Principles of Program Analysis:

Abstract Interpretation

Transparencies based on Chapter 4 of the book: Flemming Nielson, Hanne Riis Nielson and Chris Hankin: Principles of Program Analysis. Springer Verlag 2004.

A Mundane Approach to Semantic Correctness

Semantics:

$$p \vdash v_1 \rightsquigarrow v_2$$

where $v_1, v_2 \in V$.

Note: $\rightsquigarrow$ might be deterministic.

Program analysis:

$$p \vdash l_1 \rhd l_2$$

where $l_1, l_2 \in L$.

Note: $\rhd$ should be deterministic:

$$f_p(l_1) = l_2.$$

What is the relationship between the semantics and the analysis?

Restrict attention to analyses where properties directly describe sets of values i.e. *"first-order"" analyses* (rather than *"second-order" analyses*).

Example: Data Flow Analysis

Structural Operational Semantics:

Values: $V = \mathbf{State}$

Transitions:

$$S_\star \vdash \sigma_1 \leadsto \sigma_2$$

iff

$$\langle S_\star, \sigma_1 \rangle \to^* \sigma_2$$

Constant Propagation Analysis:

Properties: $L = \widehat{\mathbf{State}}_{\mathsf{CP}} = (\mathbf{Var}_\star \to \mathbf{Z}^\top)_\bot$

Transitions:

$$S_\star \vdash \widehat{\sigma}_1 \rhd \widehat{\sigma}_2$$

iff

$$\widehat{\sigma}_1 = \iota$$
$$\widehat{\sigma}_2 = \bigsqcup\{\mathsf{CP}_\bullet(\ell) \mid \ell \in \mathit{final}(S_\star)\}$$
$$(\mathsf{CP}_\circ, \mathsf{CP}_\bullet) \models \mathsf{CP}^=(S_\star)$$

Example: Control Flow Analysis

Structural Operational Semantics:

Values: $V = \mathbf{Val}$

Transitions:

$$e_\star \vdash v_1 \rightsquigarrow v_2$$

iff

$$[\,] \vdash (e_\star\ v_1^{\ell_1})^{\ell_2} \longrightarrow^* v_2^{\ell_2}$$

Pure 0-CFA Analysis:

Properties: $L = \widehat{\mathbf{Env}} \times \widehat{\mathbf{Val}}$

Transitions:

$$e_\star \vdash (\widehat{\rho}_1, \widehat{v}_1) \rhd (\widehat{\rho}_2, \widehat{v}_2)$$

iff

$$\widehat{\mathsf{C}}(\ell_1) = \widehat{v}_1$$
$$\widehat{\mathsf{C}}(\ell_2) = \widehat{v}_2$$
$$\widehat{\rho}_1 = \widehat{\rho}_2 = \widehat{\rho}$$
$$(\widehat{\mathsf{C}}, \widehat{\rho}) \models (e_\star\ \mathsf{c}^{\ell_1})^{\ell_2}$$

for some place holder constant c

Correctness Relations

$$R : V \times L \rightarrow \{\mathit{true}, \mathit{false}\}$$

Idea: $v \; R \; l$ means that the value v is described by the property l.

Correctness criterion: R is preserved under computation:

$$\begin{array}{cccccc} p & \vdash & v_1 & \leadsto & v_2 \\ & & \vdots & & \vdots \\ & & R & \Rightarrow & R \\ & & \vdots & & \vdots \\ p & \vdash & l_1 & \rhd & l_2 \end{array}$$

logical relation:

$$(p \vdash \cdot \leadsto \cdot) \; (R \twoheadrightarrow R) \; (p \vdash \cdot \rhd \cdot)$$

Admissible Correctness Relations

$$v \; R \; l_1 \;\wedge\; l_1 \sqsubseteq l_2 \;\Rightarrow\; v \; R \; l_2$$

$$(\forall l \in L' \subseteq L : v \; R \; l) \;\Rightarrow\; v \; R \; (\bigsqcap L') \qquad (\{l \mid v \; R \; l\} \text{ is a Moore family})$$

Two consequences:

$$v \; R \; \top$$

$$v \; R \; l_1 \;\wedge\; v \; R \; l_2 \;\Rightarrow\; v \; R \; (l_1 \sqcap l_2)$$

Assumption: $(L, \sqsubseteq)$ is a complete lattice.

Example: Data Flow Analysis

Correctness relation

$$R_{\mathsf{CP}} : \mathbf{State} \times \widehat{\mathbf{State}}_{\mathsf{CP}} \to \{\mathit{true}, \mathit{false}\}$$

is defined by

$$\sigma \ R_{\mathsf{CP}} \ \widehat{\sigma} \quad \text{iff} \quad \forall x \in \mathit{FV}(S_\star) : (\widehat{\sigma}(x) = \top \ \vee \ \sigma(x) = \widehat{\sigma}(x))$$

Example: Control Flow Analysis

Correctness relation

$$R_{\mathsf{CFA}} : \mathbf{Val} \times (\widehat{\mathbf{Env}} \times \widehat{\mathbf{Val}}) \to \{\mathit{true}, \mathit{false}\}$$

is defined by

$$v \; R_{\mathsf{CFA}} \; (\widehat{\rho}, \widehat{v}) \quad \text{iff} \quad v \; \mathcal{V} \; (\widehat{\rho}, \widehat{v})$$

where $\mathcal{V}$ is given by:

$$v \; \mathcal{V} \; (\widehat{\rho}, \widehat{v}) \text{ iff } \begin{cases} \mathit{true} & \text{if } v = c \\ t \in \widehat{v} \wedge \forall x \in \mathit{dom}(\rho) : \rho(x) \; \mathcal{V} \; (\widehat{\rho}, \widehat{\rho}(x)) & \text{if } v = \texttt{close } t \texttt{ in } \rho \end{cases}$$

Representation Functions

$$\beta : V \to L$$

Idea: β maps a value to the *best* property describing it.

Correctness criterion:

$$\begin{array}{ccccc}
p & \vdash & v_1 & \leadsto & v_2 \\
 & \beta & \downarrow & \Rightarrow & \downarrow \;\beta \\
 & & \sqsubseteq & & \sqsubseteq \\
p & \vdash & l_1 & \rhd & l_2
\end{array}$$

