Abstract Interpretation, Reloaded

Jan Midtgaard

Winter School, Day 3

http://janmidtgaard.dk/aiws15/

Saint Petersburg, Russia, 2015

Yesterday

Semantics overflow:

- The three counter machine
- An abstract machine for CPS terms
- A flow-chart semantics for IMP (non-deterministic!)
- A JVM-like semantics for a bytecode instruction set (objects,classes,methods,fields,. . .)

Finally we

- had a second look at collecting semantics and
- started massaging the collecting semantics of three counter machine

Today

- Approximation methods for AI (Cousot-Cousot:JLP92)
  - Lattice and fixed point theory
    - fixed points,
    - Galois connections
  - The Galois approach (p.11-. . .)
- From collecting semantics to static analysis
- More fun with Plotkin's three counter machine

Fixed points, reloaded

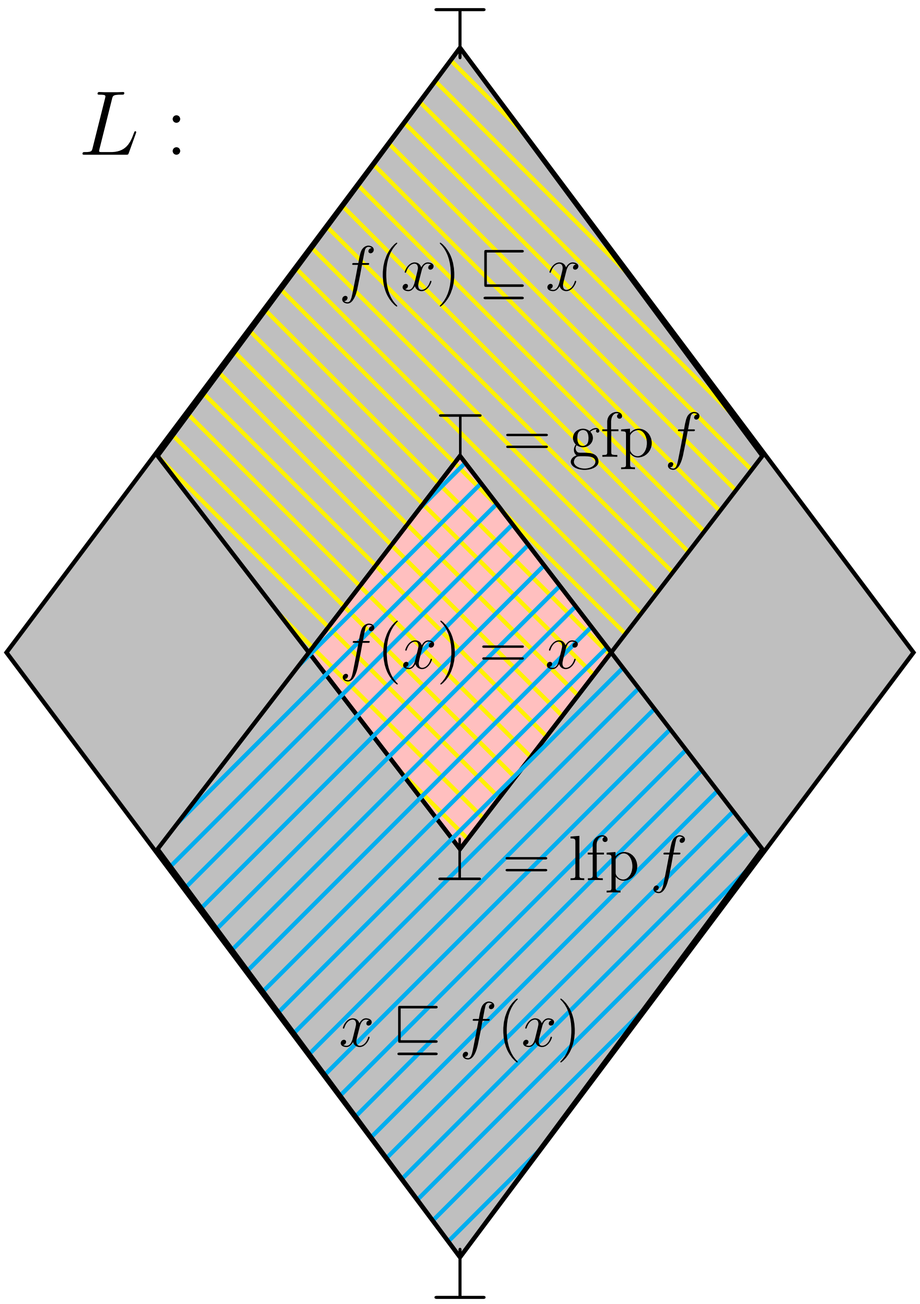
Tarski's fixed-point theorem

Theorem. *(Tarski:PJM55) Let L be a complete lattice $\langle L; \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$, and let f be a monotone function. Then the set P of all fixed points of f forms a complete lattice $\langle P; \sqsubseteq, \mathrm{lfp}\, f, \mathrm{gfp}\, f, \sqcup, \sqcap \rangle$ where*

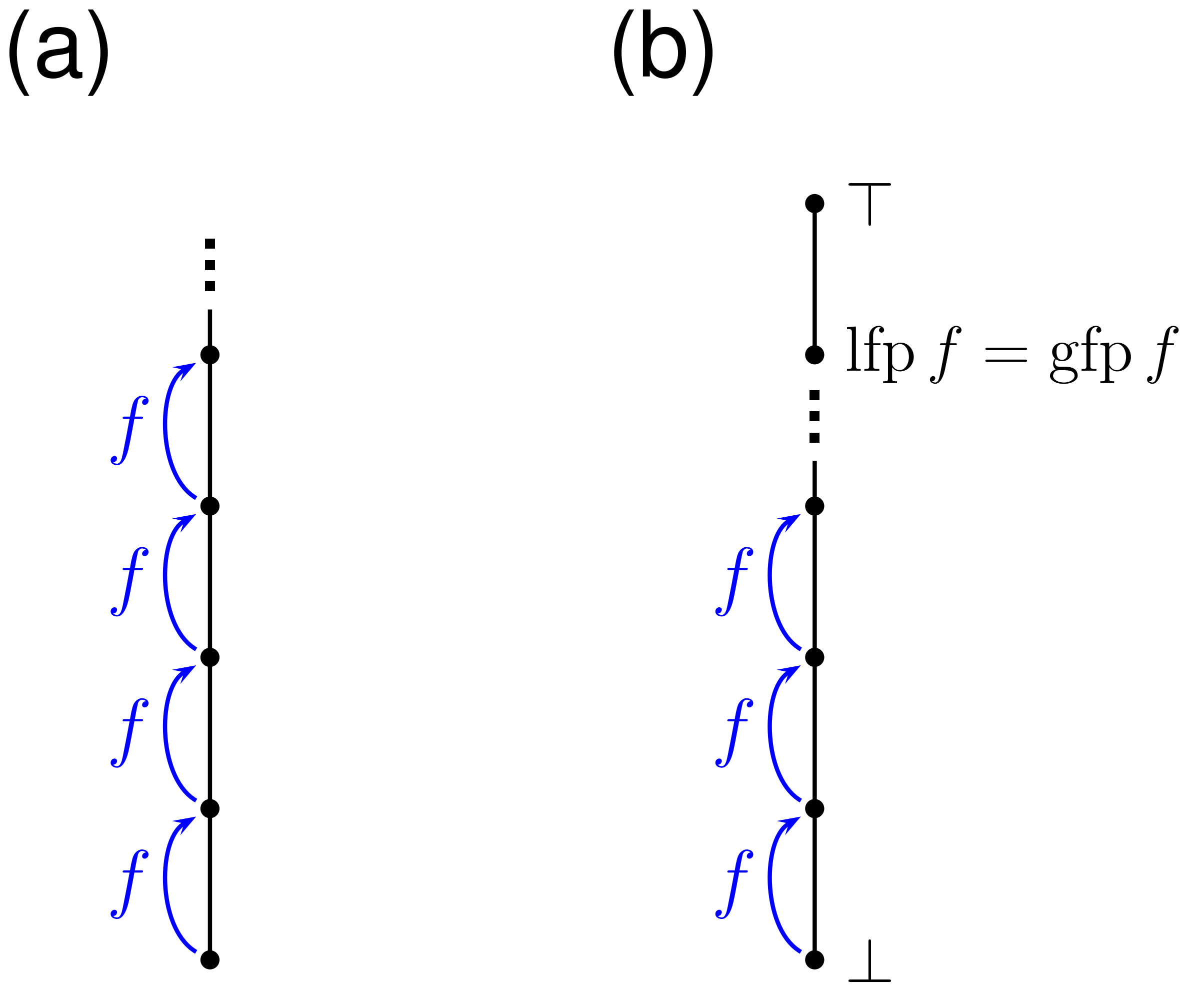
- $P = \{x \in L \mid x = f(x)\}$
- $\mathrm{lfp}\, f = \bigsqcap \{x \in L \mid f(x) \sqsubseteq x\}$
- $\mathrm{gfp}\, f = \bigsqcup \{x \in L \mid x \sqsubseteq f(x)\}$

