Comparing completeness properties of static analyses and their logics

David Schmidt

Kansas State University

www.cis.ksu.edu/~schmidt

Abstract interpretation: computing on properties

```
readInt(x)
if x>0 :
   x:= pred(x)
x:= succ(x)
writeInt(x)
```

Q: Is output pos?

A: abstractly interpret domain Int by $Sign = \{neg, zero, pos, any\}$:

```
readSign(x)
if isPos(x):
   x:= pred♯(x)
x:= succ♯(x)
writeSign(x)
```

where

$$\begin{aligned} succ^\sharp(pos) &= pos \\ succ^\sharp(zero) &= pos \\ succ^\sharp(neg) &= any \\ succ^\sharp(any) &= any \end{aligned}$$

and

$$\begin{aligned} pred^\sharp(neg) &= neg \\ pred^\sharp(zero) &= neg \\ pred^\sharp(pos) &= any \\ pred^\sharp(any) &= any \end{aligned}$$

To answer the question, calculate the *static analysis*:

$$\{zero \mapsto pos,\ neg \mapsto any,\ pos \mapsto any,\ any \mapsto any\}$$

The Question is decided only for $zero$ — the static analysis is *sound* but *incomplete*.

Let $\mathit{Sign}' = \{\mathit{neg}, \leq 0, \mathit{zero}, \geq 0, \mathit{pos}, \mathit{any}\}$

```
readInt(x)
if x>0 :
   x:= pred(x)
x:= succ(x)
writeInt(x)
```

$\mathit{readSign}(x)$
if $\mathit{isPos}(x)$:
  x:= $\mathit{pred}^\sharp(x)$
x:= $\mathit{succ}^\sharp(x)$
$\mathit{writeSign}(x)$

where

$$\begin{aligned}
\mathit{succ}^\sharp(\mathit{pos}) &= \mathit{pos} \\
\mathit{succ}^\sharp(\geq 0) &= \mathit{pos} \quad (\text{-:} \\
\mathit{succ}^\sharp(\mathit{zero}) &= \mathit{pos} \\
\mathit{succ}^\sharp(\leq 0) &= \mathit{any} \\
\mathit{succ}^\sharp(\mathit{neg}) &= \leq 0 \quad (\text{-:} \\
\mathit{succ}^\sharp(\mathit{any}) &= \mathit{any}
\end{aligned}$$

and

$$\begin{aligned}
\mathit{pred}^\sharp(\mathit{neg}) &= \mathit{neg} \\
\mathit{pred}^\sharp(\leq 0) &= \mathit{neg} \quad (\text{-:} \\
\mathit{pred}^\sharp(\mathit{zero}) &= \mathit{neg} \\
\mathit{pred}^\sharp(\geq 0) &= \mathit{any} \\
\mathit{pred}^\sharp(\mathit{pos}) &= \geq 0 \quad (\text{-:} \\
\mathit{pred}^\sharp(\mathit{any}) &= \mathit{any}
\end{aligned}$$

The static analysis on Sign':

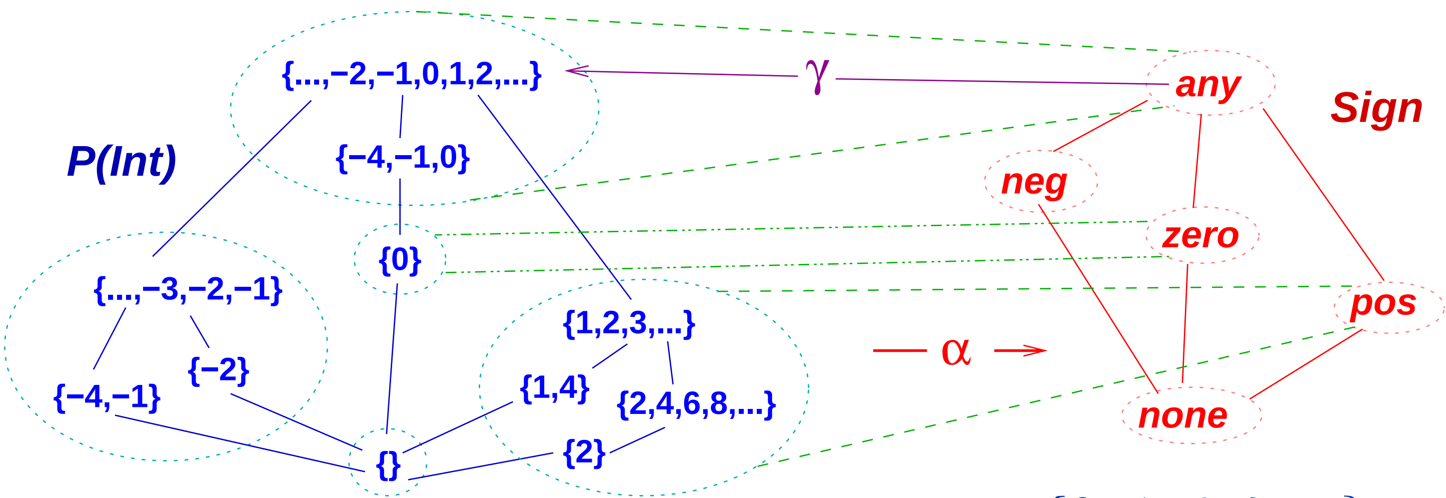
$\mathit{neg} \mapsto \mathit{neg}$ (-: $\quad \leq 0 \mapsto \mathit{any} \quad \mathit{zero} \mapsto \mathit{pos}$