Equivalence of Correctness Criteria

Given a representation function β we define a correctness relation R_β by

$$v \; R_\beta \; l \quad \text{iff} \quad \beta(v) \sqsubseteq l$$

Given a correctness relation R we define a representation function β_R by

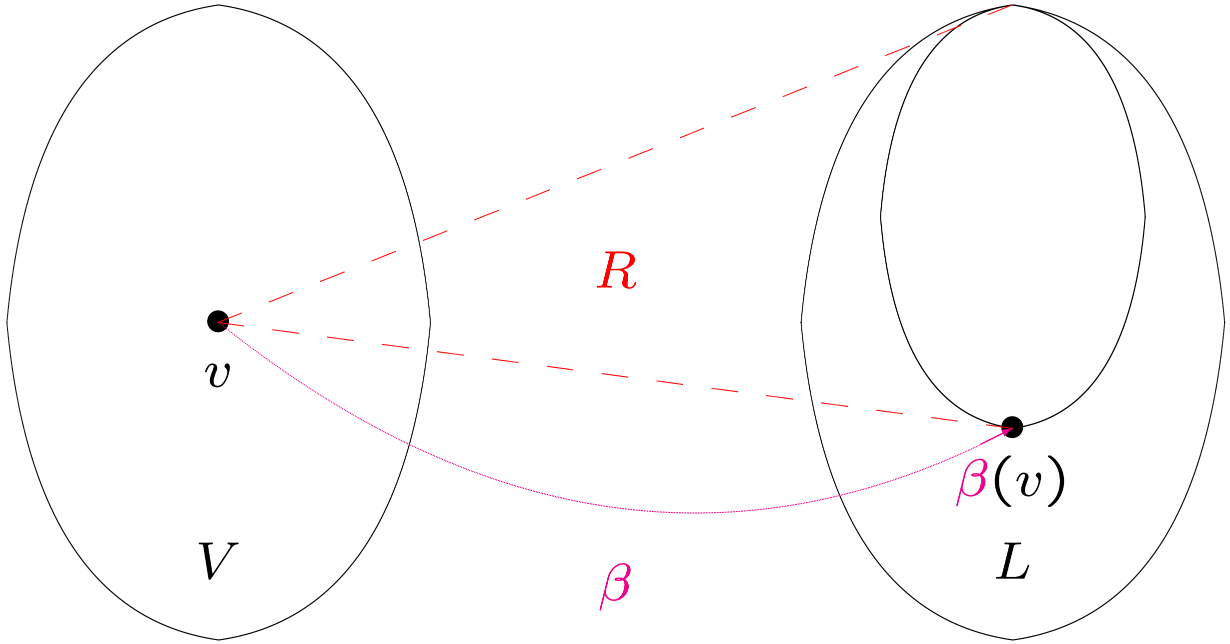
$$\beta_R(v) \;=\; \bigsqcap\{l \mid v \; R \; l\}$$

Lemma:

(i) Given $\beta : V \to L$, then the relation $R_\beta : V \times L \to \{\mathit{true}, \mathit{false}\}$ is an admissible correctness relation such that $\beta_{R_\beta} = \beta$.

(ii) Given an admissible correctness relation $R : V \times L \to \{\mathit{true}, \mathit{false}\}$, then β_R is well-defined and $R_{\beta_R} = R$.

Equivalence of Criteria: R is *generated* by β



Example: Data Flow Analysis

Representation function

$$\beta_{\mathsf{CP}} : \mathbf{State} \to \widehat{\mathbf{State}}_{\mathsf{CP}}$$

is defined by

$$\beta_{\mathsf{CP}}(\sigma) = \lambda x.\sigma(x)$$

R_{CP} is *generated by* β_{CP}:

$$\sigma \; R_{\mathsf{CP}} \; \widehat{\sigma} \quad \underline{\text{iff}} \quad \beta_{\mathsf{CP}}(\sigma) \sqsubseteq_{\mathsf{CP}} \widehat{\sigma}$$

Example: Control Flow Analysis

Representation function

$$\beta_{\mathsf{CFA}} : \mathbf{Val} \to \widehat{\mathbf{Env}} \times \widehat{\mathbf{Val}}$$

is defined by

$$\beta_{\mathsf{CFA}}(v) = \begin{cases} (\lambda x.\emptyset, \emptyset) & \text{if } v = c \\ (\beta^{E}_{\mathsf{CFA}}(\rho), \{t\}) & \text{if } v = \texttt{close}\ t\ \texttt{in}\ \rho \end{cases}$$

$$\begin{aligned} \beta^{E}_{\mathsf{CFA}}(\rho)(x) \;=\; & \bigcup\{\widehat{\rho}_y(x) \mid \beta_{\mathsf{CFA}}(\rho(y)) = (\widehat{\rho}_y, \widehat{v}_y) \text{ and } y \in \mathit{dom}(\rho)\} \\ & \cup \begin{cases} \{\widehat{v}_x\} & \text{if } x \in \mathit{dom}(\rho) \text{ and } \beta_{\mathsf{CFA}}(\rho(x)) = (\widehat{\rho}_x, \widehat{v}_x) \\ \emptyset & \text{otherwise} \end{cases} \end{aligned}$$

R_{CFA} is *generated by* β_{CFA}:

$$v\ R_{\mathsf{CFA}}\ (\widehat{\rho}, \widehat{v}) \quad \underline{\text{iff}} \quad \beta_{\mathsf{CFA}}(v) \sqsubseteq_{\mathsf{CFA}} (\widehat{\rho}, \widehat{v})$$

A Modest Generalisation

Semantics:

$$p \vdash v_1 \leadsto v_2$$

where $v_1 \in V_1, v_2 \in V_2$

Program analysis:

$$p \vdash l_1 \rhd l_2$$

where $l_1 \in L_1, l_2 \in L_2$

$$\begin{array}{cccccc} p & \vdash & v_1 & \leadsto & v_2 \\ & & \vdots & & \vdots \\ & & R_1 & \Rightarrow & R_2 \\ & & \vdots & & \vdots \\ p & \vdash & l_1 & \rhd & l_2 \end{array}$$

logical relation:

$$(p \vdash \cdot \leadsto \cdot)\ (R_1 \twoheadrightarrow R_2)\ (p \vdash \cdot \rhd \cdot)$$

Higher-Order Formulation

Assume that

- R_1 is an admissible correctness relation for V_1 and L_1 that is *generated by* the representation function $\beta_1 : V_1 \to L_1$
- R_2 is an admissible correctness relation for V_2 and L_2 that is *generated by* the representation function $\beta_2 : V_2 \to L_2$

Then the relation $R_1 \twoheadrightarrow R_2$ is an admissible correctness relation for $V_1 \to V_2$ and $L_1 \to L_2$
that is *generated by* the representation function $\beta_1 \twoheadrightarrow \beta_2$ defined by

$$(\beta_1 \twoheadrightarrow \beta_2)(\leadsto) = \lambda l_1 . \bigsqcup \{\beta_2(v_2) \mid \beta_1(v_1) \sqsubseteq l_1 \ \wedge \ v_1 \leadsto v_2\}$$

Example:

Semantics:

$\texttt{plus} \vdash (z_1, z_2) \leadsto z_1 + z_2$

where $z_1, z_2 \in \mathbf{Z}$

Program analysis:

