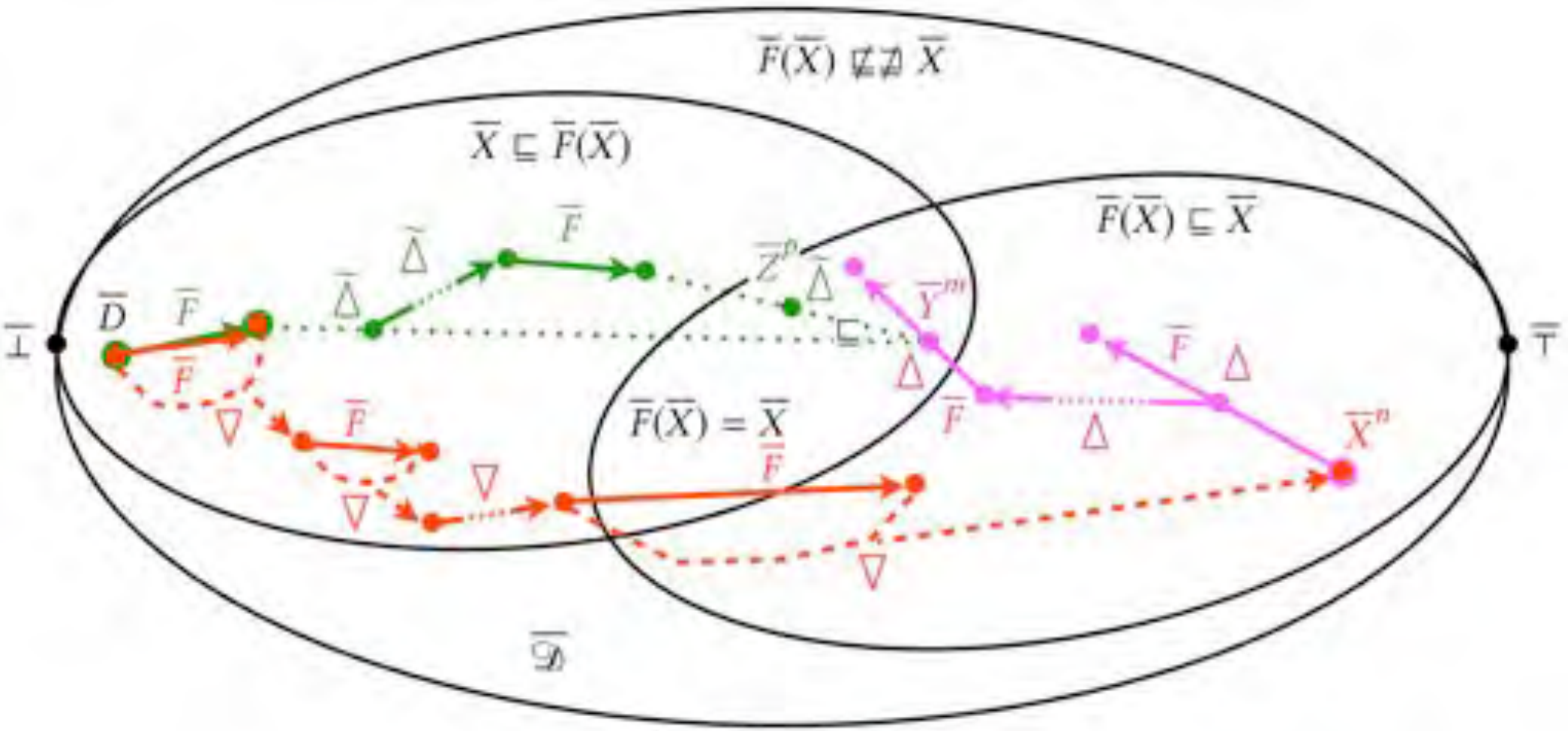
If the abstract transformer $\overline{F} \in \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$ *is reductive on the abstract iterates* $\langle \overline{X}^\delta,\ \delta \in \mathbb{O}\rangle$ (i.e. $\forall \delta \in \mathbb{O} : \overline{F}(\overline{X}^\delta) \sqsubseteq \overline{X}^\delta$) *then their concretization* $\langle \gamma(\overline{X}^\delta),\ \delta \in \mathbb{O}\rangle$ *is decreasing and ultimately stationary with limit* $\gamma(\overline{X}^\varepsilon)$ *such that* $\forall \delta \in \mathbb{O} : \boldsymbol{gfp}^{\subseteq}_D F = X^\epsilon \subseteq \gamma(\overline{X}^\varepsilon) \subseteq \gamma(\overline{X}^\delta)$. □

Decreasing narrowing iterations

Lemma 16. *The more traditional hypothesis that* $(P \sqsubseteq Q) \implies (P \sqsubseteq P \vartriangle Q \sqsubseteq Q)$, $\forall i \in \Delta : (P \sqsubseteq Q_i) \implies (P \sqsubseteq \bigtriangleup_{j \in \Delta} Q_j \sqsubseteq Q_i)$, $\overline{F}(\overline{X}^0) \sqsubseteq \overline{X}^0$, *and* $\overline{F}$ *is increasing imply that* $\overline{F}$ *is reductive on the iterates.* □

Increasing iteration with non-monotonic transformer/dual-widening

Increasing iteration with dual-narrowing



Hypotheses on dual-narrowing

- A narrowing:

$$P \sqsubseteq Q \implies P \sqsubseteq P \Delta Q \sqsubseteq Q$$

- By order-duality, for a dual-narrowing:

$$P \sqsupseteq Q \implies P \sqsupseteq P \tilde{\Delta} Q \sqsupseteq Q$$

$\iff$ renaming

$$Q \sqsupseteq P \implies Q \sqsupseteq Q \tilde{\Delta} P \sqsupseteq P$$

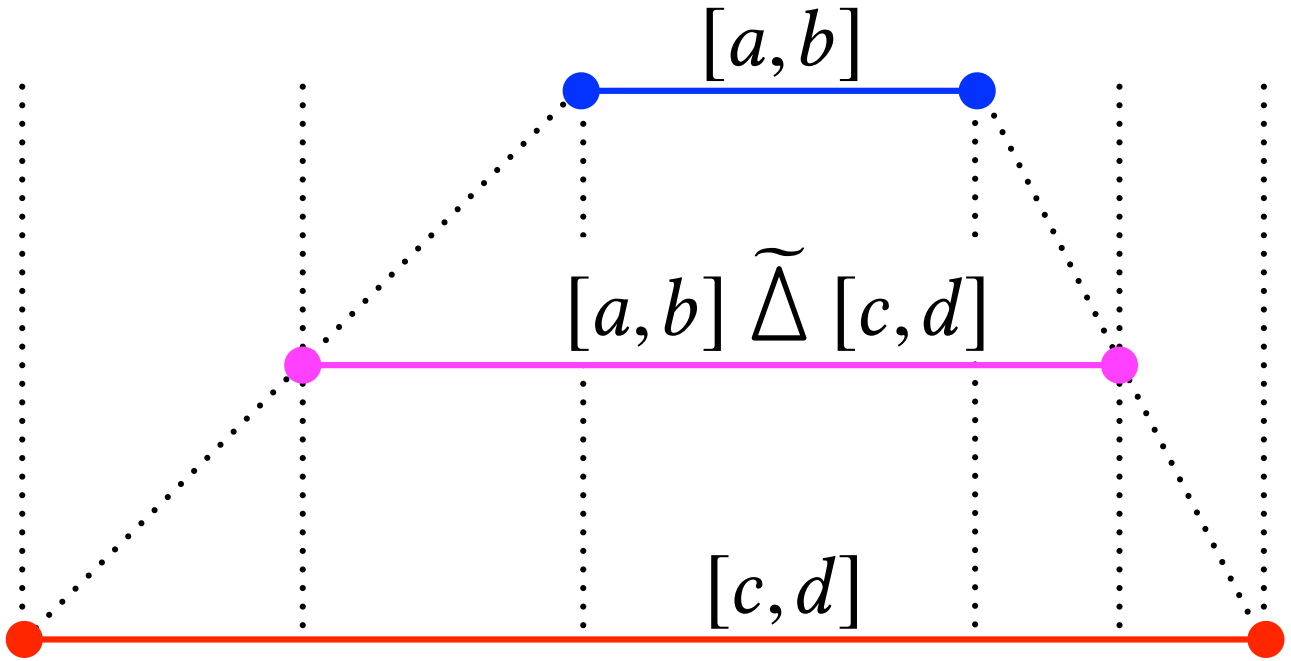
$\iff$ order-duality

$$P \sqsubseteq Q \implies P \sqsubseteq Q \tilde{\Delta} P \sqsubseteq Q$$

- A dual-narrowing is a narrowing with inverted parameters (and inversely)!