Note: (1) $\mathrm{lfp}\, f$ is greatest lower bound of the set of post fixed points of f, and (2) $\mathrm{gfp}\, f$ is least upper bound of the set of pre fixed points of f.

Tarski's fixed point theorem, graphically



Fixed points, intuition



(a) On a poset a monotone function is not guaranteed to have a fixed point, (b) lfp and gfp may coincide, or (c) the fixed points may form a sub-lattice.

Galois connections, reloaded

Galois connection motivation

Partial orders model precision of properties: $a \sqsubseteq a'$ if the properties a and a' are *comparable* and a is *more precise* than a'.

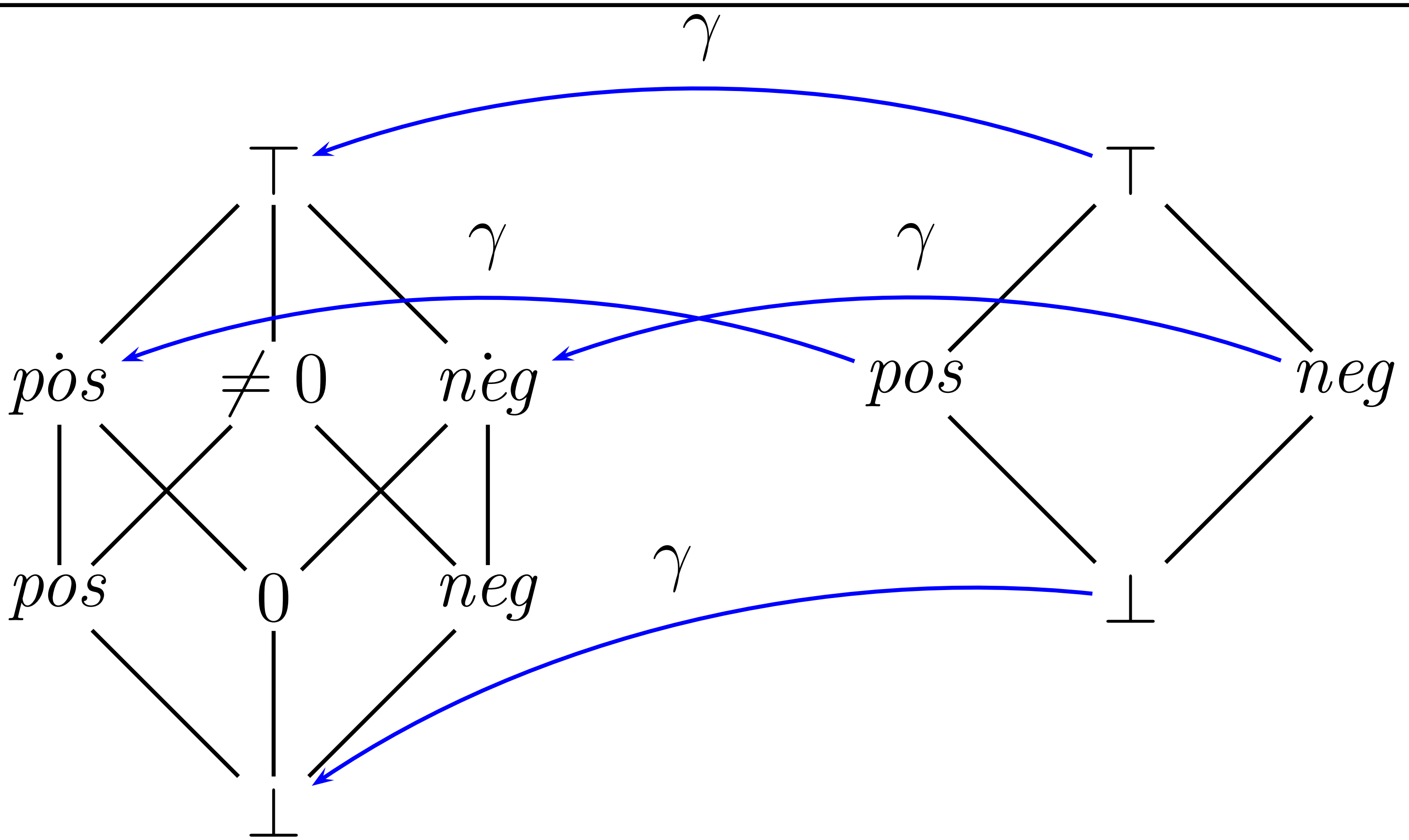
Example. *Recall from the Parity domain:*

The property $even$ *meaning* $\{n \in \mathbb{N}_0 \mid n \bmod 2 = 0\}$ *is more precise than the property* $\top$ *meaning* $\mathbb{N}_0$

The meaning of an abstract property is expressed by the concretization function γ.