$\texttt{plus} \vdash ZZ \rhd \{z_1 + z_2 \mid (z_1, z_2) \in ZZ\}$

where $ZZ \subseteq \mathbf{Z} \times \mathbf{Z}$

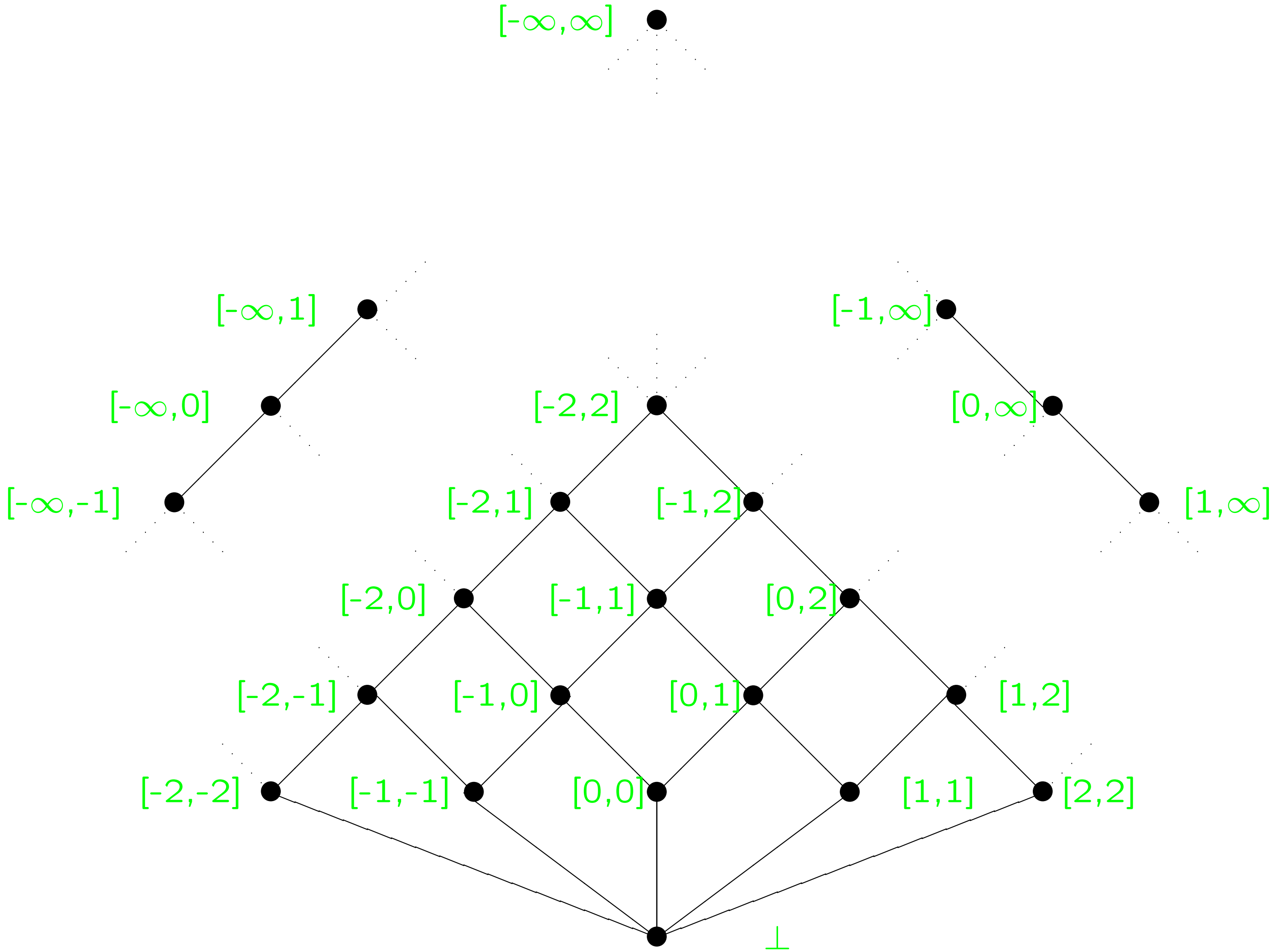
	Correctness relations	Representation functions
result	R_{Z}	$\beta_{\mathsf{Z}}(z) = \{z\}$
argument	$R_{\mathsf{Z}\times\mathsf{Z}}$	$\beta_{\mathsf{Z}\times\mathsf{Z}}(z_1, z_2) = \{(z_1, z_2)\}$
$\texttt{plus}$	$(\texttt{plus} \vdash \cdot \leadsto \cdot)\ (R_{\mathsf{Z}\times\mathsf{Z}} \twoheadrightarrow R_{\mathsf{Z}})\ (\texttt{plus} \vdash \cdot \rhd \cdot)$	$(\beta_{\mathsf{Z}\times\mathsf{Z}} \twoheadrightarrow \beta_{\mathsf{Z}})(\texttt{plus} \vdash \cdot \leadsto \cdot) \sqsubseteq (\texttt{plus} \vdash \cdot \rhd \cdot)$

Approximation of Fixed Points

- Fixed points
- Widening
- Narrowing

Example: lattice of intervals for *Array Bound Analysis*

The complete lattice **Interval** = (**Interval**, $\sqsubseteq$)



Fixed points

Let $f : L \to L$ be a *monotone function* on a complete lattice $L = (L, \sqsubseteq, \sqcup, \sqcap, \bot, \top)$.

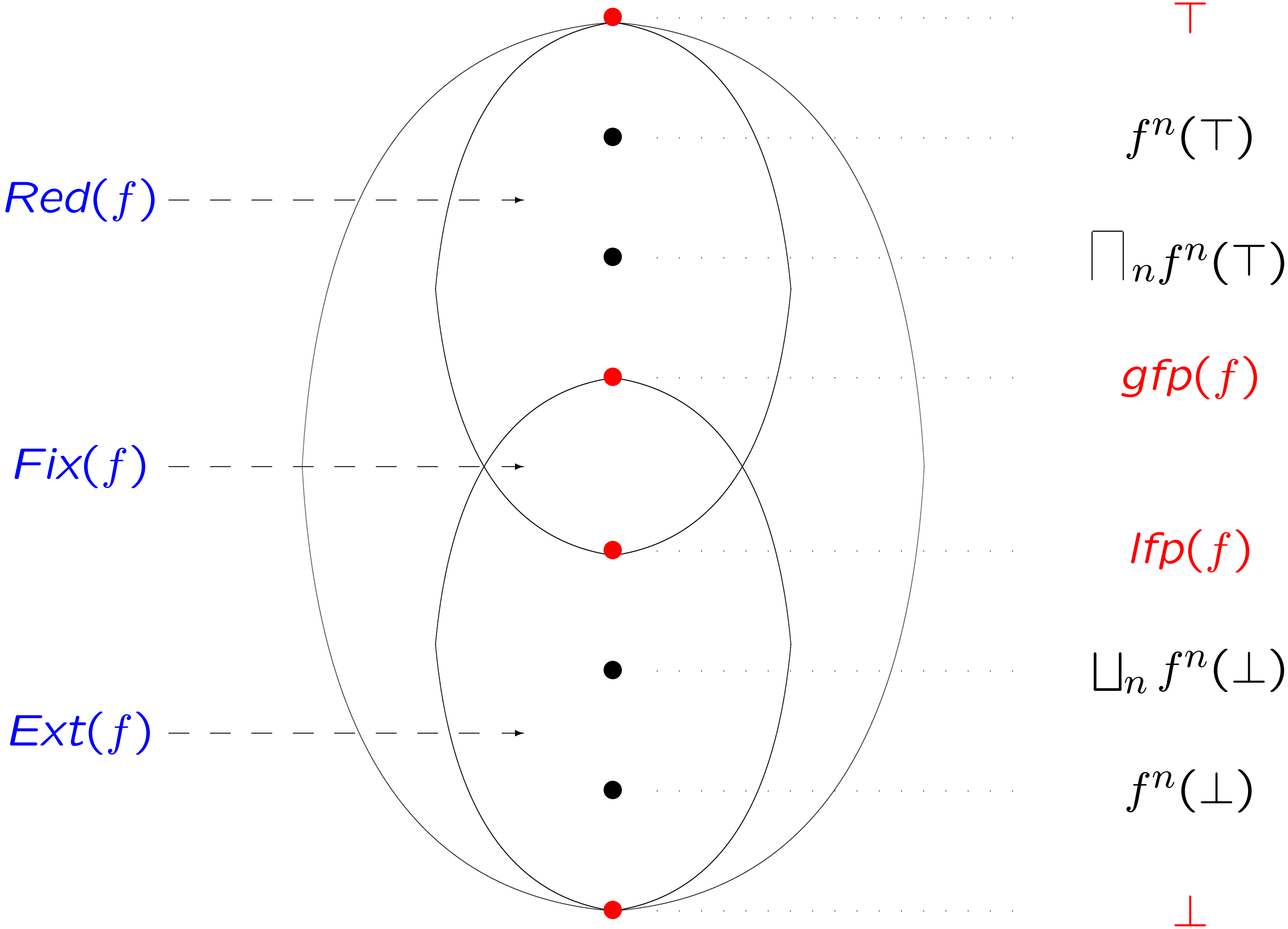
l is a *fixed point*	iff	$f(l) = l$	$\mathit{Fix}(f) = \{l \mid f(l) = l\}$
f is *reductive* at l	iff	$f(l) \sqsubseteq l$	$\mathit{Red}(f) = \{l \mid f(l) \sqsubseteq l\}$
f is *extensive* at l	iff	$f(l) \sqsupseteq l$	$\mathit{Ext}(f) = \{l \mid f(l) \sqsupseteq l\}$