$\mathit{pos} \mapsto \mathit{pos}$ (-: $\quad \mathit{any} \mapsto \mathit{any} \quad \geq 0 \mapsto \mathit{any}$)-:

Summary of the talk

1. Every static analysis employs an *abstract domain*, and every abstract domain possesses an *internal logic*.
2. Abstract state transformers must be *sound*, and perhaps they are *Backwards- Forwards-complete*.
3. Most program logics *extend* an internal logic, and their abstractions must be sound.
4. There are both *over-* and *underapproximating Galois connections* for approximating program logics; these define *F-, B-*, and *O-logical-completeness*.
5. The completeness notions are independent (and the independences are significant), but *coverings* are used to relate them.

Concrete data abstracts to (logical) properties



$\gamma(\mathrm{pos}) = \{1, 2, 3, \cdots\}$

$\gamma(\mathrm{zero}) = \{0\}$, etc.

$\gamma(\mathrm{any}) = \mathrm{Int}$, etc.

$\alpha\{2, 4, 6, 8, ...\} = \mathrm{pos}$

$\alpha\{-4, -1, 0\} = \mathrm{any}$

$\alpha\{0\} = \mathrm{zero}$

$\alpha\{\} = \mathrm{none}$, etc.

$(\mathcal{P}(\mathrm{Int}), \subseteq)\langle\alpha, \gamma\rangle(\mathrm{Sign}, \sqsubseteq)$ is a *Galois connection*: γ interprets the properties, and $\alpha(S) = \sqcap\{a \mid \gamma(a) \subseteq S\}$ maps concrete set S to the property that best describes it [CousotCousot77] .

We use such structures to do *static analysis*.

A Galois connection defines an **internal logic**

For $(PC, \subseteq)\langle\alpha, \gamma\rangle(A, \sqsubseteq)$, $S \in PC$, and $a \in A$, define

$$S \models a \text{ iff } S \subseteq \gamma(a) \text{ iff } \alpha(S) \sqsubseteq a$$

Example: For $Sign$, $\{2, 8\} \models pos$.

A Galois connection defines a logic with conjunction:

$$\phi \ ::= \ a \mid \phi_1 \sqcap \phi_2$$

because γ preserves $\sqcap$ as $\cap$ (that is, $\gamma(a_1 \sqcap a_2) = \gamma(a_1) \cap \gamma(a_2)$):

$$S \models a_1 \sqcap a_2 \text{ iff } S \models a_1 \text{ and } S \models a_2.$$

Example: In $Sign$, $\{2, 5\} \models pos \sqcap any$.

But the logic for $Sign$ excludes disjunction, e.g., $\{0\} \models any = neg \sqcup pos$, yet $\{0\} \not\models neg$ and $\{0\} \not\models pos$. This is because γ does not preserve $\sqcup$ as $\cup$.

Abstract transformers compute on properties

For $f : PC \rightarrow PC$, $f^\sharp : A \rightarrow A$ is *sound* iff

$$\alpha \circ f \sqsubseteq f^\sharp \circ \alpha \quad \text{iff} \quad f \circ \gamma \sqsubseteq \gamma \circ f^\sharp$$

α and γ act as *semi-homomorphisms*; $f^\sharp$ is a *postcondition transformer*.