Example 1 of dual-narrowing

Example 17 (Interval dual-narrowing). If $[a,b] \subseteq [c,d]$ then $c \leqslant a \leqslant b \leqslant d$ so we can define $[a,b] \mathbin{\widetilde{\Delta}} [c,d] \triangleq [\llbracket c = -\infty \mathbin{?} a \mathbin{:} \lfloor (a+c)/2 \rfloor \rrbracket, \llbracket d = \infty \mathbin{?} b \mathbin{:} \lceil (b+d)/2 \rceil \rrbracket]$ where $\lfloor x \rfloor$ is the largest integer not greater than real x and $\lceil x \rceil$ is the smallest integer not less than real x since $c \leqslant \lfloor (a+c)/2 \rfloor \leqslant a \leqslant b \leqslant \lceil (b+d)/2 \rceil \leqslant d$ and therefore $[a,b] \subseteq ([a,b] \mathbin{\widetilde{\Delta}} [c,d]) \subseteq [c,d]$. □



Example 11 of dual-narrowing

- Craig interpolation:
  - first-order formulae φ and ψ such that $\neg(\varphi \wedge \psi)$
  - $\exists$ first-order formula ρ,
    $\psi \implies \rho$, $\neg(\rho \wedge \psi)$, and $\mathrm{Vars}[\![\rho]\!] \subseteq (\mathrm{Vars}[\![\varphi]\!] \cap \mathrm{Vars}[\![\psi]\!])$.

- Letting $\psi' \triangleq \neg\psi$, if $\varphi \implies \psi'$ then $\exists$ interpolant ρ:
  $\varphi \implies \rho \implies \psi'$.

- So in the poset of first-order formulæ ordered by implication $\implies$, we can define:

$$\varphi \mathbin{\widetilde{\Delta}} \psi \triangleq \rho$$

Iteration with dual-narrowing

Theorem 23 (Over-approximation of bounded increasing iterates with dual-narrowing). *Let $\langle X^\delta, \delta \in \mathbb{O}\rangle$ be the least upper bound iterates of the increasing transformer $F \in \mathcal{D} \mapsto \mathcal{D}$ on a concrete poset $\langle \mathcal{D}, \subseteq\rangle$ from $D \in \mathcal{D}$ such that $D \subseteq F(D)$. By* **Lem. 3** (b), *$\langle X^\delta, \delta \in \mathbb{O}\rangle$ is therefore increasing and ultimately stationary at $X^\epsilon = \mathit{lfp}^{\subseteq}_D F$.*

Let the abstract domain $\langle \overline{\mathcal{D}}, \sqsubseteq\rangle$ be a poset, the concretization $\gamma \in \overline{\mathcal{D}} \mapsto \mathcal{D}$ be increasing, the bound specification $\overline{S} \in \overline{\mathcal{D}}$, the abstract transformer be $\overline{F} \in \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$, $\widetilde{\Delta} \in \overline{\mathcal{D}} \times \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$ be a dual-narrowing $\widetilde{\Delta}$ satisfying the order dual of **Hyp. 13** (a), *and $\widetilde{\Delta} \in \wp(\overline{\mathcal{D}}) \mapsto \overline{\mathcal{D}}$ satisfying the order dual of* **Hyp. 13** (b) *for $\mathfrak{X} \triangleq \{\overline{X}^\lambda \mid \delta < \lambda \wedge \lambda \in \mathbb{O}$ is a limit ordinal$\}$, where the abstract iterates are the transfinite sequence $\langle \overline{X}^\delta \in \overline{\mathcal{D}}, \delta \in \mathbb{O}\rangle$ such that $D \subseteq \gamma(\overline{X}^0)$ and $\overline{X}^0 \sqsubseteq \overline{S}$, $\overline{X}^{\delta+1} \triangleq (\!| \overline{F}(\overline{X}^\delta) \sqsubseteq \overline{S} \;?\; \overline{F}(\overline{X}^\delta) \,\widetilde{\Delta}\, \overline{S} \;:\; \overline{S} |\!)$, $\overline{X}^\lambda \triangleq \widetilde{\Delta}_{\beta<\lambda}\, \overline{X}^\beta$ for limit ordinals λ, which are assumed to satisfy the semi-commutation condition $\forall \delta \in \mathbb{O} : F \circ \gamma(\overline{X}^\delta) \subseteq \gamma \circ \overline{F}(\overline{X}^\delta)$. Then*

(a) *The concretization $\langle \gamma(\overline{X}^\delta), \delta \in \mathbb{O}\rangle$ of the abstract iterates $\langle \overline{X}^\delta, \delta \in \mathbb{O}\rangle$ is such that*

$$\forall \delta \in \mathbb{O} : (X^\delta \subseteq \gamma(\overline{S})) \implies (X^\delta \subseteq \gamma(\overline{X}^\delta) \subseteq \gamma(\overline{S}));$$

(b) $\forall \delta \in \mathbb{O} : \overline{F}(\overline{X}^\delta) \sqsubseteq \overline{X}^\delta \implies \mathit{lfp}^{\subseteq}_D F = X^\delta \subseteq \gamma(\overline{X}^\delta) \subseteq \gamma(\overline{S})$. □

Ternary dual-narrowing

- The iteration with binary dual-narrowing does not exploit all available information ($\overline{X}^{\delta}$):

$$\overline{X}^{\delta+1} \triangleq (\!| \overline{F}(\overline{X}^{\delta}) \sqsubseteq \overline{S} \mathbin{?} \overline{F}(\overline{X}^{\delta}) \mathbin{\widetilde{\Delta}} \overline{S} \mathbin{:} \overline{S} |\!)$$

- We can also use a ternary dual-narrowing:

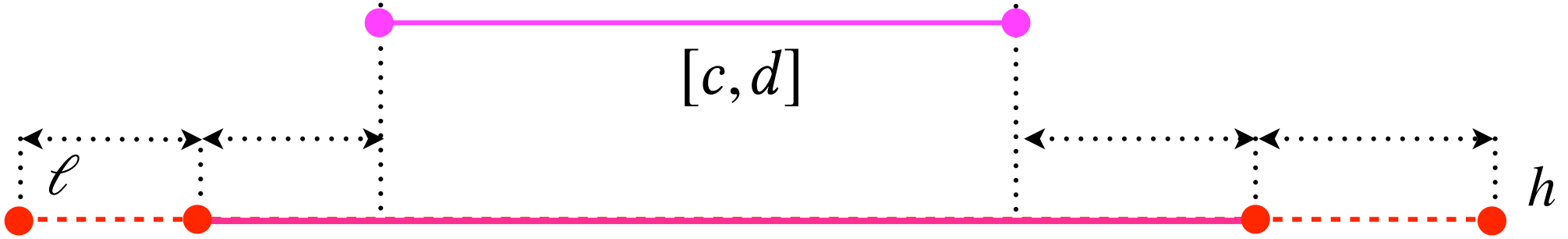
$$\overline{X}^{\delta+1} \triangleq (\!| \overline{F}(\overline{X}^{\delta}) \sqsubseteq \overline{S} \mathbin{?} \widetilde{\Delta}(\overline{X}^{\delta}, \overline{F}(\overline{X}^{\delta}), \overline{S}) \mathbin{:} \overline{S} |\!)$$

- Soundness requirement:

$$\overline{P} \sqsubseteq \overline{Q} \sqsubseteq \overline{S} \text{ implies } \overline{Q} \sqsubseteq \widetilde{\Delta}(\overline{P}, \overline{Q}, \overline{S}) \sqsubseteq \overline{S}.$$

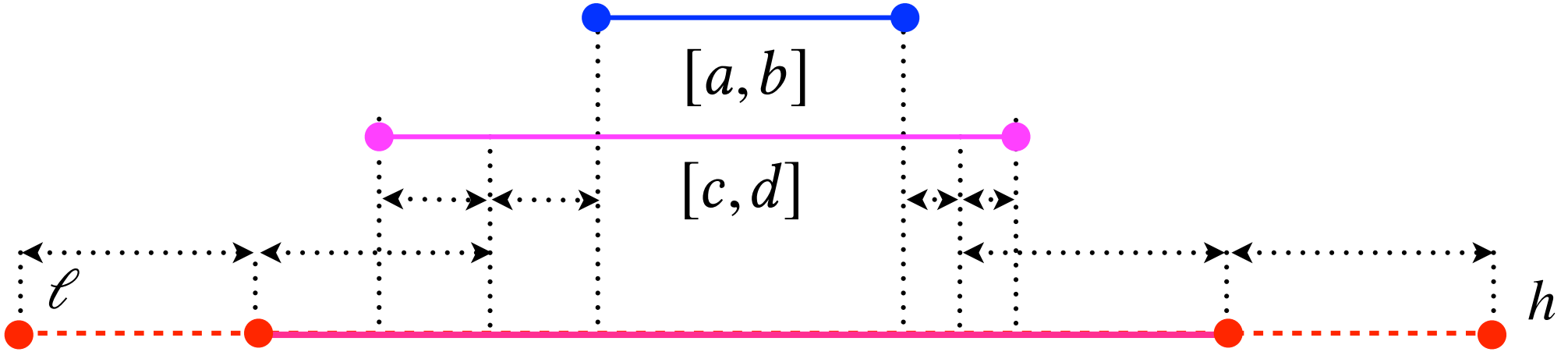
Example of binary/ternary dual-narrowing

- Binary dual-narrowing:



$$\tilde{\Delta}([c,d],[\ell,h]) \triangleq [\frac{a-\ell}{2}, \frac{d+h}{2}]$$

- Ternary dual-narrowing:



$$\tilde{\Delta}([a,b],[c,d],[\ell,h]) \triangleq [\frac{c+a-2\ell}{2}, \frac{b+d+2h}{2}]$$

Ternary dual-narrowing versus bounded widening

- A bounded widening provides a ternary dual-narrowing:

  $\widetilde{\Delta}(\overline{P},\overline{Q},\overline{S}) \triangleq \overline{P} \,\nabla_{\overline{S}}\, \overline{Q}$ is a dual narrowing since if $\overline{P} \sqsubseteq \overline{Q} \sqsubseteq \overline{S}$ then by **Hyp.** 7 (a′), $\overline{Q} \sqsubseteq \overline{P} \nabla_{\overline{S}} \overline{Q}$ and $\overline{P} \nabla_{\overline{S}} \overline{Q} \sqsubseteq \overline{S}$ since the widening is bounded so that $\overline{Q} \sqsubseteq \widetilde{\Delta}(\overline{P},\overline{Q},\overline{S}) \sqsubseteq \overline{S}$.