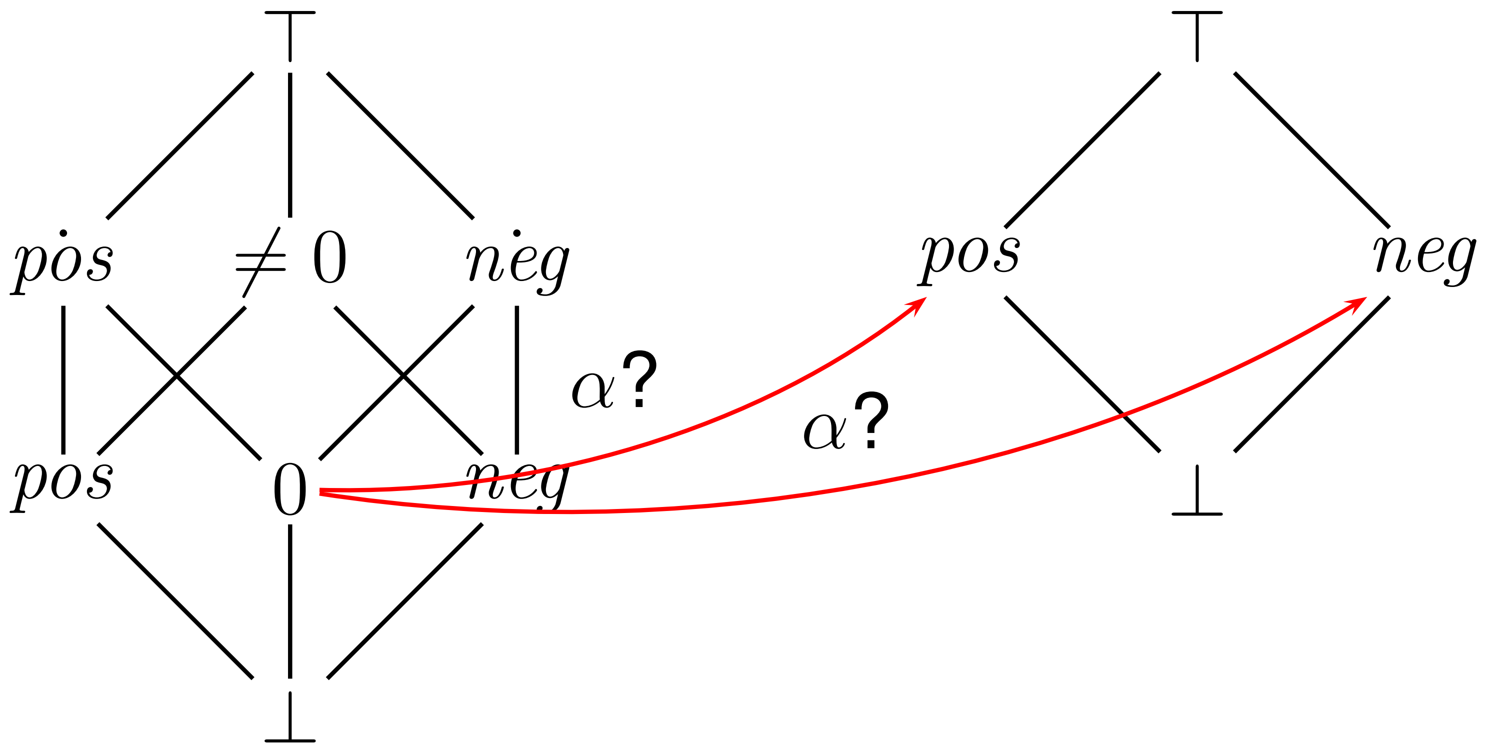
Approximation is captured by the abstraction function α: it maps each concrete property to its *best* abstract counterpart.

Galois connection non-example



γ assigns meaning to each abstract element.

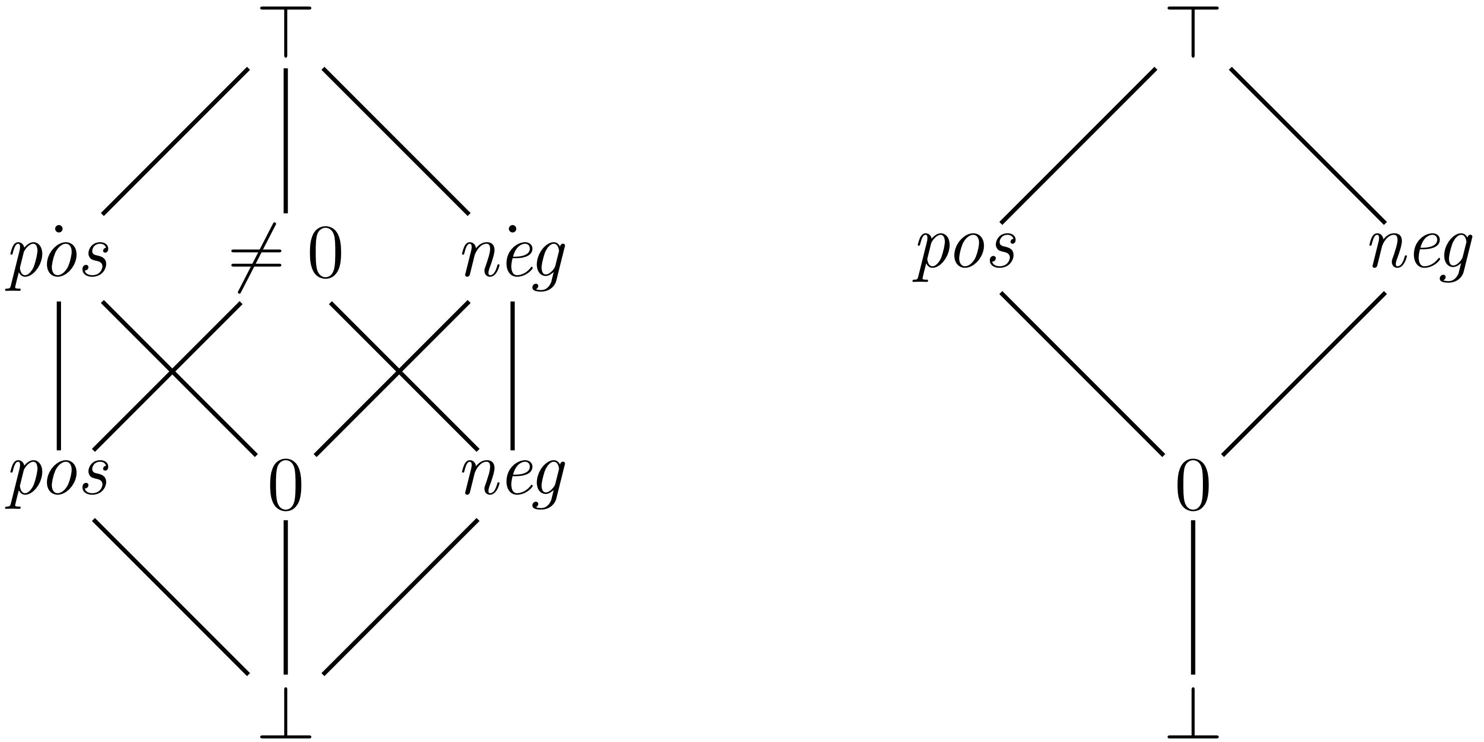
Galois connection non-example



γ assigns meaning to each abstract element.

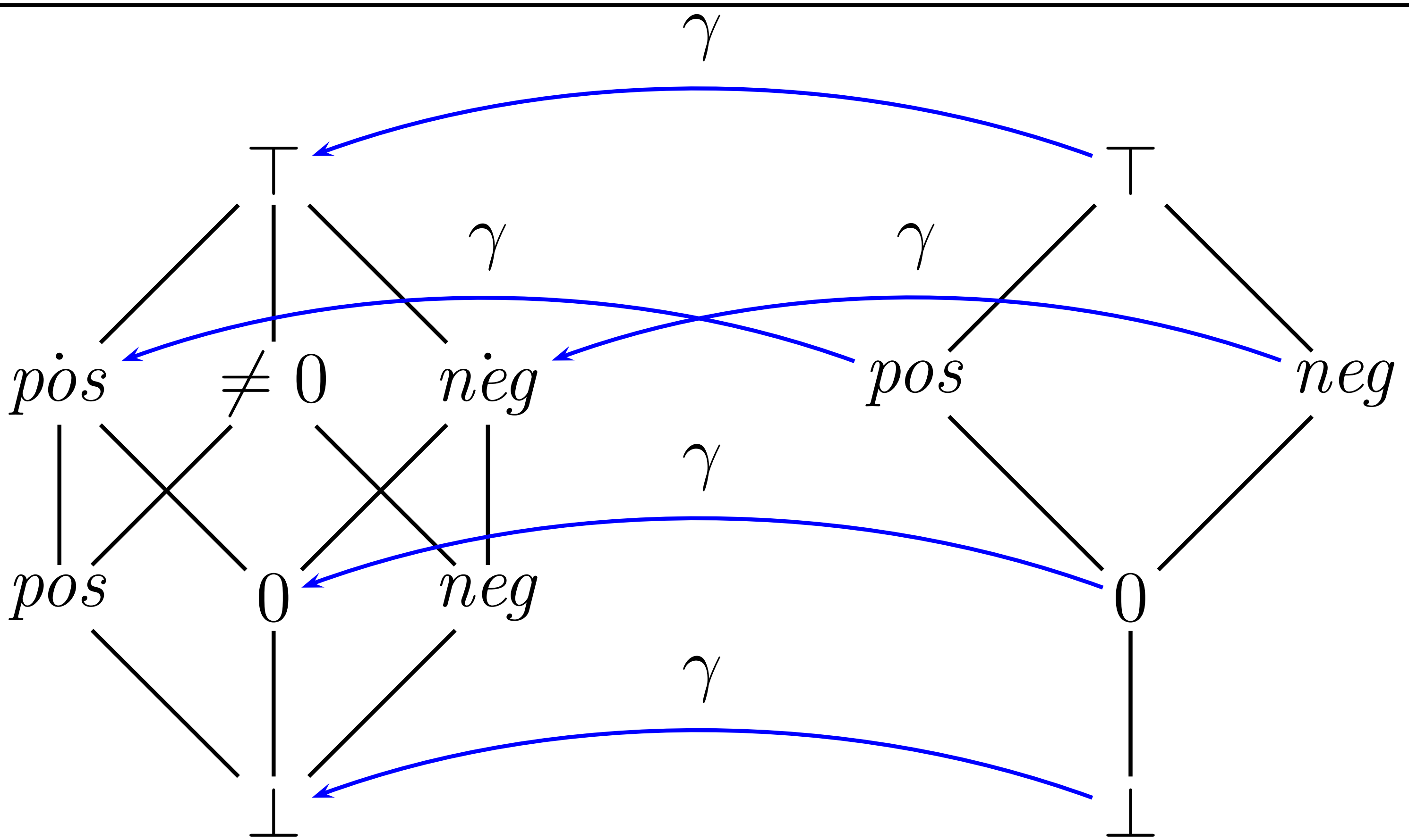
Problem: however there is no best (unique) abstraction for 0!