Example: For $succ : \mathcal{P}(Int) \rightarrow \mathcal{P}(Int)$, $succ\{0\} = \{1\}$, $succ^\sharp(zero) = pos$.

Consequences: $f(S) \models f^\sharp(\alpha(S))$ and $f(\gamma(a)) \models f^\sharp(a)$.

For example, $\{0\} \models zero$ and $succ\{0\} \models succ^\sharp(zero) = pos$.

$f^\sharp_{best} = \alpha \circ f \circ \gamma$ is the best — strongest postcondition — transformer in A's internal logic.

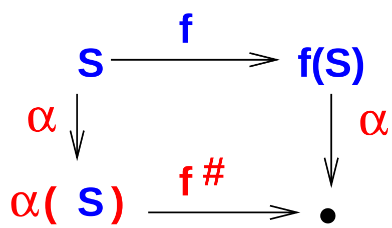
(Functional) completeness: *from semi-homomorphism to homomorphism*

For $f : PC \to PC$, $f^\sharp : A \to A$:

Backwards(α)-completeness

[Cousots79,GiacobazziJACM00] :

$$\alpha \circ f = f^\sharp \circ \alpha$$



α is a homomorphism from PC to A — it preserves f as $f^\sharp$.

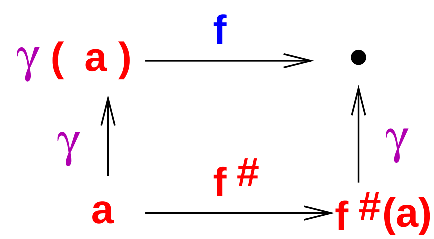
Corollary: $f^\sharp(\alpha(S)) \sqsubseteq a$ iff $f(S) \models a$.

That is, we can *decide* properties of f in A.

Forwards(γ)-completeness

[GiacobazziQuintarelli01] :

$$f \circ \gamma = \gamma \circ f^\sharp$$



γ is a homomorphism from A to PC — it preserves $f^\sharp$ as f.

Corollary: $S \models f^\sharp(a)$ iff $S \subseteq f(\gamma(a))$.

That is, $f^\sharp$ is a logical connective in A's internal logic (like $\sqcap$ is).

A typical program logic extends A's internal logic

Given Galois connection, $(\mathcal{P}(D), \subseteq)\langle\alpha, \gamma\rangle(A, \sqsubseteq)$, define $\mathcal{L}$ as follows:

$$a \in \mathrm{Prim} = A \text{ (the primitive assertions)}$$

$$\mathcal{L} \ni \phi ::= a \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid [f]\phi$$

The interpretation, $[\![\cdot]\!] : \mathcal{L} \to \mathcal{P}(D)$, is defined as

$[\![a]\!] = \gamma(a)$

$[\![\phi_1 \wedge \phi_2]\!] = [\![\phi_1]\!] \cap [\![\phi_2]\!]$

$[\![\phi_1 \vee \phi_2]\!] = [\![\phi_1]\!] \cup [\![\phi_2]\!]$

$[\![[f]\phi]\!] = \widetilde{pre}_f[\![\phi]\!]$

where $\widetilde{pre}_f(S) = \{c \in D \mid f(c) \subseteq S\}$, and $f : D \to \mathcal{P}(D)$ is a state-transition function

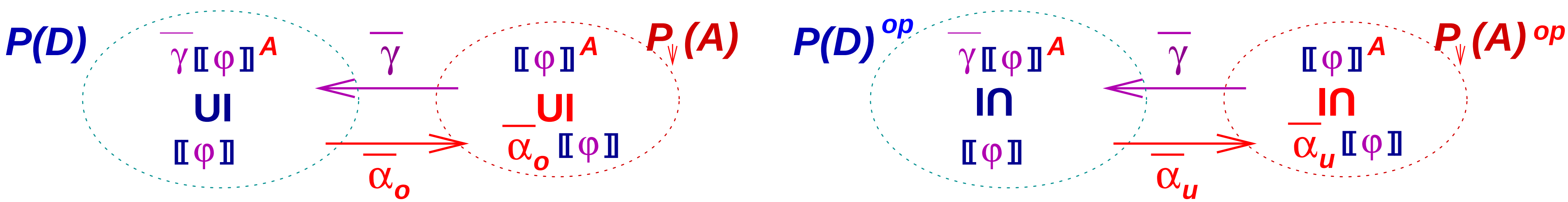
Say that $S \models \phi$ iff $S \subseteq [\![\phi]\!]$.