- Reciprocally, a ternary dual-narrowing provides a bounded widening (for increasing chains only):

  if $\overline{P} \sqsubseteq \overline{Q}$ then $\overline{P} \,\nabla_{\overline{S}}\, \overline{Q} \triangleq \widetilde{\Delta}(\overline{P},\overline{Q},\overline{S})$ is a bounded widening.

Increasing/decreasing operators

- ∇ and dual-narrowing $\widetilde{\Delta}$ are increasing (since they operate on increasing chains $\langle \overline{X}^i, i \in \mathbb{N} \rangle$)

- dual widening $\widetilde{\nabla}$ and narrowing Δ are decreasing (since they operate on decreasing chains $\langle \overline{X}^i, i \in \mathbb{N} \rangle$).

- We consider iterations of the form

$$\overline{X}^0, \ldots, \overline{X}^{i+1} \triangleq \overline{X}^i \boxtimes \overline{Y}^i, \ldots$$

where $\boxtimes \in \{\nabla, \widetilde{\nabla}, \Delta, \widetilde{\Delta}\}$

$\langle \overline{X}^i, i \in \mathbb{N} \rangle$ is a chain

$\langle \overline{Y}^i, i \in \mathbb{N} \rangle$ is arbitrary.

Terminating extrapolators/ interpolators

Termination

Definition 29 (Terminating extrapolation/interpolation operator). *An increasing (resp. decreasing) extrapolation/interpolation operator* $\boxtimes \in \{\nabla, \widetilde{\nabla}, \Delta, \widetilde{\Delta}\}$ *such that* $\boxtimes \in \overline{\mathcal{D}} \times \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$ *is* terminating *whenever for any increasing (resp. decreasing) chain* $\langle \overline{X}^i \in \overline{\mathcal{D}}, i \in \mathbb{N} \rangle$ *and arbitrary sequence* $\langle \overline{Y}^i \in \overline{\mathcal{D}}, i \in \mathbb{N} \rangle$, *the sequence* $\overline{X}^0, \ldots, \overline{X}^{i+1} \triangleq \overline{X}^i \boxtimes \overline{Y}^i, \ldots$ *is ultimately stationary at some rank* $n \in \mathbb{N}$.

Definition 30 (Terminating bounded interpolation operator). *An increasing (resp. decreasing) interpolation operator* $\boxtimes \in \{\Delta, \widetilde{\Delta}\}$ *such that* $\boxtimes \in \overline{\mathcal{D}} \times \overline{\mathcal{D}} \times \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$ *is* terminating *whenever for any increasing (resp. decreasing) chain* $\langle \overline{Y}^i \in \overline{\mathcal{D}}, i \in \mathbb{N} \rangle$ *and bound* $\overline{S} \in \overline{\mathcal{D}}$, *the sequence* $\overline{X}^0 = \overline{Y}^0, \ldots, \overline{X}^{i+1} = \boxtimes(\overline{X}^i, \overline{Y}^i, \overline{S})$[6], $\ldots$ *is ultimately stationary at some rank* $n \in \mathbb{N}$. □

[6] $\overline{X}^{i+1} = \overline{Y}^i \boxtimes \overline{S}$ for binary interpolators $\boxtimes \in \overline{\mathcal{D}} \times \overline{\mathcal{D}} \mapsto \overline{\mathcal{D}}$.

Examples

- The widenings and narrowings given as examples are all terminating
- Dual-narrowing:
  - $[a,b] \mathbin{\widetilde{\Delta}} [c,d] \triangleq [\llparenthesis\, c = -\infty \mathbin{?} a : \lfloor (a+c)/2 \rfloor \,\rrparenthesis, \llparenthesis\, d = \infty \mathbin{?} b : \lceil (b+d)/2 \rceil \,\rrparenthesis]$

    is not bounded terminating when c = -∞ or d = ∞

  - The bounded dual narrowing

    $[a,b] \mathbin{\widetilde{\Delta}} [c,d] \triangleq [\lfloor (a+c)/2 \rfloor, \lceil (b+d)/2 \rceil] \subseteq [\ell, h].$

    is always terminating but convergence is slow

Static analysis

Algorithm

Algorithm 32 (Fixpoint over-approximation by successive extrapolations and interpolations). Input $\overline{F}$ and $\overline{D}$ on $\langle \overline{\mathscr{D}}, \sqsubseteq \rangle$.

(A) Using a terminating widening $\nabla \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$, compute iteratively the iterations $\overline{X}^0 \triangleq \overline{D}, \ldots, \overline{X}^{k+1} \triangleq \overline{X}^k \nabla \overline{F}(\overline{X}^k)$ until reaching a post-fixpoint $\overline{F}(\overline{X}^n) \sqsubseteq \overline{X}^n$ at some rank n.

(B) If $\overline{F}(\overline{X}^n) \neq \overline{X}^n$ then, using a terminating narrowing $\Delta \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$, compute iteratively the iterations $\overline{Y}^0 \triangleq \overline{X}^n, \ldots, \overline{Y}^{k+1} \triangleq \overline{Y}^k \Delta \overline{F}(\overline{Y}^k)$ until reaching $\overline{Y}^{m+1} = \overline{Y}^m$ at some rank m.
Otherwise $\overline{F}(\overline{X}^n) = \overline{X}^n$ so skip this step (B) with $\overline{Y}^m = \overline{X}^n$.

(C) Using a terminating dual-narrowing $\widetilde{\Delta} \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$, compute iteratively the iterations $\overline{Z}^0 \triangleq \overline{D}, \ldots, \overline{Z}^{k+1} \triangleq \overline{F}(\overline{Z}^k) \,\widetilde{\nabla}\, \overline{Y}^m$ until reaching $\overline{Z}^{p+1} = \overline{Z}^p$ at some rank p.

Optionnally, if $\overline{Z}^p \sqsubset \overline{Y}^m$, repeat the interpolation steps (B) and (C) from $\overline{X}'^n = \overline{Z}^p \,\Delta'\, \overline{Y}^m$ (where Δ' is a terminating narrowing satisfying **Hyp. 13** (a′)) until convergence to $\overline{Z}^p \,\Delta'\, \overline{Y}^m = \overline{Y}^m$ [7]. Return $\overline{Z}^p$. □

Soundness

Theorem 33 (Soundness and termination of Alg. 32). *Let* $\langle \mathscr{D}, \subseteq, \cup \rangle$ *be a poset,* $F \in \mathscr{D} \mapsto \mathscr{D}$ *be increasing,* $D \in \mathscr{D}$ *be such that* $D \subseteq F(D)$*, and the concrete iterates* $X^0 \triangleq D$*,* $X^{\delta+1} \triangleq F(X^\delta)$ *for successor ordinals, and* $X^\lambda \triangleq \bigcup_{\beta<\lambda} X^\beta$ *for limit ordinals* λ*, be well defined in the poset* $\langle \mathscr{D}, \subseteq, \cup \rangle$ (i.e. *the lubs* $\bigcup$ *do exist*).

Let the abstract domain $\langle \overline{\mathscr{D}}, \sqsubseteq \rangle$ *be a poset, the concretization* $\gamma \in \overline{\mathscr{D}} \mapsto \mathscr{D}$ *be increasing, the abstract transformer be* $\overline{F} \in \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$ *satisfying the pointwise semi-commutation condition* $F \circ \gamma \dot{\subseteq} \gamma \circ \overline{F}$.