Tarski's Theorem ensures that

$$\mathit{lfp}(f) = \bigsqcap \mathit{Fix}(f) = \bigsqcap \mathit{Red}(f) \in \mathit{Fix}(f) \subseteq \mathit{Red}(f)$$

$$\mathit{gfp}(f) = \bigsqcup \mathit{Fix}(f) = \bigsqcup \mathit{Ext}(f) \in \mathit{Fix}(f) \subseteq \mathit{Ext}(f)$$

Fixed points of f



Widening Operators

Problem: We cannot guarantee that $(f^n(\bot))_n$ eventually stabilises nor that its least upper bound necessarily equals *lfp*(f).

Idea: We replace $(f^n(\bot))_n$ by a new sequence $(f_\nabla^n)_n$ that is known to eventually stabilise and to do so with a value that is a safe (upper) approximation of the least fixed point.

The new sequence is parameterised on the widening operator ∇: an upper bound operator satisfying a finiteness condition.

Upper bound operators

$\check{\sqcup} : L \times L \to L$ is an *upper bound operator* iff

$$l_1 \;\sqsubseteq\; l_1 \mathbin{\check{\sqcup}} l_2 \;\sqsupseteq\; l_2$$

for all $l_1, l_2 \in L$.

Let $(l_n)_n$ be a sequence of elements of L. Define the sequence $(l_n^{\check{\sqcup}})_n$ by:

$$l_n^{\check{\sqcup}} = \begin{cases} l_n & \text{if } n = 0 \\ l_{n-1}^{\check{\sqcup}} \mathbin{\check{\sqcup}} l_n & \text{if } n > 0 \end{cases}$$

Fact: If $(l_n)_n$ is a sequence and $\check{\sqcup}$ is an upper bound operator then $(l_n^{\check{\sqcup}})_n$ is an ascending chain; furthermore $l_n^{\check{\sqcup}} \sqsupseteq \bigsqcup\{l_0, l_1, \cdots, l_n\}$ for all n.

Example:

Let *int* be an arbitrary but fixed element of **Interval**.

An upper bound operator:

$$int_1 \breve{\sqcup}^{int}\ int_2 = \begin{cases} int_1 \sqcup int_2 & \text{if } int_1 \sqsubseteq int \ \vee\ int_2 \sqsubseteq int_1 \\ [-\infty, \infty] & \text{otherwise} \end{cases}$$

Example: $[1,2]\breve{\sqcup}^{[0,2]}[2,3] = [1,3]$ and $[2,3]\breve{\sqcup}^{[0,2]}[1,2] = [-\infty,\infty]$.

Transformation of:	$[0,0], [1,1], [2,2], [3,3], \boxed{[4,4]}, [5,5], \cdots$
If $int = [0,\infty]$:	$[0,0], [0,1], [0,2], [0,3], \boxed{[0,4]}, [0,5], \cdots$
If $int = [0,2]$:	$[0,0], [0,1], [0,2], [0,3], \boxed{[-\infty,\infty]}, [-\infty,\infty], \cdots$

Widening operators

An operator $\nabla : L \times L \to L$ is a *widening operator* iff

- it is an upper bound operator, and
- for all ascending chains $(l_n)_n$ the ascending chain $(l_n^{\nabla})_n$ eventually stabilises.

Widening operators

Given a monotone function $f : L \to L$ and a widening operator ∇ define the sequence $(f_\nabla^n)_n$ by

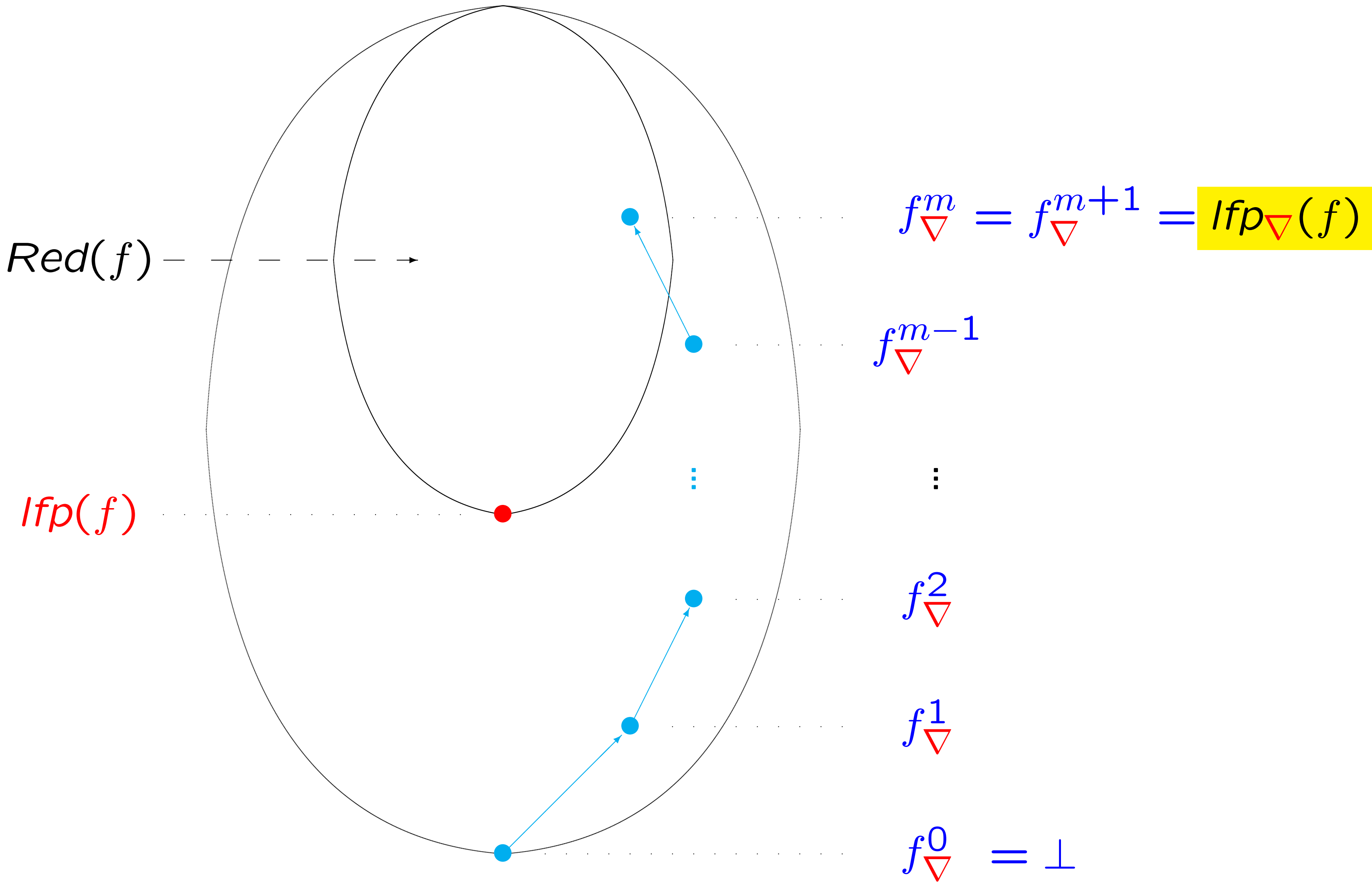
$$f_\nabla^n = \begin{cases} \bot & \text{if } n = 0 \\ f_\nabla^{n-1} & \text{if } n > 0 \ \wedge \ f(f_\nabla^{n-1}) \sqsubseteq f_\nabla^{n-1} \\ f_\nabla^{n-1} \ \nabla \ f(f_\nabla^{n-1}) & \text{otherwise} \end{cases}$$