$\phi_1 \vee \phi_2$ and $[f]\phi$ might not fall in A's internal logic. (E.g., for Sign, there is no $\cup : \mathit{Sign} \times \mathit{Sign} \to \mathit{Sign}$ such that $\gamma(\mathit{neg} \cup \mathit{pos}) = [\![\phi_1 \vee \phi_2]\!]$.)

Q: How do we approximate $[\![\cdot]\!] : \mathcal{L} \to \mathcal{P}(D)$?
A: Define $[\![\cdot]\!]^A : \mathcal{L} \to \mathcal{P}_\downarrow(A)$

Given $(\mathcal{P}(D), \subseteq)\langle\alpha, \gamma\rangle(A, \sqsubseteq)$, we have *two* relevant Galois connections between $\mathcal{P}(D)$ and $\mathcal{P}_\downarrow(A)$:

Define $\overline{\gamma}(T) = \cup_{a \in T}\, \gamma(a)$.



Overapproximating abstraction:

$$\overline{\alpha_o}(S) = \bigcap\{T \mid S \subseteq \overline{\gamma}(T)\}$$
$$= \downarrow\{\alpha\{c\} \mid c \in S\}$$

where

$\downarrow T = \{a \mid$ exists $a' \in T, a \sqsubseteq a'\}$.

Underapproximating abstraction:

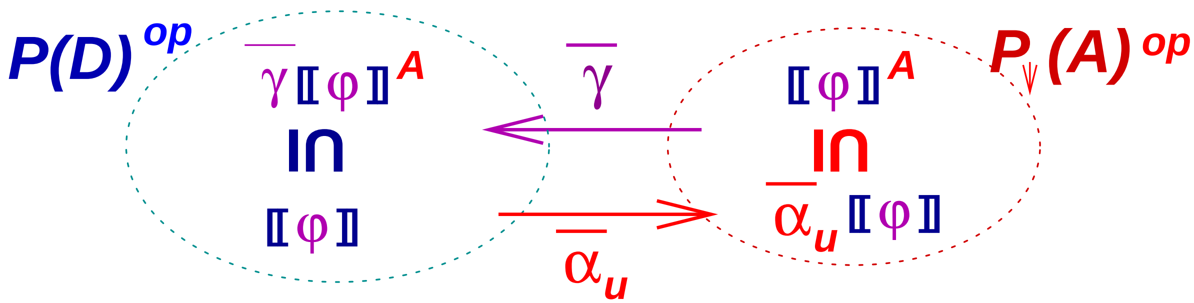
$$\overline{\alpha_u}(S) = \bigcup\{T \mid \overline{\gamma}(T) \subseteq S\}$$
$$= \{a \mid \gamma(a) \subseteq S\}$$

where

$(P, \sqsubseteq_P)^{op}$ is $(P, \sqsupseteq_P)$.

Abstracted, underapproximated logic

This is the best inductively defined underapproximation:



$$[\![a]\!]^A_{\text{best}} = \overline{\alpha_u}(\gamma(a))$$

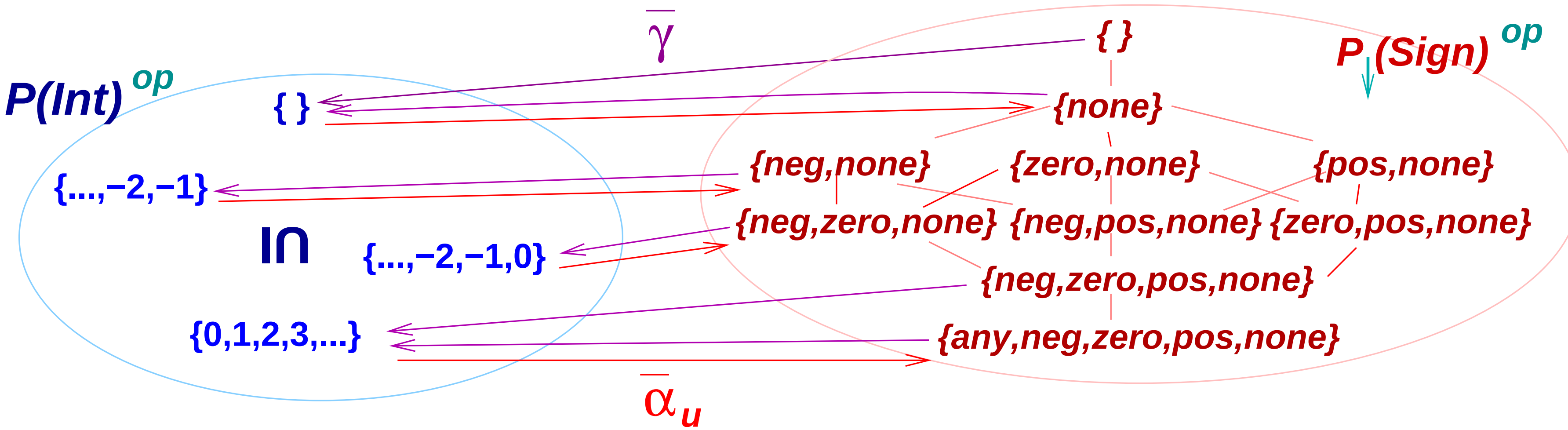
$$[\![\phi_1 \wedge \phi_2]\!]^A_{\text{best}} = [\![\phi_1]\!]^A_{\text{best}}(\overline{\alpha_u} \circ \cap \circ \overline{\gamma}^2)[\![\phi_2]\!]^A_{\text{best}}$$