Let $\overline{D} \in \overline{\mathscr{D}}$ *be such that* $D \subseteq \gamma(\overline{D})$ *and* $\forall \overline{X} \in \overline{\mathscr{D}} : (\overline{D} \sqsubseteq \overline{X} \wedge \overline{F}(\overline{X}) \sqsubseteq \overline{X}) \implies (\overline{D} \sqsubseteq \overline{F}(\overline{X}))$*,* $\overline{\nabla} \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$ *be a terminating widening satisfying* **Hyp. 8** (a′)*,* $\overline{\Delta} \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$ *be a terminating narrowing satisfying* **Hyp. 14** (a) *such that* $\forall \overline{X} \in \overline{\mathscr{D}} : (\overline{F}(\overline{X}) \sqsubseteq \overline{X}) \implies (\overline{F}(\overline{X} \mathbin{\overline{\Delta}} \overline{F}(\overline{X})) \sqsubseteq \overline{X} \mathbin{\overline{\Delta}} \overline{F}(\overline{X}))$*, and* $\widetilde{\overline{\Delta}} \in \overline{\mathscr{D}} \times \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$ *be a terminating dual-narrowing satisfying the order dual of* **Hyp. 14** (a′).

Then static analysis **Alg. 32** *always terminates with a sound fixpoint over-approximation* $\overline{Z}^p$ *such that* $\mathbf{lfp}^{\subseteq}_D F \subseteq \gamma(\overline{Z}^p) \subseteq \gamma(\overline{Y}^m) \subseteq \gamma(\overline{X}^n)$*. We have* $\overline{Z}^p \sqsubseteq \overline{Y}^m \sqsubseteq \overline{X}^n$ *so the fixpoint over-approximation* $\overline{Z}^p$ *is improved by the successive interpolations* (B) *and* (C).

Given an abstract specification $\overline{S} \in \overline{\mathscr{D}}$*, if* $\overline{Z}^p \sqsubseteq \overline{S}$ *then* $\mathbf{lfp}^{\subseteq}_D F \subseteq \gamma(\overline{S})$ *else it is unknown whether the specification holds.* □

Verification, checking, analysis

Static verification, analysis & checking

The static inductive proof $\exists \bar{I} \in \overline{\mathscr{D}} : \bar{F}(\bar{I}) \sqsubseteq \bar{I} \wedge \bar{I} \sqsubseteq \bar{S}$ can be done in various forms.

(a) In static verification by deductive verification methods, the induction hypothesis $\bar{I}$ is provided by the end-user so that the problem is to generate and check the verification condition $\bar{F}(\bar{I}) \sqsubseteq \bar{I} \wedge \bar{I} \sqsubseteq \bar{S}$.

(b) In static checking, the induction hypothesis $\bar{I}$ must be automatically inferred from the transformer $\bar{F}$ and the specification $\bar{S}$ (and also checked to satisfy the verification condition $\bar{F}(\bar{I}) \sqsubseteq \bar{I} \wedge \bar{I} \sqsubseteq \bar{S}$).

(c) In static analysis, the induction hypothesis $\bar{I}$ must be automatically inferred from the transformer $\bar{F}$ (independently of a particular specification $\bar{S}$) and checked to satisfy the verification $\bar{F}(\bar{I}) \sqsubseteq \bar{I}$. Then when a specification $\bar{S}$ is given, it remains to check that $\bar{I} \sqsubseteq \bar{S}$.

Checking $\Longrightarrow$ analysis

– Static analysis (c) is static checking (b) where the specification $S = \overline{\top}$ is the always true *i.e.* $\forall \bar{I} : \bar{I} \sqsubseteq \overline{\top}$.

Analysis $\Longrightarrow$ checking

– Static checking (b) is static analysis (c) in the abstract domain $\overline{\mathcal{D}}' \triangleq \{P \in \overline{\mathcal{D}} \mid P \sqsubseteq \overline{S}\}$. The idea is therefore to assume that the specification $\overline{S}$ does hold and to calculate by **Alg. 32** a more precise inductive fixpoint over-approximation $\overline{Z}^p$ in $\overline{\mathcal{D}}'$. Upon termination it remains to check that the fixpoint over-approximation $\overline{Z}^p$ is inductive and stronger than the specification $\overline{S}$ in $\overline{\mathcal{D}}$.

Theorem 35 (Static checking). *Assume the hypotheses of* **Th. 33** *and let* $\overline{S} \in \overline{\mathcal{D}}$ *be a (non-inductive) abstract specification (such that* $D \subseteq \gamma(\overline{S})$ *and* $\overline{F}(\overline{S}) \not\sqsubseteq \overline{S}$[9]*). Let* $\overline{Z}'^p$ *be the result of* **Alg. 32** *applied to the restriction* $\overline{F}'(\overline{X}) \triangleq (\!|\, \overline{F}(\overline{X}) \sqsubseteq \overline{S} \;?\; \overline{F}(\overline{X}) \;:\; \overline{S} \,|\!)$ *of* $\overline{F}$ *to* $\overline{\mathcal{D}}' \triangleq \{P \in \overline{\mathcal{D}} \mid P \sqsubseteq \overline{S}\}$ *and* $\overline{D} \in \overline{\mathcal{D}}'$*, using the bounded widening* $\overline{X} \mathbin{\overline{\nabla}'} \overline{Y} \triangleq (\!|\, \overline{X} \mathbin{\overline{\nabla}} \overline{Y} \sqsubseteq \overline{S} \;?\; \overline{X} \mathbin{\overline{\nabla}} \overline{Y} \;:\; \overline{S} \,|\!)$ *which is the restriction of the widening* $\overline{\nabla}$ *to* $\overline{\mathcal{D}}'$*, and same narrowing satisfying* **Hyp. 14** (a′) *and same dual-narrowing satisfying the dual of* **Hyp. 14** (a′).

If $\overline{F}(\overline{Z}'^p) \sqsubseteq \overline{Z}'^p$ *then* $\boldsymbol{lfp}^{\subseteq}_{D}\, F \subseteq \gamma(\overline{S})$ *that is .* □

On specifications

On Specifications in the small

- In the small/tiny, specifications may be ``not far from the inductive argument"
  - Observed by

Morris Jr., J.H., Wegbreit, B.: Subgoal induction. Commun. ACM 20(4), 209–222 (1977)
Dijkstra, E.W.: Heuristics for a calculational proof. Inf. Process. Lett. 53(3), 141–143 (1995)
Dijkstra, E.W., Scholten, C.S.: Predicate calculus and program semantics. Texts and monographs in computer science, Springer (1990)

  - Exploited by

Dillig, I., Dillig, T., Li, B., McMillan, K.L.: Inductive invariant generation via abductive inference. In: OOPSLA. 443–456. ACM (2013)
McMillan, K.L.: Applications of Craig interpolation to model checking. In: CSL. LNCS 3210, 22–23. Springer (2004)
McMillan, K.L.: Applications of Craig interpolants in model checking. In: TACAS. LNCS 3440, 1–12. Springer (2005)

On Specifications in the large

- No specifications or specifications are ``far from'' the inductive argument

An example in ASTRÉE is the problem of finding maximal `l` and minimal `h` bounds such that `S[0], S[1] ∈ [l, h]` is invariant in the following filter program

```
typedef enum {FALSE = 0, TRUE = 1} BOOLEAN; BOOLEAN INIT; float P, X;
void filter () { static float E[2], S[2];
 if (INIT) {S[0] = X; P = X; E[0] = X;}
 else { P = (((((0.5*X)-(E[0]*0.7))+(E[1]*0.4))+(S[0]*1.5))-(S[1]*0.7));}
 E[1] = E[0]; E[0] = X; S[1] = S[0]; S[0] = P;
 /* S[0], S[1] in [l, h] */ }
void main () { X = 0.2*X+5; INIT = TRUE;
 while (1) { X = 0.9*X+35; /* simulated filer input */
  filter (); INIT = FALSE; } }
```

= -h.

On "Widening and interpolation" [44]

[44] McMillan, K.L.: Widening and interpolation. In: SAS. LNCS 6887, p. 1. Springer (2011), slides sas2011.cs.technion.ac.il/slides/mcmillan.pptx

Conclusion

- Widening/narrowing and interpolation are methods of generalizing from bounded to unbounded proofs
- Formally, widening/narrowing satisfies stronger conditions

widening/narrowing

soundness
expanding/contracting
stabilizing

interpolation

soundness

stabilization is not obtained when proving properties, however

Conclusion, cont.

- Heuristically, the difference is weak v. strong bias

strong bias

restricted proof system
incompleteness
smaller search space
domain knowledge
efficient representations

weak bias

rich proof system
completeness
large search space
Occam's razor
generic representations

- Can we combine strong and weak heuristics?
  - Fall back on weak heuristics when strong fails
  - Use weak heuristics to handle combinatorial complexity
  - Build known widenings into theory solvers in SMT?

[44][8] discusses widening (extrapolation) versus interpolation (narrowing/dual-narrowing).