One can show that:

- $(f_\nabla^n)_n$ is an ascending chain that eventually stabilises
- it happens when $f(f_\nabla^m) \sqsubseteq f_\nabla^m$ for some value of m
- Tarski's Theorem then gives $f_\nabla^m \sqsupseteq \mathit{lfp}(f)$

$$\mathit{lfp}_\nabla(f) = f_\nabla^m$$

The widening operator ∇ applied to f



Example:

Let K be a *finite* set of integers, e.g. the set of integers explicitly mentioned in a given program.

We shall define a widening operator ∇ based on K.

Idea: $[z_1, z_2]\ \nabla\ [z_3, z_4]$ is

$$[\ \mathsf{LB}(z_1, z_3)\ ,\ \mathsf{UB}(z_2, z_4)\ ]$$

where

- $\mathsf{LB}(z_1, z_3) \in \{z_1\} \cup K \cup \{-\infty\}$ is the best possible lower bound, and
- $\mathsf{UB}(z_2, z_4) \in \{z_2\} \cup K \cup \{\infty\}$ is the best possible upper bound.

The effect: a change in any of the bounds of the interval $[z_1, z_2]$ can only take place finitely many times – corresponding to the cardinality of K.

Example (cont.) — formalisation:

Let $z_i \in \mathbf{Z}' = \mathbf{Z} \cup \{-\infty, \infty\}$ and write:

$$\mathrm{LB}_K(z_1, z_3) = \begin{cases} z_1 & \text{if } z_1 \le z_3 \\ k & \text{if } z_3 < z_1 \ \wedge \ k = \max\{k \in K \mid k \le z_3\} \\ -\infty & \text{if } z_3 < z_1 \ \wedge \ \forall k \in K : z_3 < k \end{cases}$$

$$\mathrm{UB}_K(z_2, z_4) = \begin{cases} z_2 & \text{if } z_4 \le z_2 \\ k & \text{if } z_2 < z_4 \ \wedge \ k = \min\{k \in K \mid z_4 \le k\} \\ \infty & \text{if } z_2 < z_4 \ \wedge \ \forall k \in K : k < z_4 \end{cases}$$

$$int_1 \ \nabla \ int_2 = \begin{cases} \bot & \text{if } int_1 = int_2 = \bot \\ [\ \mathrm{LB}_K(\inf(int_1), \inf(int_2))\ ,\ \mathrm{UB}_K(\sup(int_1), \sup(int_2))\ ] & \text{otherwise} \end{cases}$$

Example (cont.):

Consider the ascending chain $(int_n)_n$

$$[0,1],[0,2],[0,3],[0,4],[0,5],[0,6],[0,7],\cdots$$

and assume that $K = \{3,5\}$.

Then $(int_n^{\nabla})_n$ is the chain

$$[0,1],[0,3],[0,3],[0,5],[0,5],[0,\infty],[0,\infty],\cdots$$

which eventually stabilises.

Narrowing Operators

Status: Widening gives us an upper approximation $\mathit{lfp}_\nabla(f)$ of the least fixed point of f.

Observation: $f(\mathit{lfp}_\nabla(f)) \sqsubseteq \mathit{lfp}_\nabla(f)$ so the approximation can be improved by considering the iterative sequence $(f^n(\mathit{lfp}_\nabla(f)))_n$.

It will satisfy $f^n(\mathit{lfp}_\nabla(f)) \sqsupseteq \mathit{lfp}(f)$ for all n so we can stop at an arbitrary point.

The notion of narrowing is *one way* of encapsulating a termination criterion for the sequence.

Narrowing

An operator $\triangle : L \times L \to L$ is a *narrowing operator* iff

- $l_2 \sqsubseteq l_1 \;\Rightarrow\; l_2 \sqsubseteq (l_1 \triangle l_2) \sqsubseteq l_1$ for all $l_1, l_2 \in L$, and
- for all descending chains $(l_n)_n$ the sequence $(l_n^\triangle)_n$ eventually stabilises.

Recall: The sequence $(l_n^\triangle)_n$ is defined by:

$$l_n^\triangle = \begin{cases} l_n & \text{if } n = 0 \\ l_{n-1}^\triangle \triangle l_n & \text{if } n > 0 \end{cases}$$

Narrowing

We construct the sequence $([f]^n_\triangle)_n$

$$[f]^n_\triangle = \begin{cases} \mathit{lfp}_\nabla(f) & \text{if } n = 0 \\ [f]^{n-1}_\triangle \;\triangle\; f([f]^{n-1}_\triangle) & \text{if } n > 0 \end{cases}$$