Galois connection example, fixed



We fix it by adding an element corresponding to 0.

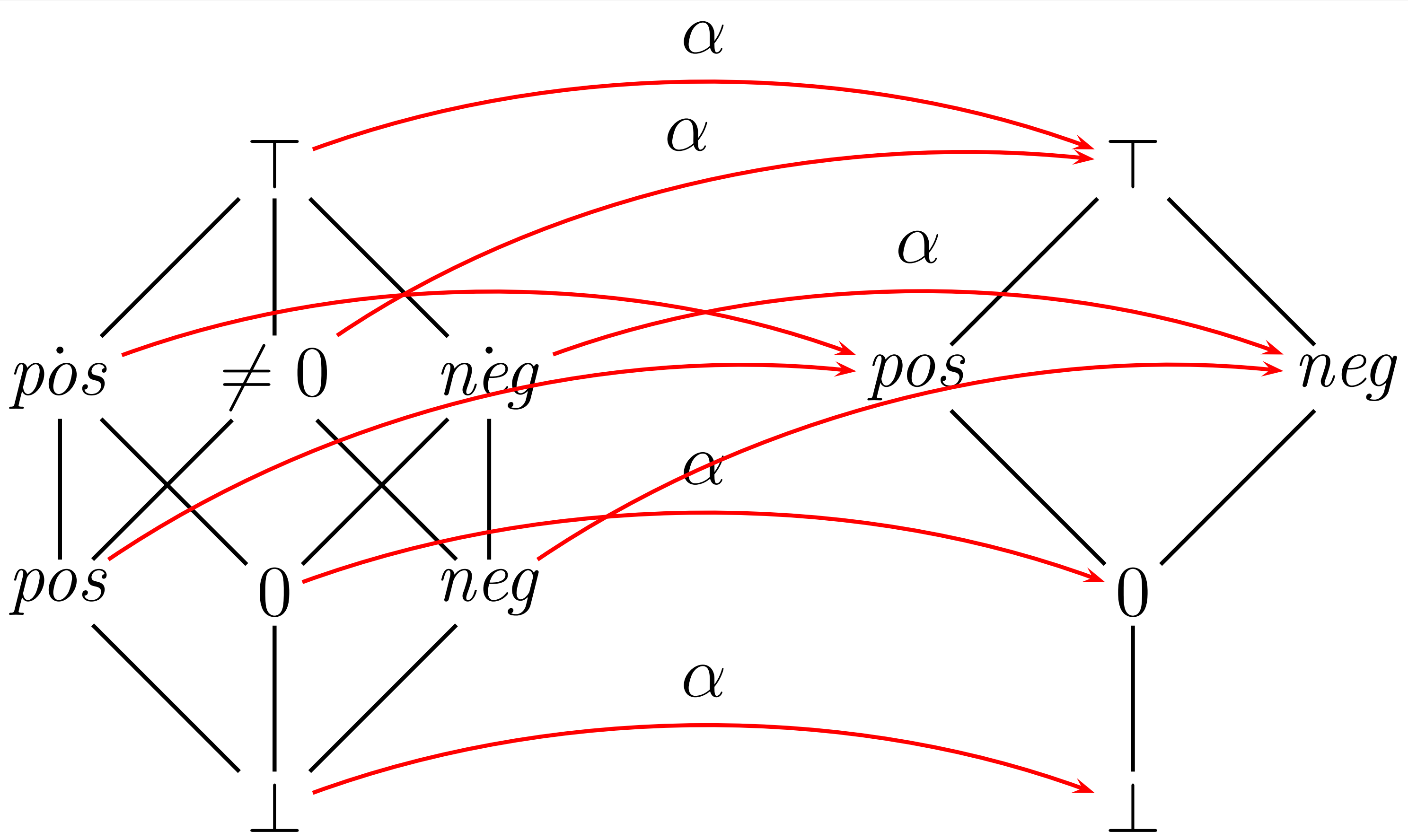
Galois connection example, fixed



γ assigns meaning to each abstract element.

Notice how γ is injective (one-to-one).

Galois connection example, fixed



α maps each element to its best abstraction.

Notice how α is surjective (onto), hence we have a Galois surjection.

Also notice the information loss.

Two soundness conditions

Condition 1:
If $a \leq a'$ for some c where $\alpha(c) = a$, then a' is a sound albeit less precise approximation of c.

Condition 2:
If $c' \sqsubseteq c$ for some a where $\gamma(a) = c$, then a is a sound albeit less precise approximation of c'.

When the two conditions are equivalent:

$$\alpha(c) \leq a' \iff c' \sqsubseteq \gamma(a)$$

we have a Galois connection.

Galois connection properties (1/2)

Observation 1: $\gamma \circ \alpha$ is extensive

Intuition: loss of information by α is sound

Observation 2: $\alpha \circ \gamma$ is reductive

Intuition: γ loses no information, i.e., α is as precise as possible

Observation 3: α and γ are monotone

Intuition: α and γ are order, i.e., soundness preserving

Galois connection properties (2/2)

Theorem. *The inverse of a Galois connection is itself a Galois connection (under reverse order):*

$$\frac{\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \leq \rangle}{\langle A; \geq \rangle \underset{\gamma}{\overset{\alpha}{\leftrightarrows}} \langle C; \sqsupseteq \rangle}$$

Galois connection properties (2/2)

Theorem. *The inverse of a Galois connection is itself a Galois connection (under reverse order):*

$$\frac{\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \leq \rangle}{\langle A; \geq \rangle \underset{\gamma}{\overset{\alpha}{\leftrightarrows}} \langle C; \sqsupseteq \rangle}$$

By the *duality principle* all results on posets have a dual.