- It is argued that extrapolation uses a weak/inexpressive abstract domain with efficient representations and small search space while interpolation uses a strong/expressive abstract domain with generic representations and large search space. In fact both approaches rely on an abstract domain and this choice is independent of the chosen iteration method. For example [24] shows that combinations of theories in SMT solvers are reduced products of abstract domains (just lacking extrapolation and interpolation operators). Some theories in SMT solvers rely on specific internal representations for efficiency (like affine inequalities).
- The fact that one reason on (relational) invariants or sets of computation histories is part of the choice of the abstract domain. For example trace-based abstraction [17,6] and trace partitioning [47] can lift any abstraction to reason by case analysis on computation histories.
- The transformers F (and $\overline{F}$) can be weakest pre- or strongest post-conditions (and their abstraction). The fact that the equivalence formalized in the concrete by the Galois connection $\langle \mathscr{D}, \subseteq \rangle \underset{\mathrm{post}[\tau]}{\overset{\widetilde{\mathrm{pre}}[\tau]}{\leftrightarrows}} \langle \mathscr{D}, \subseteq \rangle$ is preserved in the abstract depends on the abstract domain not on the convergence acceleration method (widening, narrowing, and duals).

[8] See also the slides `sas2011.cs.technion.ac.il/slides/mcmillan.pptx`.

– Incompleteness comes from the choice of the abstract domain and the extrapolation/interpolation operators. The abstract domain is fundamentally incomplete by undecidability (*e.g.* first-order predicates are incomplete for Hoare logic [13]). Extrapolation itself is not necessarily non-terminating and incomplete. A counter-example is abstract acceleration where the abstract fixpoint can be computed exactly from the first iterates [38].
– Ockham's razor (*lex parsimoniae*) can be made part of the definition of the abstract transformer and the extrapolation/interpolation operators. As pointed out in [20], it is always possible to introduce simplification heuristics *e.g.* by using $\boldsymbol{\lambda}\, X \cdot X \mathbin{\overline{\nabla}} \overline{F}(X)$ or it's n-unrolling version $\boldsymbol{\lambda}\, X \cdot (\ldots((X \mathbin{\overline{\nabla}} \overline{F}(X)) \mathbin{\overline{\nabla}} \overline{F}^2(X)) \ldots \mathbin{\overline{\nabla}} \overline{F}^n(X))$ where the local widening $\overline{\nabla}$ performs heuristic simplifications or to approximate the transformer based on interpolation *e.g.* by using $\boldsymbol{\lambda}\, X \cdot \overline{F}(X) \mathbin{\widetilde{\Delta}} \overline{S}$ as proposed in [43]. Notice that the main contribution to get a simplified transformer $\overline{F} \in \overline{\mathscr{D}} \mapsto \overline{\mathscr{D}}$ is through the careful design of the abstract domain $\overline{\mathscr{D}}$ (and that can always be done through a widening [18]).

[6] Colby, C., Lee, P.: Trace-based program analysis. In: POPL. 195–207. ACM (1996)

[13] Cousot, P.: Methods and logics for proving programs. In: Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B), pp. 841–994. Elsevier (North-Holland) (1990)

[17] Cousot, P., Cousot, R.: Systematic design of program analysis frameworks. In: POPL. 269–282. ACM (1979)

[18] Cousot, P., Cousot, R.: Comparing the Galois connection and widening/narrowing approaches to abstract interpretation. In: PLILP. LNCS 631, pp. 269–295. Springer (1992)

[20] Cousot, P., Cousot, R.: Formal language, grammar and set-constraint-based program analysis by abstract interpretation. In: FPCA. 170–181. ACM (1995)

[24] Cousot, P., Cousot, R., Mauborgne, L.: Theories, solvers and static analysis by abstract interpretation. J. ACM 59(6), 31 (2012)

[38] Jeannet, B., Schrammel, P., Sankaranarayanan, S.: Abstract acceleration of general linear loops. In: POPL. 529–540. ACM (2014)

[42] McMillan, K.L.: Applications of Craig interpolation to model checking. In: CSL. LNCS 3210, 22–23. Springer (2004)

[43] McMillan, K.L.: Applications of Craig interpolants in model checking. In: TACAS. LNCS 3440, 1–12. Springer (2005)

[47] Rival, X., Mauborgne, L.: The trace partitioning abstract domain. ACM Trans. Program. Lang. Syst. 29(5) (2007)

On refinement

Refinement: good news

- Problem: how to prove a valid abstract property $\mathrm{lfp}\, F[\![\mathtt{P}]\!] \subseteq \gamma(\overline{S})$ when $F \circ \gamma \sqsubseteq \gamma \circ \overline{F}$ but $\mathrm{lfp}\, \overline{F}[\![\mathtt{P}]\!] \not\sqsubseteq \overline{S}$?

- It is always possible to refine $\langle \overline{\mathscr{D}}, \sqsubseteq \rangle$ into a most abstract more precise abstraction $\langle \overline{\mathscr{D}}', \sqsubseteq' \rangle$ such that $\langle \overline{\mathscr{D}}', \sqsubseteq' \rangle \underset{\alpha'}{\overset{\gamma'}{\rightleftarrows}} \langle \mathscr{D}, \subseteq \rangle$ and $F' \circ \gamma \sqsubseteq \gamma \circ F'$ such that $\mathrm{lfp}\, F'[\![\mathtt{P}]\!] \sqsubseteq' \alpha' \circ \gamma\,(\overline{S}\,)$

  (thus proving $\mathrm{lfp}\, F[\![\mathtt{P}]\!] \subseteq \gamma'(\overline{S})$ which implies $\mathrm{lfp}\, F[\![\mathtt{P}]\!] \subseteq \gamma(\overline{S})$)

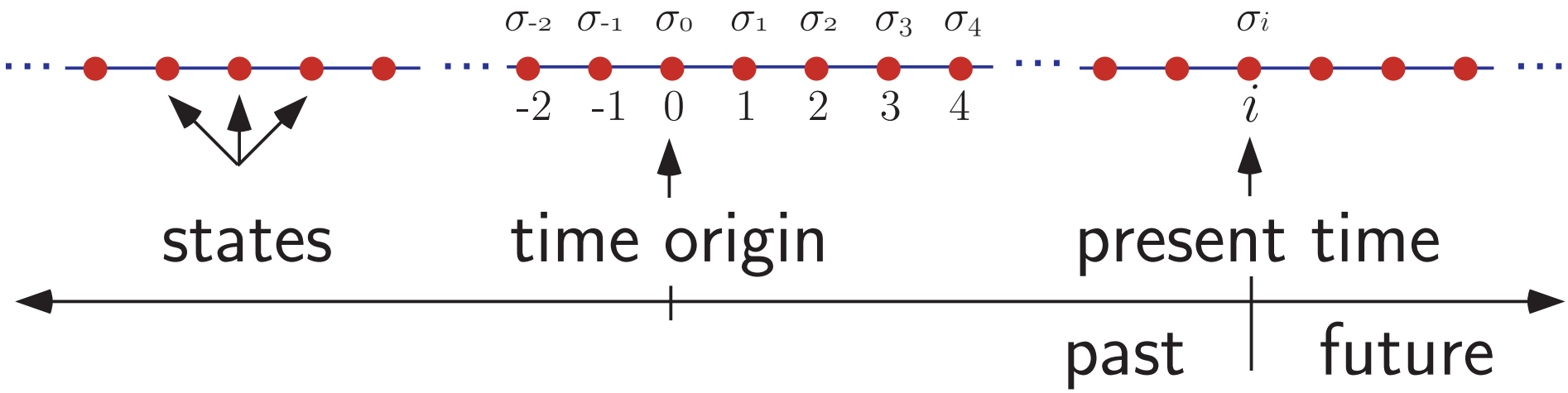
Roberto Giacobazzi, Francesco Ranzato, Francesca Scozzari: Making abstract interpretations complete. J. ACM 47(2): 361-416 (2000)

Refinement: bad news

- But, refinements of an abstraction can be intrinsically incomplete
- The only complete refinement of that abstraction for the collecting semantics is :
  - the identity (i.e. no abstraction at all)
- In that case, the only complete refinement of the abstraction is to the collecting semantics and any other refinement is always imprecise

Example of intrinsic approximate refinement

- Consider executions traces $\langle i, \sigma \rangle$ with infinite past and future:



Patrick Cousot, Radhia Cousot: Temporal Abstract Interpretation. POPL 2000: 12-25

Example of intrinsic approximate refinement

- Consider the temporal specification language $\mu^\star$ (containing LTL, CTL, CTL*, and Kozen's μ-calculus as fragments):

$\varphi ::=$	$\boldsymbol{\sigma}_S$	$S \in \wp(\mathbb{S})$	state predicate
$\mid$	$\boldsymbol{\pi}_t$	$t \in \wp(\mathbb{S} \times \mathbb{S})$	transition predicate
$\mid$	$\oplus\, \varphi_1$		next
$\mid$	$\varphi_1{}^{\curvearrowleft}$		reversal
$\mid$	$\varphi_1 \vee \varphi_2$		disjunction
$\mid$	$\neg\, \varphi_1$		negation
$\mid$	X	$X \in \mathbb{X}$	variable
$\mid$	$\boldsymbol{\mu}\, X \cdot \varphi_1$		least fixpoint
$\mid$	$\boldsymbol{\nu}\, X \cdot \varphi_1$		greatest fixpoint
$\mid$	$\forall\, \varphi_1 : \varphi_2$		universal state closure