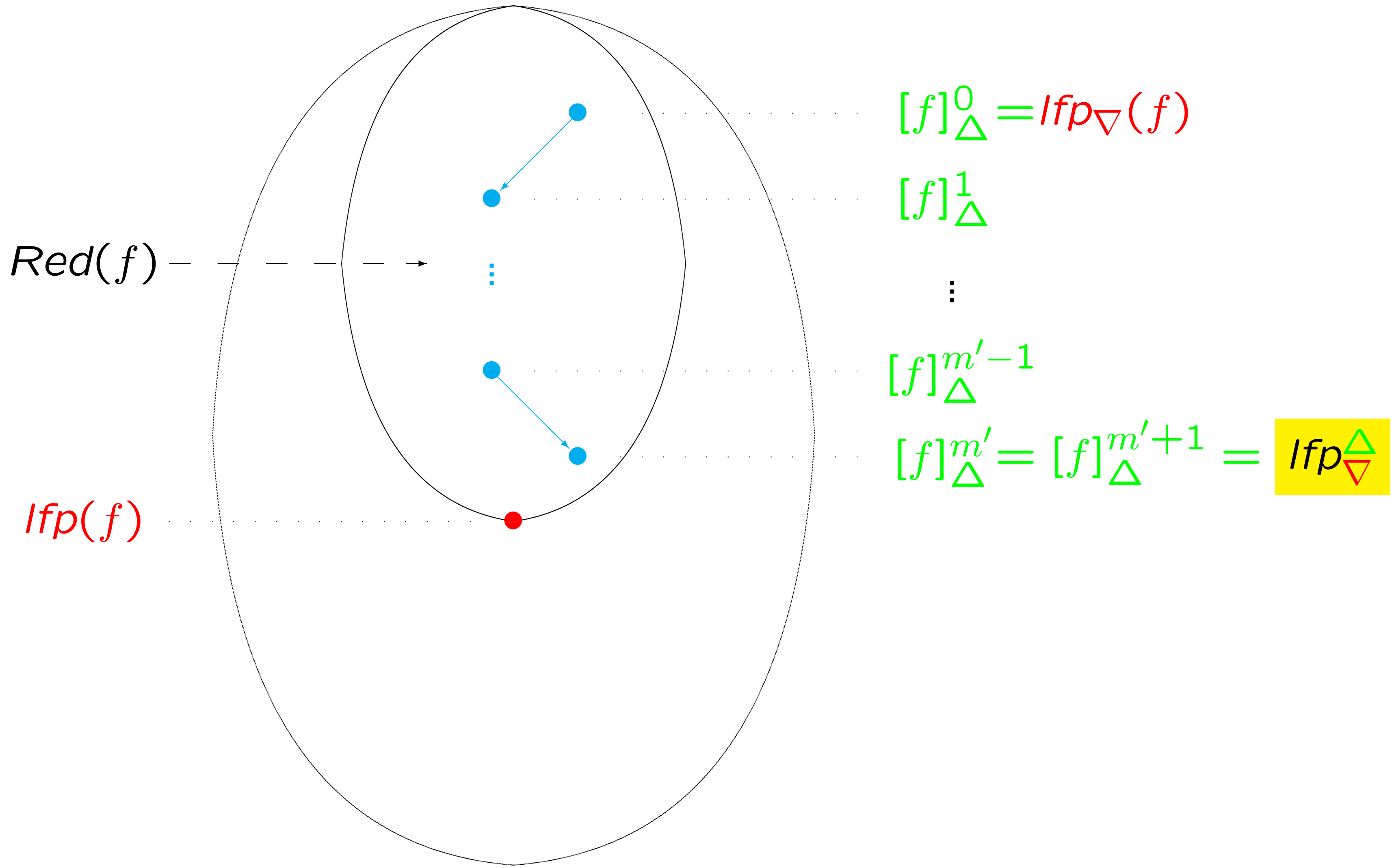
One can show that:

- $([f]^n_\triangle)_n$ is a descending chain where all elements satisfy $\mathit{lfp}(f) \sqsubseteq [f]^n_\triangle$

- the chain eventually stabilises so $[f]^{m'}_\triangle = [f]^{m'+1}_\triangle$ for some value m'

$$\boxed{\mathit{lfp}^\triangle_\nabla(f) = [f]^{m'}_\triangle}$$

The narrowing operator Δ applied to f



Example:

The complete lattice $(\mathbf{Interval}, \sqsubseteq)$ has two kinds of infinite descending chains:

- those with elements of the form $[-\infty, z]$, $z \in \mathbf{Z}$
- those with elements of the form $[z, \infty]$, $z \in \mathbf{Z}$

Idea: Given some fixed non-negative number N the narrowing operator $\triangle_N$ will force an infinite descending chain

$$[z_1, \infty], [z_2, \infty], [z_3, \infty], \cdots$$

(where $z_1 < z_2 < z_3 < \cdots$) to stabilise when $z_i > N$

Similarly, for a descending chain with elements of the form $[-\infty, z_i]$ the narrowing operator will force it to stabilise when $z_i < -N$

Example (cont.) — formalisation:

Define $\triangle = \triangle_N$ by

$$int_1 \triangle int_2 = \begin{cases} \bot & \text{if } int_1 = \bot \lor int_2 = \bot \\ [z_1, z_2] & \text{otherwise} \end{cases}$$

where

$$z_1 = \begin{cases} \inf(int_1) & \text{if } N < \inf(int_2) \land \sup(int_2) = \infty \\ \inf(int_2) & \text{otherwise} \end{cases}$$

$$z_2 = \begin{cases} \sup(int_1) & \text{if } \inf(int_2) = -\infty \land \sup(int_2) < -N \\ \sup(int_2) & \text{otherwise} \end{cases}$$

Example (cont.):

Consider the infinite descending chain $([n, \infty])_n$

$$[0, \infty], [1, \infty], [2, \infty], [3, \infty], [4, \infty], [5, \infty], \cdots$$

and assume that $N = 3$.

Then the narrowing operator $\triangle_N$ will give the sequence $([n, \infty]^{\triangle})_n$

$$[0, \infty], [1, \infty], [2, \infty], [3, \infty], [3, \infty], [3, \infty], \cdots$$

Galois Connections

- Galois connections and adjunctions
- Extraction functions
- Galois insertions
- Reduction operators

Galois connections

$$L \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} M$$

α: *abstraction function*

γ: *concretisation function*

is a Galois connection if and only if

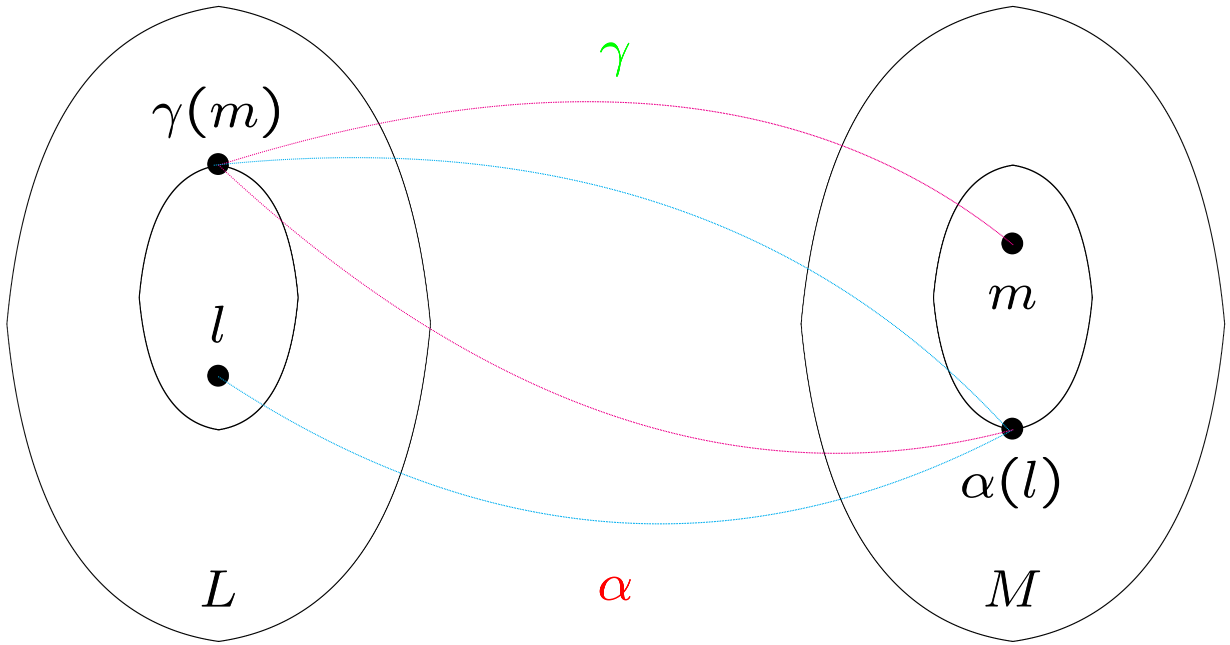
α and γ are monotone functions

that satisfy

$$\gamma \circ \alpha \sqsupseteq \lambda l.l$$

$$\alpha \circ \gamma \sqsubseteq \lambda m.m$$

Galois connections



$\gamma \circ \alpha \sqsupseteq \lambda l.l$ $\qquad$ $\alpha \circ \gamma \sqsubseteq \lambda m.m$