Hence this extends to Galois connections if we replace

- $\sqsubseteq$, $\sqsubset$, $\bot$, $\top$, $\sqcap$, and $\sqcup$ with
- $\sqsupseteq$, $\sqsupset$, $\top$, $\bot$, $\sqcup$, and $\sqcap$

Alternative 1: Closure operators (1/3)

Definition. *A function* $\rho : S \to S$ *on a poset* $\langle S; \sqsubseteq \rangle$ *is a(n upper) closure operator if* ρ *is monotone, extensive, and idempotent:* $\forall s \in S : \rho(\rho(s)) = \rho(s)$

Similarly ρ *is a* lower *closure operator if it is monotone,* reductive*, and idempotent.*

Corollary. *A Galois connection* $\langle C; \sqsubseteq \rangle \xleftarrow[\alpha]{\gamma} \longrightarrow \langle A; \leq \rangle$ *induces*

- *an upper closure operator* $\gamma \circ \alpha$ *on* C *and*
- *a lower closure operator* $\alpha \circ \gamma$ *on* A

Alternative 1: Closure operators (2/3)

Theorem. *A closure operator* $\rho : S \to S$ *on a poset* $\langle S; \sqsubseteq \rangle$ *induces a Galois connection*

$$\langle S; \sqsubseteq \rangle \underset{\rho}{\overset{1}{\leftrightarrows}} \langle \rho(S); \sqsubseteq \rangle$$

(1 being the identity function on S).

Hence it is equivalent to stay in the concrete domain and formulate abstract interpretation in terms of closure operators!

Alternative 1: Closure operators (3/3)

$\rho = \alpha \circ \gamma$ is an example of a(n optimal) *reduction operator*: It normalizes an abstract element to its best abstraction.

Since $\rho = \alpha \circ \gamma$ is a lower closure operator, a static analysis can gain precision by applying it at well-chosen locations (before/after certain operations).

Why? Once we start lifting/composing simpler domains to form more complex ones, the result may contain redundant abstract elements.

Example: $\rho(\lambda pc.\ \langle even, \bot, odd \rangle) = \lambda pc.\ \langle \bot, \bot, \bot \rangle$ in the three counter machine.

However it may be too expensive to reduce everywhere.

Alternative 2: Moore families

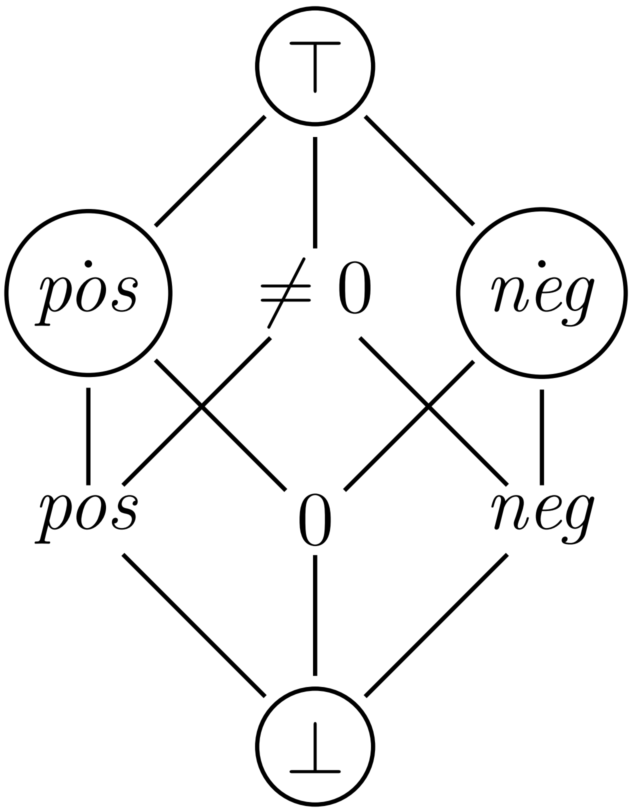
Definition. *Let* $\langle P; \sqsubseteq \rangle$ *be a poset with a top element* $\top$*. A* Moore family *is a subset* $S \subseteq P$ *such that*

- $\top \in S$
- *If* $X \subseteq S$ *then* $\sqcap X$ *exists in* P *and* $\sqcap X \in S$

Proposition. *If* $\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \leq \rangle$ *is a Galois connection and* $\langle C; \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$ *is a complete lattice, then* $\gamma(A) = \{\gamma(a) \mid a \in A\}$ *is a Moore family.*

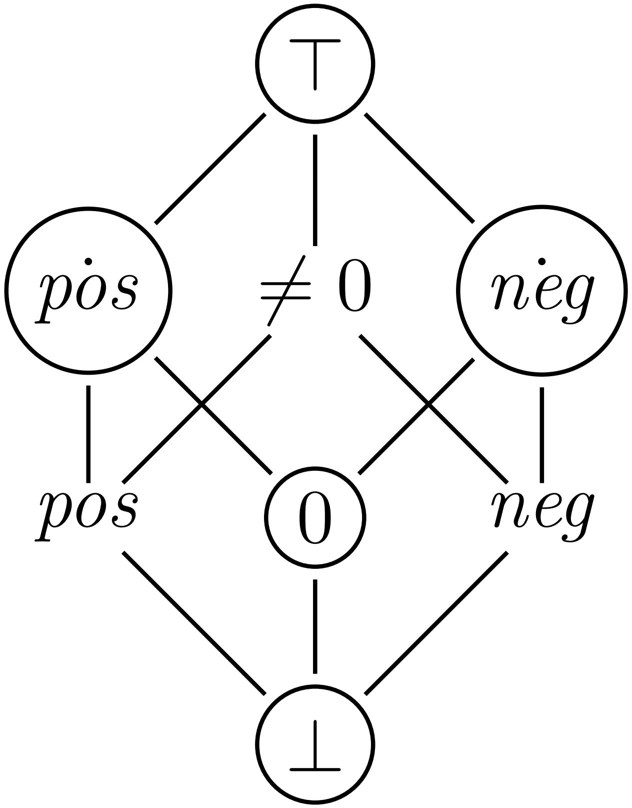
Hence, Moore families can provide a sanity check for an abstract domain.

Alternative 2: Moore family non-example



The greatest lower bound $\dot{pos} \sqcap \dot{neg}$ exists, but not in the above subset.

Alternative 2: Moore family example



The greatest lower bound $\dot{pos} \sqcap \dot{neg}$ exists, and belongs to the above subset.

More Galois connection properties

Each function uniquely determines the other:

Proposition. *If* $\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \leq \rangle$ *and* $\langle C; \sqsubseteq \rangle \underset{\alpha'}{\overset{\gamma'}{\leftrightarrows}} \langle A; \leq \rangle$ *then* $\alpha = \alpha'$ *if and only if* $\gamma = \gamma'$

Each function expresses the other:

Proposition. *If* $\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \leq \rangle$ *then*

- *for all* $c \in C : \alpha(c) = \bigwedge\{a \mid c \sqsubseteq \gamma(a)\}$
- *for all* $a \in A : \gamma(a) = \bigsqcup\{c \mid \alpha(c) \leq a\}$

Galois surjections and injections reloaded

Definition. *A* Galois surjection *(or insertion)* $\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftarrows\!\!\twoheadrightarrow}} \langle A; \leq \rangle$ *is a Galois connection where* α *is surjective (equivalently* γ *is injective, or* $\forall a \in A : \alpha \circ \gamma(a) = a$*).*

Definition. *A* Galois injection $\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\twoheadleftarrow\!\!\rightarrows}} \langle A; \leq \rangle$ *is a Galois connection in which* γ *is surjective (equivalently* α *is injective, or* $\forall c \in C : \gamma \circ \alpha(c) = c$*).*

Proposition. *If* $\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftarrows\!\!\twoheadrightarrow}} \langle A; \leq \rangle$ *is a Galois surjection and* C *is a complete lattice* $\langle C; \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$ *then* A *is a complete lattice.*

(Intuitively, we inherit least upper (greatest lower) bounds from the Galois connection)

Reduction of an abstract domain

By equating abstract elements with the same concretization, we obtain a Galois surjection:

Proposition. *If* $\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \leq \rangle$ *is a Galois connection, then*

- $a \equiv a' = (\gamma(a) = \gamma(a'))$ *is an equivalence relation, such that*
- $\langle C; \sqsubseteq \rangle \underset{\alpha_\equiv}{\overset{\gamma_\equiv}{\leftrightarrows}} \langle A/_\equiv; \leq_\equiv \rangle$ *is a Galois surjection,*

$$\textit{where } X \leq_\equiv Y \textit{ if } (\exists a \in X : \exists a' \in Y : a \leq a')$$
$$\alpha_\equiv(c) = \{a \mid a \equiv \alpha(c)\}$$
$$\gamma_\equiv(X) = \gamma(a) \;\; \textit{where} \;\; a \in X$$

Example: intervals

Consider the abstract domain of *intervals*.