$$[\![\phi_1 \vee \phi_2]\!]^A_{\text{best}} = [\![\phi_1]\!]^A_{\text{best}}(\overline{\alpha_u} \circ \cup \circ \overline{\gamma}^2)[\![\phi_2]\!]^A_{\text{best}}$$

$$[\![[f]\phi]\!]^A_{\text{best}} = (\overline{\alpha_u} \circ \widetilde{\text{pre}}_{f\sharp} \circ \overline{\gamma})[\![\phi]\!]^A_{\text{best}}$$

But it is not finitely computable in A....

Here is a less precise, but sound and finitely computable underapproximation for Sign:



$$[\![a]\!]^{\mathit{Sign}} = \ \downarrow\{a\} \ = \{a' \mid a' \sqsubseteq a\}$$

$$[\![\phi_1 \wedge \phi_2]\!]^{\mathit{Sign}} = [\![\phi_1]\!]^{\mathit{Sign}} \cap [\![\phi_2]\!]^{\mathit{Sign}}$$

$$[\![\phi_1 \vee \phi_2]\!]^{\mathit{Sign}} = [\![\phi_1]\!]^{\mathit{Sign}} \cup [\![\phi_2]\!]^{\mathit{Sign}}$$

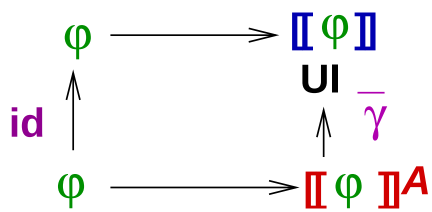
$$[\![[f]\phi]\!]^{\mathit{Sign}} = \ \widetilde{\mathit{pre}}_{f^\sharp}[\![\phi]\!]^{A}$$

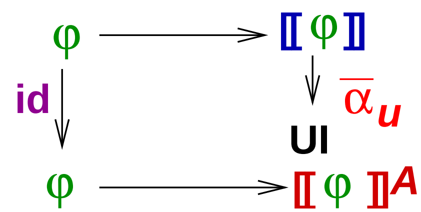
We have soundness: $\overline{\alpha_u}[\![\phi]\!] \supseteq [\![\phi]\!]^{\mathit{Sign}}_{\mathit{best}} \supseteq [\![\phi]\!]^{\mathit{Sign}}$, for all ϕ.

Logical soundness and completeness

$[\![\cdot]\!]^A : \mathcal{L} \to \mathcal{P}_\downarrow(\mathrm{Sign})$ is *sound* for $[\![\cdot]\!] : \mathcal{L} \to \mathcal{P}(D)$ iff

$$\overline{\gamma}[\![\phi]\!]^A \subseteq [\![\phi]\!] \quad \text{iff} \quad [\![\phi]\!]^A \subseteq \overline{\alpha_u}[\![\phi]\!]$$



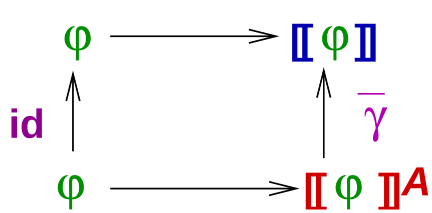


There are two forms of *completeness* of $[\![\cdot]\!]^A$ for $[\![\cdot]\!]$:

Forwards-completeness

[RanzatoTapparo06] :