Patrick Cousot, Radhia Cousot: Temporal Abstract Interpretation. POPL 2000: 12-25

Example of intrinsic approximate refinement

- Consider universal model-checking abstraction:

$$\mathrm{MC}^{\forall}_{M}(\phi) = \alpha^{\forall}_{M}([\![\phi]\!]) \in \wp(Traces) \to \wp(States)$$

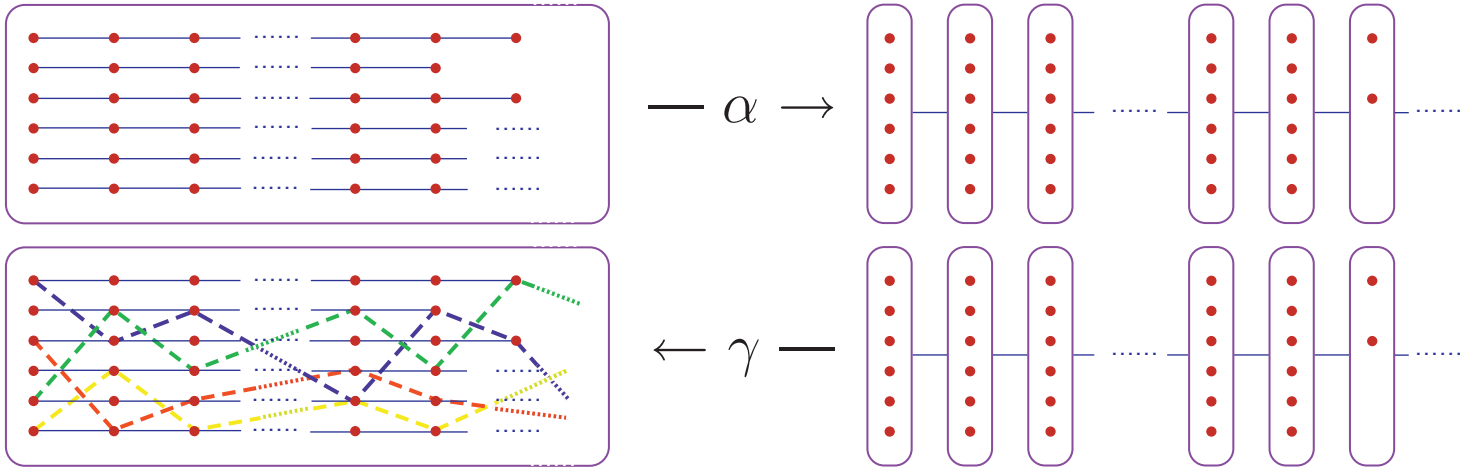
$$= \{s \in States \mid \forall \langle i, \sigma \rangle \in Traces_M . (\sigma_i = s) \Rightarrow \langle i, \sigma \rangle \in [\![\phi]\!]\}$$

where M is defined by a transition system

(and dually the existential model-checking abstraction)

Example of intrinsic approximate refinement

- The abstraction from a set of traces to a trace of sets is sound but *incomplete*, even for finite systems (*)

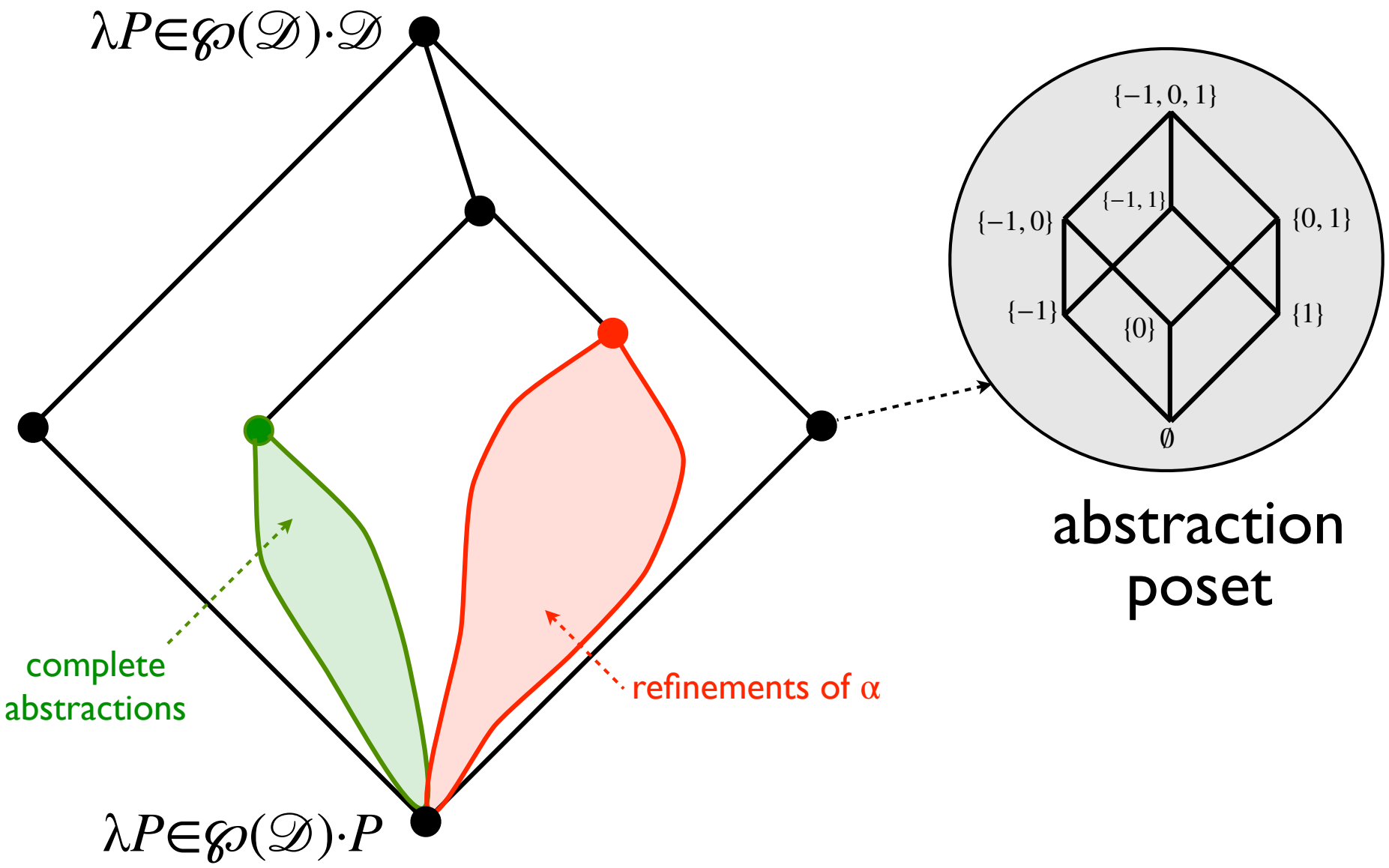


- Any refinement of this abstraction is *incomplete* (but to the infinite past/future trace semantics itself) (**)

(*) Patrick Cousot, Radhia Cousot: Temporal Abstract Interpretation. POPL 2000: 12-25

(**) Roberto Giacobazzi, Francesco Ranzato: Incompleteness of states w.r.t. traces in model checking. Inf. Comput. 204(3): 376-407 (2006)

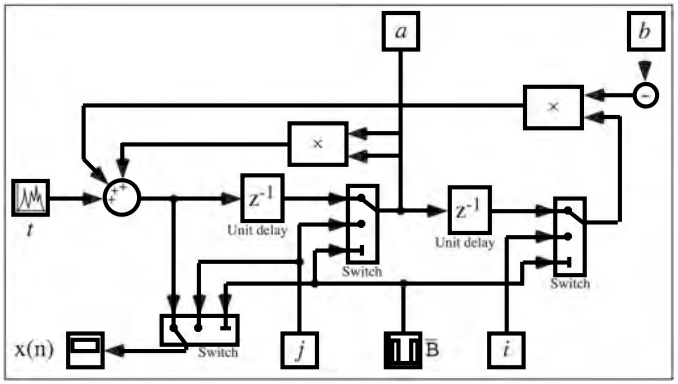
Intrinsic approximate refinement



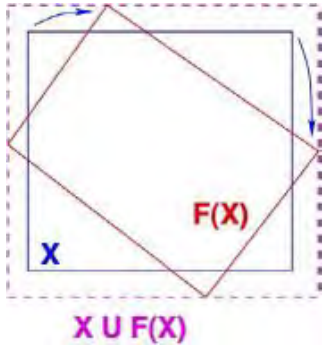
In general refinement does not terminate

- Example:filter invariant abstraction:

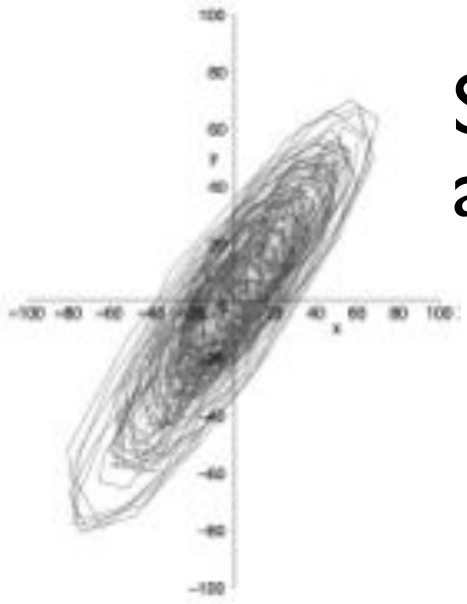
2nd order filter:



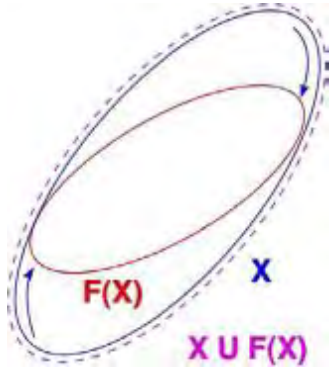
Unstable polyhedral abstraction:



Counter-example guided refinement will indefinitely add missing points according to the execution trace:



Stable ellipsoidal abstraction:



Julien Bertrane, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, & Xavier Rival. Static Analysis and Verification of Aerospace Software by Abstract Interpretation. In *AIAA Infotech@@Aerospace 2010*, Atlanta, Georgia. American Institute of Aeronautics and Astronautics, 20—22 April 2010. © AIAA.

In general refinement does not terminate

- Narrowing is needed to stop infinite iterated automatic refinements:

  e.g. SLAM stops refinement after 20mn (now abandoned by MS)

- Intelligence is needed for refinement:

  e.g. human-driven refinement of Astrée

Thomas Ball, Vladimir Levin, Sriram K. Rajamani: A decade of software model checking with SLAM. Commun. ACM 54(7): 68-76 (2011)

Julien Bertrane, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, & Xavier Rival. Static Analysis and Verification of Aerospace Software by Abstract Interpretation. In *AIAA Infotech@@Aerospace 2010*, Atlanta, Georgia. American Institute of Aeronautics and Astronautics, 20—22 April 2010. © AIAA.

Industrialization

Daniel Kästner, Christian Ferdinand, Stephan Wilhelm, Stefana Nevona, Olha Honcharova, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, Xavier Rival, and Élodie-Jane Sims. Astrée: Nachweis der Abwesenheit von Laufzeitfehlern. In *Workshop ``Entwicklung zuverlässiger Software-Systeme''*, Regensburg, Germany, June 18th, 2009.

Olivier Bouissou, Éric Conquet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Khalil Ghorbal, Éric Goubault, David Lesens, Laurent Mauborgne, Antoine Miné, Sylvie Putot, Xavier Rival, & Michel Turin. Space Software Validation using Abstract Interpretation. In *Proc. of the Int. Space System Engineering Conf., Data Systems in Aerospace (DASIA 2009)*. Istambul, Turkey, May 2009, 7 pages. ESA.

Jean Souyris, David Delmas: Experimental Assessment of Astrée on Safety-Critical Avionics Software. SAFECOMP 2007: 479-490

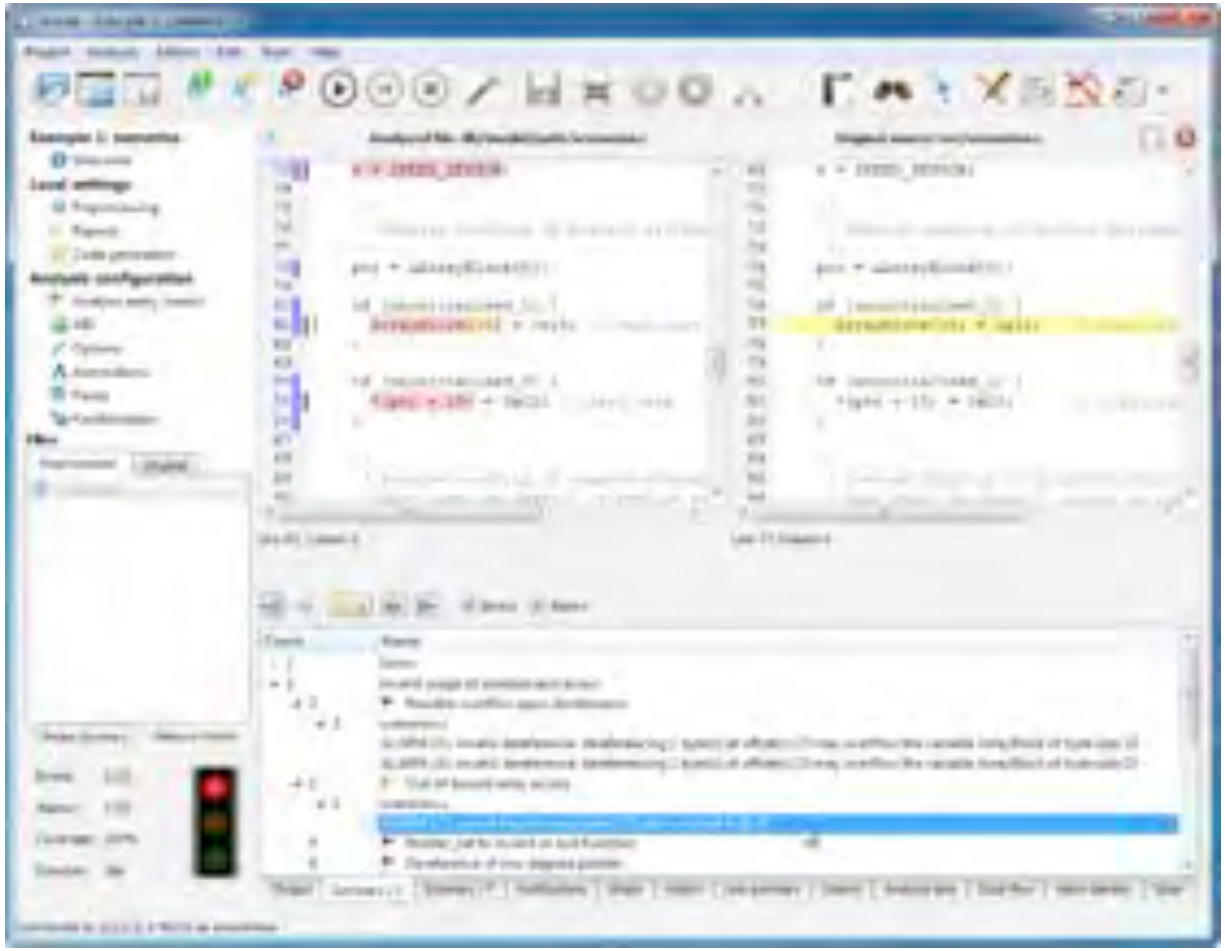
David Delmas, Jean Souyris: Astrée: From Research to Industry. SAS 2007: 437-451

Jean Souyris: Industrial experience of abstract interpretation-based static analyzers. IFIP Congress Topical Sessions 2004: 393-400

Stephan Thesing, Jean Souyris, Reinhold Heckmann, Famantanantsoa Randimbivololona, Marc Langenbach, Reinhard Wilhelm, Christian Ferdinand: An Abstract Interpretation-Based Timing Validation of Hard Real-Time Avionics Software. DSN 2003: 625-632

Astrée

- Commercially available: `www.absint.com/astree/`

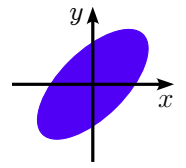


- Effectively used in production to qualify truly large and complex software in transportation, communications, medicine, *etc*

Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, Xavier Rival: **A static analyzer for large safety-critical software.** *PLDI 2003*: 196-207