Assume that elements are of the form $[a; b]$ where $a \in \mathbb{Z} \cup \{-\infty\}$ and $b \in \mathbb{Z} \cup \{\infty\}$

Ordering: $[a; b] \sqsubseteq [a'; b']$ if $a' \leq a \ \wedge \ b \leq b'$

Concretization: $\gamma([a; b]) = \{n \mid a \leq n \leq b\}$

All elements $[a; b]$ for which $a > b$ represent the empty set $\emptyset$ can be eliminated. Usually this reduction has already (implicitly) taken place.

For example, $\emptyset = \gamma([32; 0]) = \gamma([5; 4]) = \emptyset$

Compositional design of Galois connections

Known composition from day 1

Theorem. *The composition of two Galois connections* $\langle C; \sqsubseteq \rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle B; \subseteq \rangle$ *and* $\langle B; \subseteq \rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle A; \leq \rangle$ *is itself a Galois connection:*

$$\langle C; \sqsubseteq \rangle \underset{\alpha_2 \circ \alpha_1}{\overset{\gamma_1 \circ \gamma_2}{\leftrightarrows}} \langle A; \leq \rangle$$

The above theorem typeset as an inference rule:

$$\frac{\langle C; \sqsubseteq \rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle B; \subseteq \rangle \qquad \langle B; \subseteq \rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle A; \leq \rangle}{\langle C; \sqsubseteq \rangle \underset{\alpha_2 \circ \alpha_1}{\overset{\gamma_1 \circ \gamma_2}{\leftrightarrows}} \langle A; \leq \rangle}$$

The Cartesian product of Galois connections

Theorem. *Let* $\langle C_1; \sqsubseteq_1 \rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle A_1; \leq_1 \rangle$ *and* $\langle C_2; \sqsubseteq_2 \rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle A_2; \leq_2 \rangle$ *be Galois connections. Then we can form a Galois connection between the Cartesian product of the concrete and abstract domains:*

$$\langle C_1 \times C_2; \sqsubseteq_1 \times \sqsubseteq_2 \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A_1 \times A_2; \leq_1 \times \leq_2 \rangle$$

where

$$\alpha(\langle c_1, c_2 \rangle) = \langle \alpha_1(c_1), \alpha_2(c_2) \rangle$$
$$\gamma(\langle a_1, a_2 \rangle) = \langle \gamma_1(a_1), \gamma_2(a_2) \rangle$$

The Cartesian product of Galois connections

Theorem. *(same, now typeset as inference rule)*

$$\frac{\langle C_1;\sqsubseteq_1\rangle \xrightleftharpoons[\alpha_1]{\gamma_1} \langle A_1;\leq_1\rangle \qquad \langle C_2;\sqsubseteq_2\rangle \xrightleftharpoons[\alpha_2]{\gamma_2} \langle A_2;\leq_2\rangle}{\langle C_1\times C_2;\sqsubseteq_1\times\sqsubseteq_2\rangle \xrightleftharpoons[\alpha]{\gamma} \langle A_1\times A_2;\leq_1\times\leq_2\rangle}$$

where

$$\alpha(\langle c_1,\, c_2\rangle) = \langle \alpha_1(c_1),\, \alpha_2(c_2)\rangle$$
$$\gamma(\langle a_1,\, a_2\rangle) = \langle \gamma_1(a_1),\, \gamma_2(a_2)\rangle$$

The Cartesian product of Galois connections

Theorem. *(same, now typeset as inference rule)*

$$\frac{\langle C_1; \sqsubseteq_1\rangle \xleftrightarrow[\alpha_1]{\gamma_1} \langle A_1; \leq_1\rangle \qquad \langle C_2; \sqsubseteq_2\rangle \xleftrightarrow[\alpha_2]{\gamma_2} \langle A_2; \leq_2\rangle}{\langle C_1 \times C_2; \sqsubseteq_1 \times \sqsubseteq_2\rangle \xleftrightarrow[\alpha]{\gamma} \langle A_1 \times A_2; \leq_1 \times \leq_2\rangle}$$

where

$$\alpha(\langle c_1,\, c_2\rangle) = \langle \alpha_1(c_1),\, \alpha_2(c_2)\rangle$$
$$\gamma(\langle a_1,\, a_2\rangle) = \langle \gamma_1(a_1),\, \gamma_2(a_2)\rangle$$

Example: we can abstract a pair of natural number sets to a Parity pair:

$$\frac{\langle \wp(\mathbb{N}_0); \subseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle Par; \sqsubseteq\rangle \qquad \langle \wp(\mathbb{N}_0); \subseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle Par; \sqsubseteq\rangle}{\langle \wp(\mathbb{N}_0) \times \wp(\mathbb{N}_0); \subseteq \times \subseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle Par \times Par; \sqsubseteq \times \sqsubseteq\rangle}$$

Reduced product

A *reduced product* improves two (or more) abstractions of the same domain:

Theorem. *Let* $\langle C; \sqsubseteq \rangle \xleftrightarrow[\alpha_1]{\gamma_1} \langle A_1; \leq_1 \rangle$ *and* $\langle C; \sqsubseteq \rangle \xleftrightarrow[\alpha_2]{\gamma_2} \langle A_2; \leq_2 \rangle$ *be Galois connections between complete lattices. Then the reduced product is a Galois surjection:*

$$\langle C; \sqsubseteq \rangle \xleftrightarrow[\alpha]{\gamma} \langle A_1 \times A_2; \leq_1 \times \leq_2 \rangle$$

$$\textit{where} \quad \alpha(c) = \langle \alpha_1(c),\ \alpha_2(c) \rangle$$
$$\gamma(\langle a_1,\ a_2 \rangle) = \gamma_1(a_1) \sqcap \gamma_2(a_2)$$

Note: the paper contains a much more general version

Example: reduced product

Imagine we abstract an integer variable x using both *Sign* and *Parity* abstract domains.

If $\mathrm{x} = 0$ from the Sign domain ($\gamma(0) = \{0\}$) and x is odd from the Parity domain ($\gamma(odd) = \{1, 3, 5, \ldots\}$), we gain information by combining it.

A reduction tells us, no integers are 0 *and* odd, hence we reduce to $\gamma(0) \cap \gamma(odd) = \emptyset$.

Note: Not transferring information from one domain to the other corresponds to running the analyses separately.

Partitioning

Definition. *Let* L *be a set of labels. A* partition *of a complete lattice* $\langle C; \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$ *is a function* $\delta : L \to C$ *that (a) covers C:* $\top = \sqcup_{l \in L} \delta(l)$*, and (b) is disjoint:* $\forall \ell, \ell' \in L : \ell \neq \ell' \implies \delta(\ell) \sqcap \delta(\ell') = \bot$

Proposition. *Let* $\delta : L \to C$ *be a partition of a complete lattice* $\langle C; \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$*. Then the abstract domain* $A = \Pi_{\ell \in L} \{ c \sqcap \delta(\ell) \mid c \in C \}$ *ordered componentwise* $a \leq a' \iff \forall \ell \in L : a(\ell) \sqsubseteq a'(\ell)$ *forms a Galois connection:*

$$\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \leq \rangle$$

where $\alpha(c) = \lambda \ell.\, c \sqcap \delta(\ell) \qquad \gamma(a) = \bigsqcup_{\ell \in L} a(\ell)$

By reducing the domain we can obtain a Galois surj.