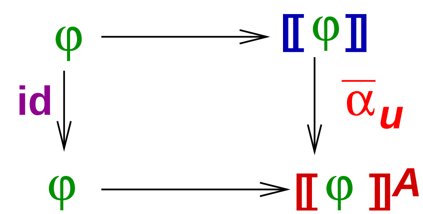
$$\overline{\gamma}[\![\phi]\!]^A = [\![\phi]\!]$$



Backwards-completeness

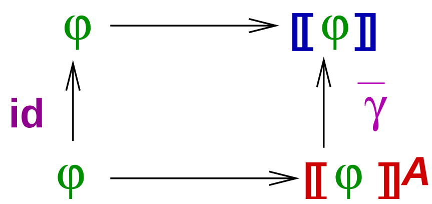
[CousotCousot00] :

$$[\![\phi]\!]^A = \overline{\alpha_u}[\![\phi]\!]$$

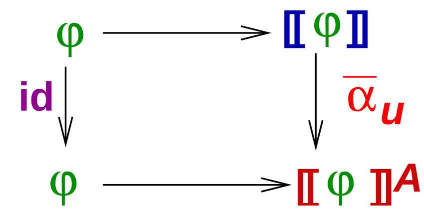


Strong, best, and lower preservation

F-complete: $\overline{\gamma}[\![\phi]\!]^A = [\![\phi]\!]$



B-complete: $[\![\phi]\!]^A = \overline{\alpha_u}[\![\phi]\!]$



- *best preservation:* for all $\phi \in \mathcal{L}$ and $T \in \mathcal{P}_{\downarrow}(A)$,
  $T \subseteq [\![\phi]\!]^A$ iff $\overline{\gamma}(T) \subseteq [\![\phi]\!]$.

- *strong preservation:* for all $\phi \in \mathcal{L}$ and $S \in \mathcal{P}(D)$,
  $S \subseteq [\![\phi]\!]$ iff $\overline{\alpha_o}(S) \subseteq [\![\phi]\!]^A$.

- *lower preservation:* for all $\phi \in \mathcal{L}$ and $S \in \mathcal{P}(D)$,
  $[\![\phi]\!] \subseteq S$ iff $[\![\phi]\!]^A \subseteq \overline{\alpha_u}(S)$.

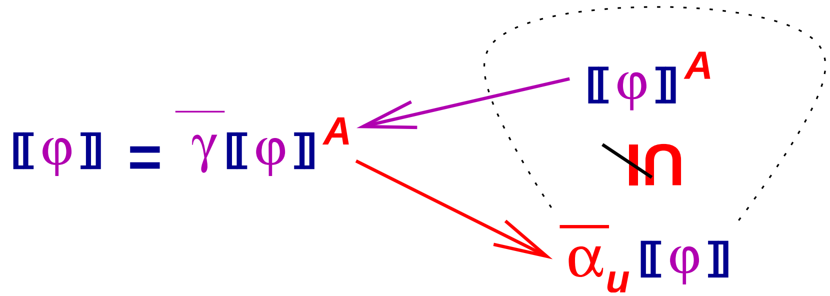
Theorem:

- B-complete iff best preservation

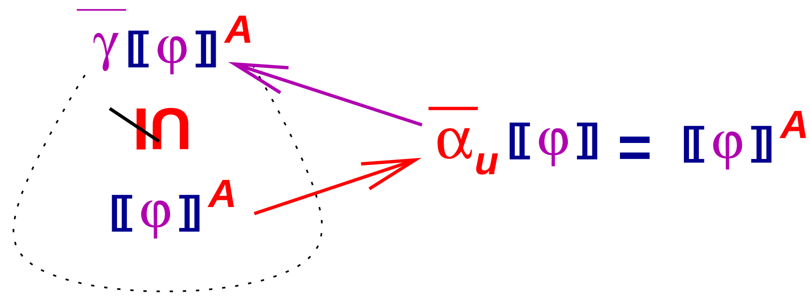
- F-complete iff strong preservation iff lower preservation

The two forms of completeness are **independent**

F-complete and not B-complete:



B-complete and not F-complete:



Absence of B-completeness: we fail to validate $any \in [\![neg \vee zero \vee pos]\!]^{Sign}$. We must use a *focus* operation: $focus(any) = \{neg, zero, pos\}$ and validate $a \in [\![neg \vee zero \vee pos]\!]^{Sign}$, for all $a \in focus(any)$ [Dams04,Sagiv02] .