Example:

Galois connection

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\text{ZI}}, \gamma_{\text{ZI}}, \textbf{Interval})$$

with concretisation function

$$\gamma_{\text{ZI}}(\mathit{int}) = \{z \in \mathbf{Z} \mid \text{inf}(\mathit{int}) \leq z \leq \text{sup}(\mathit{int})\}$$

and abstraction function

$$\alpha_{\text{ZI}}(Z) = \begin{cases} \bot & \text{if } Z = \emptyset \\ [\text{inf}'(Z), \text{sup}'(Z)] & \text{otherwise} \end{cases}$$

Examples:

$$\begin{aligned} \gamma_{\text{ZI}}([0,3]) &= \{0,1,2,3\} \\ \gamma_{\text{ZI}}([0,\infty]) &= \{z \in \mathbf{Z} \mid z \geq 0\} \\ \alpha_{\text{ZI}}(\{0,1,3\}) &= [0,3] \\ \alpha_{\text{ZI}}(\{2*z \mid z > 0\}) &= [2,\infty] \end{aligned}$$

Adjunctions

$$L \quad \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \quad M$$

is an *adjunction* if and only if

$$\alpha : L \rightarrow M \text{ and } \gamma : M \rightarrow L \text{ are total functions}$$

that satisfy

$$\alpha(l) \sqsubseteq m \qquad \underline{\text{iff}} \qquad l \sqsubseteq \gamma(m)$$

for all $l \in L$ and $m \in M$.

Proposition: (α, γ) is an adjunction iff it is a Galois connection.

Galois connections from representation functions

A representation function $\beta : V \to L$ gives rise to a Galois connection

$$(\mathcal{P}(V), \alpha, \gamma, L)$$

where

$$\begin{array}{rcl}\alpha(V') & = & \bigsqcup\{\beta(v) \mid v \in V'\} \\ \gamma(l) & = & \{v \in V \mid \beta(v) \sqsubseteq l\}\end{array}$$

for $V' \subseteq V$ and $l \in L$.

This indeed defines an adjunction:

$$\begin{array}{rcl}\alpha(V') \sqsubseteq l & \Leftrightarrow & \bigsqcup\{\beta(v) \mid v \in V'\} \sqsubseteq l \\ & \Leftrightarrow & \forall v \in V' : \beta(v) \sqsubseteq l \\ & \Leftrightarrow & V' \subseteq \gamma(l)\end{array}$$

Galois connections from extraction functions

An *extraction function*

$$\eta : V \to D$$

maps the values of V to their best descriptions in D.

It gives rise to a representation function $\beta_\eta : V \to \mathcal{P}(D)$ (corresponding to $L = (\mathcal{P}(D), \subseteq)$) defined by

$$\beta_\eta(v) = \{\eta(v)\}$$

The associated Galois connection is

$$(\mathcal{P}(V), \alpha_\eta, \gamma_\eta, \mathcal{P}(D))$$

where

$$\begin{array}{rcl} \alpha_\eta(V') & = & \bigcup\{\beta_\eta(v) \mid v \in V'\} \quad = \quad \{\eta(v) \mid v \in V'\} \\ \gamma_\eta(D') & = & \{v \in V \mid \beta_\eta(v) \subseteq D'\} \quad = \quad \{v \mid \eta(v) \in D'\} \end{array}$$

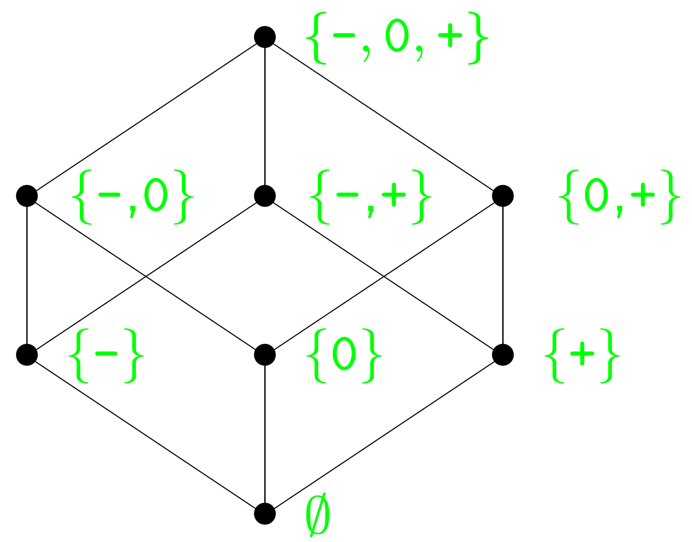
Example:

Extraction function

$$\text{sign} : \mathbf{Z} \rightarrow \mathbf{Sign}$$

specified by

$$\text{sign}(z) = \begin{cases} - & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ + & \text{if } z > 0 \end{cases}$$



Galois connection

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\text{sign}}, \gamma_{\text{sign}}, \mathcal{P}(\mathbf{Sign}))$$

with

$$\begin{aligned} \alpha_{\text{sign}}(Z) &= \{\text{sign}(z) \mid z \in Z\} \\ \gamma_{\text{sign}}(S) &= \{z \in \mathbf{Z} \mid \text{sign}(z) \in S\} \end{aligned}$$

Properties of Galois Connections

Lemma: If (L, α, γ, M) is a Galois connection then:

- α uniquely determines γ by $\gamma(m) = \bigsqcup\{l \mid \alpha(l) \sqsubseteq m\}$
- γ uniquely determines α by $\alpha(l) = \bigsqcap\{m \mid l \sqsubseteq \gamma(m)\}$
- α is completely additive and γ is completely multiplicative

In particular $\alpha(\bot) = \bot$ and $\gamma(\top) = \top$.

Lemma:

- If $\alpha : L \to M$ is completely additive then there exists (an upper adjoint) $\gamma : M \to L$ such that (L, α, γ, M) is a Galois connection.
- If $\gamma : M \to L$ is completely multiplicative then there exists (a lower adjoint) $\alpha : L \to M$ such that (L, α, γ, M) is a Galois connection.

Fact: If (L, α, γ, M) is a Galois connection then

- $\alpha \circ \gamma \circ \alpha = \alpha$ and $\gamma \circ \alpha \circ \gamma = \gamma$

Example:

Define $\gamma_{\mathrm{IS}} : \mathcal{P}(\mathbf{Sign}) \to \mathbf{Interval}$ by:

$$\begin{array}{rcl}
\gamma_{\mathrm{IS}}(\{\text{-},0,\text{+}\}) & = & [-\infty,\infty] \\
\gamma_{\mathrm{IS}}(\{\text{-},\text{+}\}) & = & [-\infty,\infty] \\
\gamma_{\mathrm{IS}}(\{\text{-}\}) & = & [-\infty,-1] \\
\gamma_{\mathrm{IS}}(\{\text{+}\}) & = & [1,\infty]
\end{array}
\qquad
\begin{array}{rcl}
\gamma_{\mathrm{IS}}(\{\text{-},0\}) & = & [-\infty,0] \\
\gamma_{\mathrm{IS}}(\{0,\text{+}\}) & = & [0,\infty] \\
\gamma_{\mathrm{IS}}(\{0\}) & = & [0,0] \\
\gamma_{\mathrm{IS}}(\emptyset) & = & \bot
\end{array}$$

Does there exist an abstraction function

$$\alpha_{\mathrm{IS}} : \mathbf{Interval} \to \mathcal{P}(\mathbf{Sign})$$

such that $(\mathbf{Interval}, \alpha_{\mathrm{IS}}, \gamma_{\mathrm{IS}}, \mathcal{P}(\mathbf{Sign}))$ is a Galois connection?