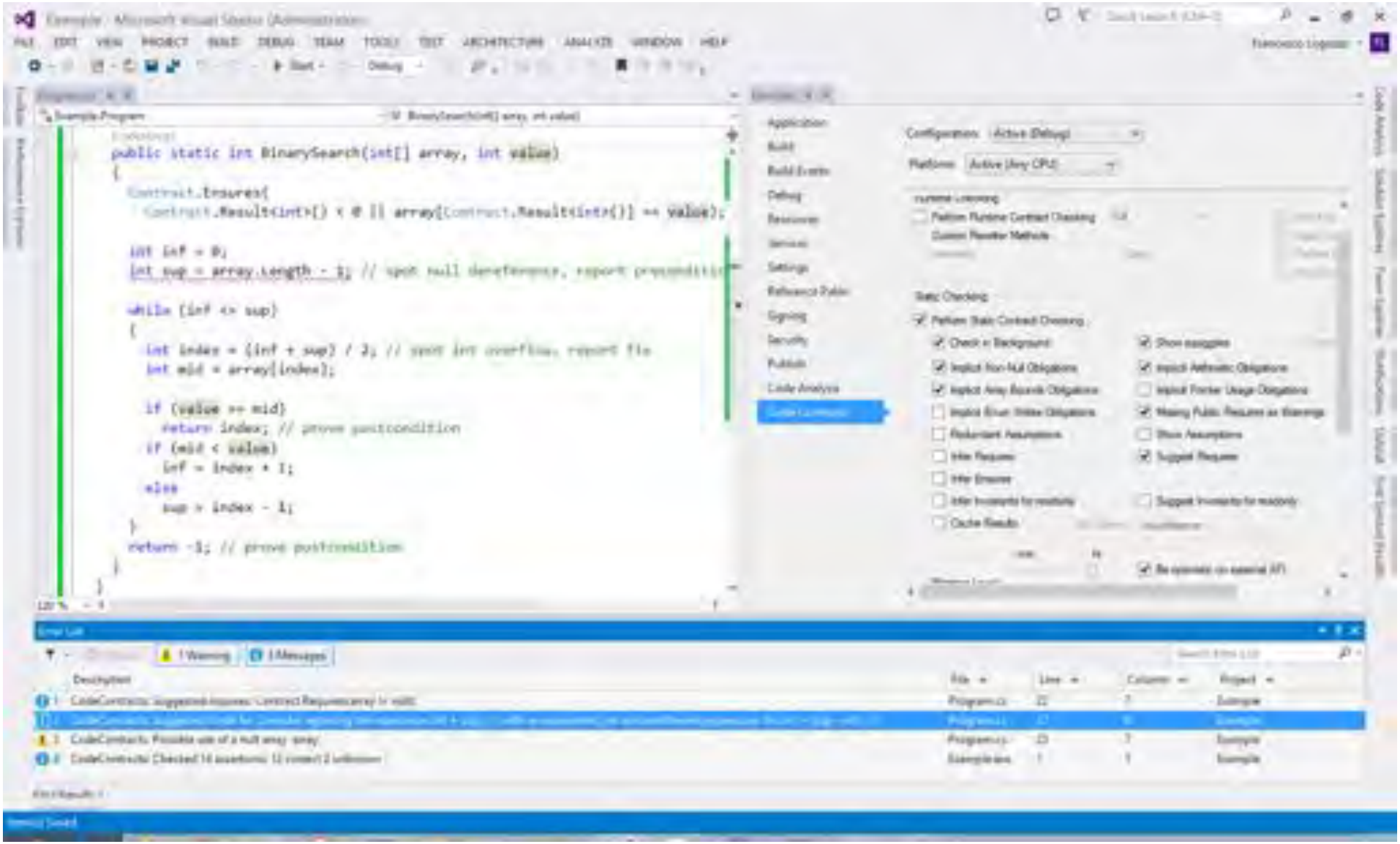
Example of domain-specific abstraction: ellipses

```
typedef enum {FALSE = 0, TRUE = 1} BOOLEAN;
BOOLEAN INIT; float P, X;
void filter () {
  static float E[2], S[2];
  if (INIT) { S[0] = X; P = X; E[0] = X; }
  else { P = (((((0.5 * X) - (E[0] * 0.7)) + (E[1] * 0.4))
             + (S[0] * 1.5)) - (S[1] * 0.7)); }
  E[1] = E[0]; E[0] = X; S[1] = S[0]; S[0] = P;
  /* S[0], S[1] in [-1327.02698354, 1327.02698354] */
}
void main () { X = 0.2 * X + 5; INIT = TRUE;
  while (1) {
    X = 0.9 * X + 35; /* simulated filter input */
    filter (); INIT = FALSE; }
}
```



Code Contract Static Checker (cccheck)

- Available within MS Visual Studio



Manuel Fähndrich, Francesco Logozzo: **Static Contract Checking with Abstract Interpretation**. *FoVeOOS 2010*: 10-30

Comments on screenshot (courtesy Francesco Logozzo)

- A screenshot from Clousot/`cccheck` on the classic binary search.
- The screenshot shows from left to right and top to bottom
  1. C# code + CodeContracts with a buggy BinarySearch
  2. `cccheck` integration in VS (right pane with all the options integrated in the VS project system)
  3. `cccheck` messages in the VS error list
- The features of `cccheck` that it shows are:
  1. basic abstract interpretation:
     a. the loop invariant to prove the array access correct and that the arithmetic operation may overflow is inferred fully automatically
     b. different from deductive methods as e.g. ESC/Java or Boogie where the loop invariant must be provided by the end-user
  2. inference of necessary preconditions:
     a. Clousot finds that array may be null (message 3)
     b. Clousot suggests and propagates a necessary precondition invariant (message 1)
  3. array analysis (+ disjunctive reasoning):
     a. to prove the postcondition should infer property of the content of the array
     b. please note that the postcondition is true even if there is no precondition requiring the array to be sorted.
  4. verified code repairs:
     a. from the inferred loop invariant does not follow that index computation does not overflow
     b. suggest a code fix for it (message 2)

Software

- Ait: static analysis of the worst-case execution time of control/command software (www.absint.com/ait/)
- Astrée: proof of absence of runtime errors in embedded synchronous real time control/command software (www.absint.com/astree/), AstréeA for asynchronous programs (www.astreea.ens.fr/)
- C Global Surveyor, NASA, static analyzer for flight software of NASA missions (www.cmu.edu/silicon-valley/faculty-staff/venet-arnaud.html)
- IKOS (Inference Kernel for Open Static Analyzers), (www.cmu.edu/silicon-valley/software-systems-management/software-verification.html)
- Checkmate: static analyzer of multi-threaded Java programs (www.pietro.ferrara.name/checkmate/)
- CodeContracts Static Checker, Microsoft (msdn.microsoft.com/en-us/devlabs/dd491992.aspx)
- Fluctuat: static analysis of the precision of numerical computations (www-list.cea.fr/labos/gb/LSL/fluctuat/index.html)

Software

- Infer: Static analyzer for C/C++ (monoidics.com/)
- Julia: static analyzer for Java and Android programs (www.juliasoft.com/juliasoft-android-java-verification.aspx?Id=201177234649)
- Predator: static analyzer of C dynamic data structures using separation logic (www.fit.vutbr.cz/research/groups/verifit/tools/predator/)
- Terminator: termination proof (www.cs.ucl.ac.uk/staff/p.ohearn/Invader/Invader/Invader_Home.html)
- etc.

- Apron numerical domains library (apron.cri.ensmp.fr/library/)
- Parma Polyhedral Library (bugseng.com/products/ppl/)
- etc.

Hardware

- (Generalized) symbolic trajectory evaluation (Intel)

Intel's Successes with Formal Methods

John Harrison

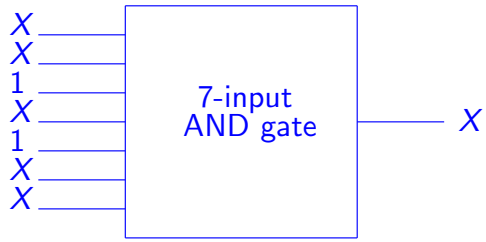
Intel Corporation

15 March 2012

Tsinghua software day, March 15, 2012, Tsinghua University, Beijing, China

Example of ternary simulation

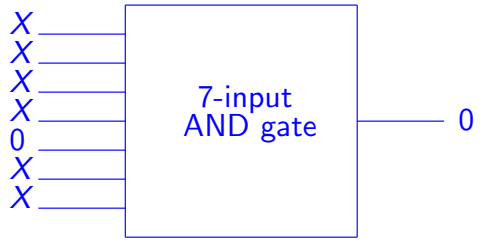
If some inputs are undefined, the output often is too, but not always:



$0 = \{0\}$

$1 = \{1\}$

$X = \{0, 1\}$



Jin Yang and Carl-Johan H. Seger, *Generalized Symbolic Trajectory Evaluation — Abstraction in Action*, Formal Methods in Computer-Aided Design, Lecture Notes in Computer Science, 2002, Volume 2517/2002, 70–87.

Jin Yang; Seger, C.-J.H.; *Introduction to generalized symbolic trajectory evaluation*, IEEE Transactions onVery Large Scale Integration (VLSI) Systems 11(3), June 2003, 345–353.

Biology

- Kappa – A language for modeling protein interaction networks by a set of rules and analyse that set directly deploying techniques from abstract interpretation (www.kappalanguage.org/ and fontana.med.harvard.edu/www/Documents/Lab/research.signaling.htm)

References

References

- Many references to be found in

  Patrick Cousot and Radhia Cousot
  Abstract interpretation: Past, Present, and Future

  In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/ IEEE Symposium on Logic In Computer Science (LICS)* , July 14—18, 2014, Vienna, Austria.

Conclusion

Abstract interpretation

- Intellectual tool (not to be confused with its specific application to iterative static analysis with ∇ & Δ)

- No cathedral would have been built without plumb-line and square, certainly not enough for skyscrapers:

  Powerful tools are needed for progress and applicability of formal methods

Abstract interpretation

- Varieties of researchers in formal methods:

  (i) explicitly use abstract interpretation, and are happy to extend its scope and broaden its applicability

  (ii) implicitly use abstract interpretation, and hide it

  (iii) pretend to use abstract interpretation, but misuse it

  (iv) don't know that they use abstract interpretation, but would benefit from it

- Never too late to upgrade

The End, Thank You