Example: partitioning

Intuitively, we divide a set into a number of regions:

```
  _______              _______
 /       \___         /    /  \___
/            \       /  1 /   /   \
|            |  <--  |   / 2 /.../|
|            |       |  /\  /\ _/ |
\_           /  -->  \_/  \/  \.../
  \         /          \ 3 \ 4 \ /
   \___ ___/            \___\___/
```

For example, the first abstraction of the 3 counter machine collecting semantics, groups quadruples with same $\texttt{pc}$: $L = PC$

$$\delta(pc) = \{\langle pc,\ xv,\ yv,\ zv\rangle \mid xv \in \mathbb{N}_0, yv \in \mathbb{N}_0, zv \in \mathbb{N}_0\}$$

$$\wp(PC \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} PC \to \wp(PC \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0)$$

From concrete to abstract semantics

Correctness, optimality, and completeness

Definition. *If* $\alpha \circ F \mathrel{\dot{\leq}} F^{\#} \circ \alpha$ *we say* $F^{\#}$ *is a (locally) correct (or sound) approximation of* F

Definition. *If* $F^{\#} = \alpha \circ F \circ \gamma$ *we say* $F^{\#}$ *is an optimal approximation of* F

Intuitively we can't do better with the available abstract information.

Definition. *If* $\alpha \circ F = F^{\#} \circ \alpha$ *we say* $F^{\#}$ *is a complete approximation of* F *(no loss of information)*

Intuitively we can't do better with the available concrete information.

These definitions generalize to n-ary functions F and $F^{\#}$.

Example

Consider abstract addition ($\widehat{+}$) over the Sign domain.

Addition is not complete, e.g.:

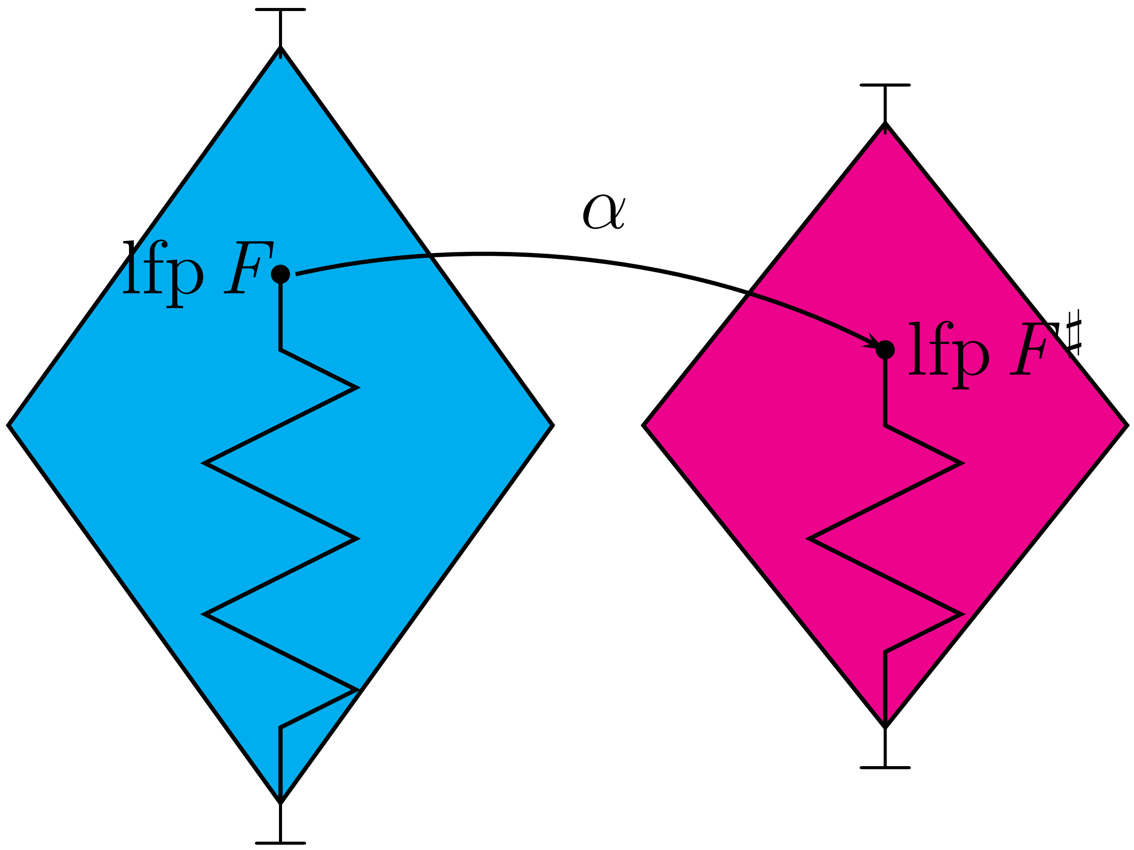
$$
\begin{aligned}
0 &= \alpha(42 + (-42)) \\
&\sqsubseteq \alpha(42) \mathbin{\widehat{+}} \alpha(-42) = pos \mathbin{\widehat{+}} neg = \top
\end{aligned}
$$

However addition is an optimal approximation, e.g.:

$$
\begin{aligned}
&\alpha(\gamma(pos) + \gamma(neg)) \\
&= \alpha(\{n \mid n \geq 0\} + \{n \mid n \leq 0\}) \\
&= \alpha(\{n + n' \mid n \geq 0 \ \wedge \ n' \leq 0\}) \\
&= \alpha(\mathbb{Z}) = \top
\end{aligned}
$$

Joy of completeness (Cousot-Cousot:POPL79)

By the *stronger fixed-point transfer theorem* we can compute a direct abstraction of the collecting semantics:



Theorem. *Let* $\langle C; \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \leq \rangle$ *be a Galois connection between complete lattices. If* F *and* $F^\sharp$ *are monotone and* $\alpha \circ F = F^\sharp \circ \alpha$ *then* $\alpha(\text{lfp}\, F) = \text{lfp}\, F^\sharp$

From concrete to abstract operator, constructively

These definitions lead us to the following two "recipes" for approximating a concrete operator F:

1. Push α's under the function definition:

   $$\alpha \circ F(c) = \cdots = F^{\#}(\alpha(c))$$

   (geared towards complete approximation, however it is still correct/sound if we upward judge underway)

2. Compose F with α and γ:

   $$\alpha \circ F \circ \gamma(a) = \cdots = F^{\#}(a)$$

   (geared towards optimal approximation, however it is still correct/sound if we upward judge underway)

The art of calculation...

"We habitually use this proposition constructively in order to derive the abstract semantics from the definition of the concrete semantics: for the basis we simply let [$\perp^\sharp$] be [$\alpha(\perp)$]. For the semantic function [$F^\sharp$] starting from the term $\alpha(F(c))$ we replace α and F by their definitions and then simplify the expression in order to let the term $\alpha(c)$ come out, in which case we let the resulting expression (where $\alpha(c)$ is replaced by a) be the definition of [$F^\sharp(a)$]."