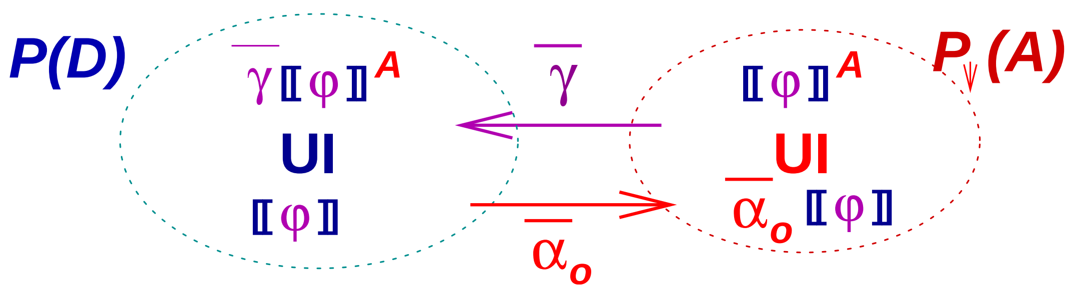
Absence of F-completeness: Say that $eq1 \in \mathcal{L}$ and $[\![eq1]\!] = \{1\}$, making $[\![eq1]\!]^{Sign} = \{none\}$. Then, a static analysis of

```
x:= 1; if x=1 then safe() else error()
```

announces `error()` is reachable. *Counterexample guided abstraction refinement (CEGAR)* repairs the problem by adding new elements to $Sign$ [Ball02,Clarke00,Saidi00] .

O-completeness: *subset-inclusion completeness*

For $\mathcal{P}(\mathrm{D})\langle\overline{\alpha_{\mathrm{o}}}, \overline{\gamma}\rangle\mathcal{P}_{\downarrow}(A)$,



the inclusion,

$$\overline{\alpha_{\mathrm{o}}}[\![\phi]\!] \subseteq [\![\phi]\!]^{A}$$

does not ensure soundness. Nonetheless, we can define one more variant of completeness (which *is* sound):

$$[\![\cdot]\!]^{A} \text{ is } B(\overline{\alpha_{\mathrm{o}}})\text{-complete for } [\![\cdot]\!] \text{ iff } \overline{\alpha_{\mathrm{o}}}[\![\phi]\!] = [\![\phi]\!]^{A}.$$

We use *O-complete* as a synonym for *B($\overline{\alpha_{\mathrm{o}}}$)-complete*.

With coverings, *we make many connections*

For $[\![\cdot]\!] : \mathcal{L} \to \mathcal{P}(\mathrm{D})$ and $\overline{\gamma} : \mathrm{Q} \to \mathcal{P}(\mathrm{D})$,

$\overline{\gamma}$ *covers* $[\![\cdot]\!]$ iff for all $\phi \in \mathcal{L}$, $[\![\phi]\!] \in \overline{\gamma}[\mathrm{Q}]$.

For $[\![\cdot]\!]^A : \mathcal{L} \to \mathcal{P}_\downarrow(A)$ and $\overline{\alpha} : \mathrm{P} \to \mathcal{P}_\downarrow(A)$,

$\overline{\alpha}$ *covers* $[\![\cdot]\!]^A$ iff for all $\phi \in \mathcal{L}$, $[\![\phi]\!]^A \in \overline{\alpha}[\mathrm{P}]$.

Theorem:

- If $[\![\cdot]\!]^A$ is F-complete for $[\![\cdot]\!]$ and $\overline{\alpha_u}$ covers $[\![\cdot]\!]^A$, then $[\![\cdot]\!]^A$ is B-complete.
- If $[\![\cdot]\!]^A$ is B-complete for $[\![\cdot]\!]$ and $\overline{\gamma}$ covers $[\![\cdot]\!]$, then $[\![\cdot]\!]^A$ is F-complete.
- If $[\![\cdot]\!]^A$ is F-complete for $[\![\cdot]\!]$ and $\overline{\alpha_o}$ covers $[\![\cdot]\!]^A$, then $[\![\cdot]\!]^A$ is O-complete.
- If $[\![\cdot]\!]^A$ is O-complete for $[\![\cdot]\!]$ and $\overline{\gamma}$ covers $[\![\cdot]\!]$, then $[\![\cdot]\!]^A$ is *sound* as well as F-complete.