– Cousot-Cousot:JLC92

More fun with the three counter machine

Previously: analysing the 3 counter machine

```
  Var ::= x  |  y  |  z
 Inst ::= inc var  |  dec var  |  zero var m else n  |  stop
States  =  PC x \N_0 x \N_0 x \N_0
```

Transition relation:

```
<pc, xv, yv, zv> --> <pc+1, xv+1, yv, zv>                  if P_pc = inc x
        -        --> <pc+1, xv, yv+1, zv>                  if P_pc = inc y
        -        --> <pc+1, xv, yv, zv+1>                  if P_pc = inc z

<pc, xv, yv, zv> --> <pc+1, xv-1, yv, zv>          if P_pc = dec x /\ xv>0
        -        --> <pc+1, xv, yv-1, zv>          if P_pc = dec y /\ yv>0
        -        --> <pc+1, xv, yv, zv-1>          if P_pc = dec z /\ zv>0

<pc, xv, yv, zv> --> <pc', xv, yv, zv>      if P_pc = zero x pc' else pc''
                                               /\ xv=0
        -        --> <pc'', xv, yv, zv>     if P_pc = zero x pc' else pc''
                                               /\ xv<>0

<pc, xv, yv, zv> --> <pc', xv, yv, zv>      if P_pc = zero y pc' else pc''
                                               /\ yv=0
        -        --> <pc'', xv, yv, zv>     if P_pc = zero y pc' else pc''
                                               /\ yv<>0

<pc, xv, yv, zv> --> <pc', xv, yv, zv>      if P_pc = zero z pc' else pc''
                                               /\ zv=0
        -        --> <pc'', xv, yv, zv>     if P_pc = zero z pc' else pc''
                                               /\ zv<>0
```

We left off here:

```
F#(S#) =  Ø. [1 -> { <i,0,0> | i in N_0 }]

  U.
            U.         Ø. [pc+1 -> { <xv+1, yv, zv> }]
   { <xv, yv, zv> } C S#(pc)
      P_pc = inc x                (...and for y and z)
  U.
            U.         Ø. [pc+1 -> { <xv-1, yv, zv> }]
   { <xv, yv, zv> } C S#(pc)
      P_pc = dec x
         xv>0                     (...and for y and z)
  U.
            U.         Ø. [pc' -> { <xv, yv, zv> }]
   { <xv, yv, zv> } C S#(pc)
  P_pc = zero x pc' else pc''
          xv=0                    (...and for y and z)
  U.
            U.         Ø. [pc'' -> { <xv, yv, zv> }]
   { <xv, yv, zv> } C S#(pc)
  P_pc = zero x pc' else pc''
         xv<>0                    (...and for y and z)
```

Call-by-need Galois connections :-) (1/3)

Abstracting a set valued function:

Given a Galois connection between complete lattices, we can lift it pointwise to function spaces (also complete lattices):

$$\frac{\langle \wp(C); \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A; \sqsubseteq \rangle}{\langle D \to \wp(C); \dot{\subseteq} \rangle \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} \langle D \to A; \dot{\sqsubseteq} \rangle}$$

where $\dot{\alpha}(F) = \lambda d.\, \alpha(F(d))$

$\dot{\gamma}(F^{\#}) = \lambda d.\, \gamma(F^{\#}(d))$

Call-by-need Galois connections :-) (2/3)

Abstracting a set of triples by a triple of sets:

$$\dfrac{}{\langle \wp(A \times B \times C); \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(A) \times \wp(B) \times \wp(C); \subseteq_\times\rangle}$$

between complete lattices (the latter being reduced) where

$$\begin{aligned} \subseteq_\times &= \ \subseteq \times \subseteq \times \subseteq \\ \alpha(T) &= \langle \pi_1(T),\ \pi_2(T),\ \pi_3(T)\rangle \\ \gamma(\langle X,\ Y,\ Z\rangle) &= X \times Y \times Z \end{aligned}$$

Call-by-need Galois connections :-) (3/3)

Abstracting a triple of sets by an abstract triple:

Given three Galois connections between complete lattices, we can form a new Galois connection (also over complete lattices):

$$\frac{\langle \wp(A); \subseteq\rangle \underset{\alpha_A}{\overset{\gamma_A}{\leftrightarrows}} \langle A'; \sqsubseteq_a\rangle \qquad \langle \wp(B); \subseteq\rangle \underset{\alpha_B}{\overset{\gamma_B}{\leftrightarrows}} \langle B'; \sqsubseteq_b\rangle \qquad \langle \wp(C); \subseteq\rangle \underset{\alpha_C}{\overset{\gamma_C}{\leftrightarrows}} \langle C'; \sqsubseteq_c\rangle}{\langle \wp(A) \times \wp(B) \times \wp(C); \subseteq_\times\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A' \times B' \times C'; \sqsubseteq_\times\rangle}$$

where

$$\begin{aligned}
\subseteq_\times &= \subseteq \times \subseteq \times \subseteq \\
\sqsubseteq_\times &= \sqsubseteq_a \times \sqsubseteq_b \times \sqsubseteq_c \\
\alpha(\langle X, Y, Z\rangle) &= \langle \alpha_A(X), \alpha_B(Y), \alpha_C(Z)\rangle \\
\gamma(\langle X', Y', Z'\rangle) &= \langle \gamma_A(X), \gamma_B(Y), \gamma_C(Z)\rangle
\end{aligned}$$

Three counter analysis from 10000 feet[1]

The Parity analysis is composed in two.

Yesterday:

$$\frac{}{\wp(PC \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0) \leftrightarrows PC \to \wp(\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0)}$$

Today:

$$\frac{\dfrac{\dfrac{}{\wp(\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0) \leftrightarrows \wp(\mathbb{N}_0) \times \wp(\mathbb{N}_0) \times \wp(\mathbb{N}_0)} \qquad \dfrac{\dfrac{}{\wp(\mathbb{N}_0) \mathrel{\substack{\longleftarrow \\ \twoheadrightarrow}} Par} \quad \dfrac{}{\wp(\mathbb{N}_0) \mathrel{\substack{\longleftarrow \\ \twoheadrightarrow}} Par} \quad \dfrac{}{\wp(\mathbb{N}_0) \mathrel{\substack{\longleftarrow \\ \twoheadrightarrow}} Par}}{\wp(\mathbb{N}_0) \times \wp(\mathbb{N}_0) \times \wp(\mathbb{N}_0) \leftrightarrows Par \times Par \times Par}}{\wp(\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0) \leftrightarrows Par \times Par \times Par}}{PC \to \wp(\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0) \leftrightarrows PC \to Par \times Par \times Par}$$

Hence by transitivity:

$$\frac{}{\wp(PC \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0) \leftrightarrows PC \to Par \times Par \times Par}$$

1. and therefore in a very small font

At home: operators/property transformers

Yesterday you calculated abstract operators:

```
 =0 : Parity -> Parity
<>0 : Parity -> Parity
 +1 : Parity -> Parity
 -1 : Parity -> Parity
```

from concrete ones over $\wp(N_0)$:

```
 =0 : P(N0) -> P(N0)
    = \S. {s | s in S /\ s=0 }
<>0 : P(N0) -> P(N0)
    = \S. {s | s in S /\ s<>0 }
 +1 : P(N0) -> P(N0)
    = \S. {s+1 | s in S}
 -1 : P(N0) -> P(N0)
    = \S. {s-1 | s in S /\ s>0 }
```

Result

```
\S#.

    <bot,bot,bot>.[ 1 -> <top, even, even> ]

      U.
          U.      <bot,bot,bot>.[ pc+1 -> [var++]#(S#(pc)) ]
   P_pc = inc var

      U.
          U.      <bot,bot,bot>.[ pc+1 -> [var--]#(S#(pc)) ]
   P_pc = dec var

      U.
                  <bot,bot,bot>.[ pc' -> [var=0](S#(pc)) ]
          U.        U. <bot,bot,bot>.[ pc'' -> [var<>0](S#(pc))]
P_pc = zero var pc' else pc''
```

Summary

Summary

We've taken a more in depth look at AI based on Cousot-Cousot:JLP92.

- □ Foundations: Fixed points, Galois connections, ...
- □ The Galois approach and friends: closure operators, Moore families, ...
- □ From collecting semantics to analysis

+ analysis of Plotkin's three counter machine