An application: partition domains

Let D and A be discretely ordered sets, and let $\delta : D \to A$ be an onto function, defining the *partition*, $c \sim_\delta c'$ iff $\delta(c) = \delta(c')$. Define $\gamma : A \to \mathcal{P}(D)$ as $\gamma(a) = \delta^{-1}(a)$. We have this propositional logic:

$$[\![a]\!] = \gamma(a) \qquad [\![\phi_1 \wedge \phi_2]\!] = [\![\phi_1]\!] \cap [\![\phi_2]\!]$$

$$[\![\neg\phi]\!] = \sim[\![\phi]\!] \qquad [\![\phi_1 \vee \phi_2]\!] = [\![\phi_1]\!] \cup [\![\phi_2]\!]$$

The abstract logic,

$$[\![a]\!]^A = \overline{\alpha_u}(\gamma(a)) \qquad [\![\phi_1 \wedge \phi_2]\!]^A = [\![\phi_1]\!]^A \cap [\![\phi_2]\!]^A$$

$$[\![\neg\phi]\!]^A = \sim [\![\phi]\!]^A \qquad [\![\phi_1 \vee \phi_2]\!]^A = [\![\phi_1]\!]^A \cup [\![\phi_2]\!]^A$$

is F-complete and equals $[\![\cdot]\!]^A_{best}$. Since both $\overline{\alpha_u}$ and $\overline{\alpha_o}$ cover $[\![\cdot]\!]^A$, the logic is also B- and O-complete.

The usual application of a partition domain is to model checking, whose logic includes the modality, $[f]\phi$, for $f : D \rightarrow \mathcal{P}(D)$, which is abstracted by a sound $f^\sharp : A \rightarrow \mathcal{P}(A)$ as follows:

$$[\![[f]\phi]\!]^A = \widetilde{pre}_{f^\sharp_{best}} [\![\phi]\!]^A, \quad \text{where } \widetilde{pre}_{f^\sharp}(T) = \{a' \mid f^\sharp(a') \subseteq T\}.$$

We know that $\widetilde{pre}_{f^\sharp_{best}} \; = \; (\widetilde{pre}_f)^\sharp{}_{best} \; = \; \overline{\alpha_u} \circ \widetilde{pre}_f \circ \overline{\gamma}$ [Schmidt06] .
The definition is sound but might not be complete — *this depends on* f:

Theorem: For $\widetilde{pre}_f : \mathcal{P}(D) \rightarrow \mathcal{P}(D)$, $f : D \rightarrow \mathcal{P}(D)$, and $f^* : \mathcal{P}(D) \rightarrow \mathcal{P}(D)$, defined as $f^*(S) = \cup_{c \in S} f(c)$,

1. $\widetilde{pre}_f$ is F($\overline{\gamma}$)-complete iff f^* is B($\overline{\alpha_o}$)-complete.

2. $\widetilde{pre}_f$ is B($\overline{\alpha_u}$)-complete iff f^* is F($\overline{\gamma}$)-complete.

Summary

1. Every static analysis employs an *abstract domain*, and every abstract domain possesses an *internal logic*.

2. Abstract state transformers must be *sound*, and perhaps they are *Backwards- Forwards-complete*.

3. Most program logics *extend* an internal logic, and their abstractions must be sound.

4. There are both *over-* and *underapproximating Galois connections* for approximating program logics; these define *F-, B-*, and *O-logical-completeness*.

5. The completeness notions are independent (and the independences are significant), but *coverings* are used to relate them.

References **This talk:** `www.cis.ksu.edu/~schmidt/papers`

1. E.M. Clarke, et al. Counterexample-guided abstraction refinement. CAV'00.
2. P. Cousot and R.Cousot. Systematic design of program analysis frameworks. POPL'79.
3. P. Cousot and R.Cousot. Temporal abstract interpretation. POPL'00.
4. R. Giacobazzi, F. Ranzato, and F. Scozzari. Making abstract interpretations complete. *J. ACM* 47(2000).
5. R. Giacobazzi and E. Quintarelli. Incompleteness, counterexamples, and refinements in abstract model checking. SAS'01.
6. F. Ranzato and F. Tapparo. Strong preservation of temporal fixpoint-based operators by abstract interpretation. VMCAI'06.
7. M. Sagiv, T. Reps, and R. Wilhelm. Parametric shape analysis via 3-valued logic. *ACM TOPLAS* 24(2002).
8. D.A. Schmidt. Comparing completeness properties of static analyses and their logics. APLAS'06.