Example (cont.):

Is γ_{IS} completely multiplicative?

- if yes: then there exists a Galois connection
- if no: then there cannot exist a Galois connection

Lemma: If L and M are complete lattices and M is finite then $\gamma : M \to L$ is completely multiplicative if and only if the following hold:

- $\gamma : M \to L$ is monotone,
- $\gamma(\top) = \top$, and
- $\gamma(m_1 \sqcap m_2) = \gamma(m_1) \sqcap \gamma(m_2)$ whenever $m_1 \not\sqsubseteq m_2 \wedge m_2 \not\sqsubseteq m_1$

We calculate

$$\begin{array}{rcccl} \gamma_{\text{IS}}(\{\text{-},0\} \cap \{\text{-},\text{+}\}) & = & \gamma_{\text{IS}}(\{\text{-}\}) & = & [-\infty, -1] \\ \gamma_{\text{IS}}(\{\text{-},0\}) \sqcap \gamma_{\text{IS}}(\{\text{-},\text{+}\}) & = & [-\infty, 0] \sqcap [-\infty, \infty] & = & [-\infty, 0] \end{array}$$

showing that there is no Galois connection involving γ_{IS}.

Galois Connections are the Right Concept

We use the mundane approach to correctness to demonstrate this for:

- Admissible correctness relations

- Representation functions

The mundane approach: correctness relations

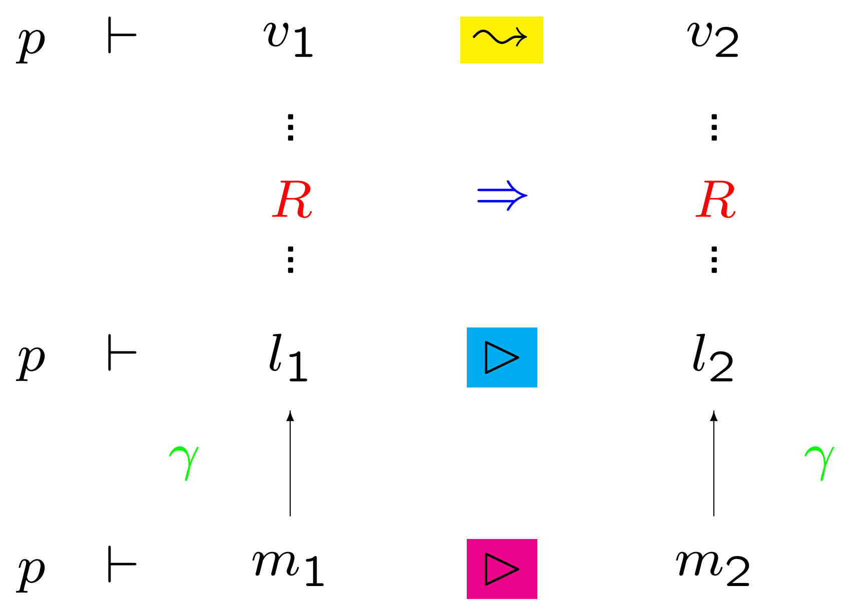
Assume

- $R : V \times L \to \{\textit{true}, \textit{false}\}$ is an admissible correctness relation
- (L, α, γ, M) is a Galois connection

Then $S : V \times M \to \{\textit{true}, \textit{false}\}$ defined by

$$v \; S \; m \qquad \underline{\text{iff}} \qquad v \; R \; (\gamma(m))$$

is an admissible correctness relation between V and M



The mundane approach: representation functions

Assume

- $R : V \times L \to \{true, false\}$ is *generated by* $\beta : V \to L$
- (L, α, γ, M) is a Galois connection

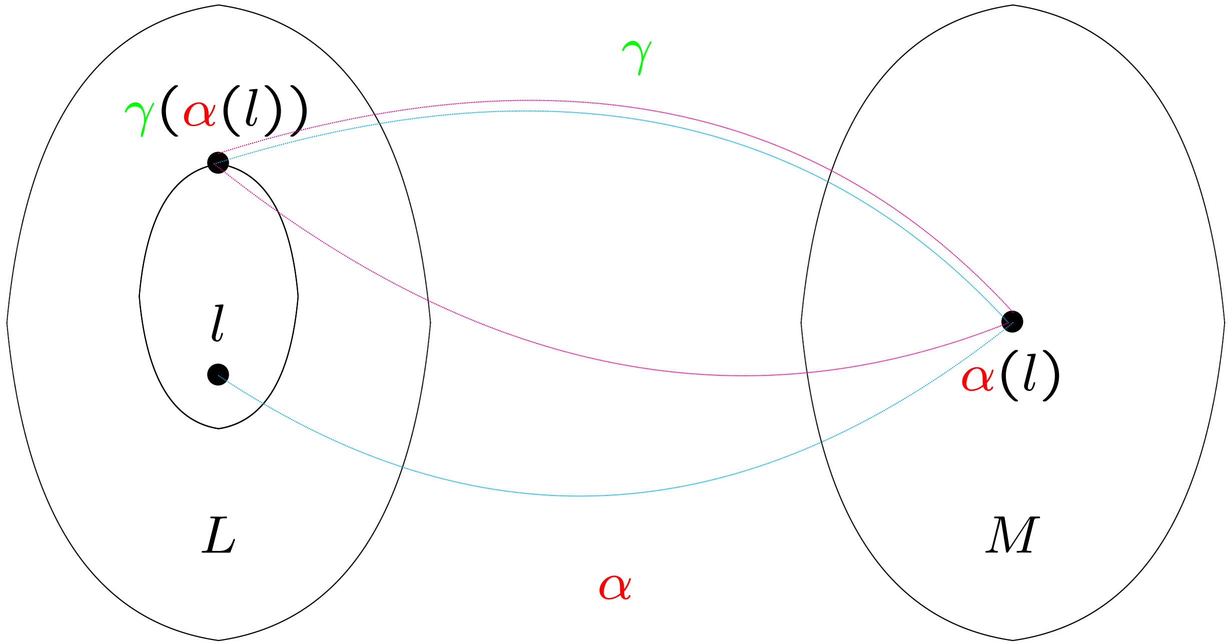
Then $S : V \times M \to \{true, false\}$ defined by

$$v \; S \; m \qquad \underline{\text{iff}} \qquad v \; R \; (\gamma(m))$$

is *generated by* $\alpha \circ \beta : V \to M$

$$\begin{array}{ccccccc} p & \vdash & v_1 & & \leadsto & & v_2 \\ & & & & & & \\ & \beta & \downarrow & & \Rightarrow & & \downarrow & \beta \\ & & \sqsubseteq & & & & \sqsubseteq \\ p & \vdash & l_1 & & \rhd & & l_2 \\ & \alpha & \downarrow & & & & \downarrow & \alpha \\ p & \vdash & m_1 & & \rhd & & m_2 \end{array}$$

Galois Insertions



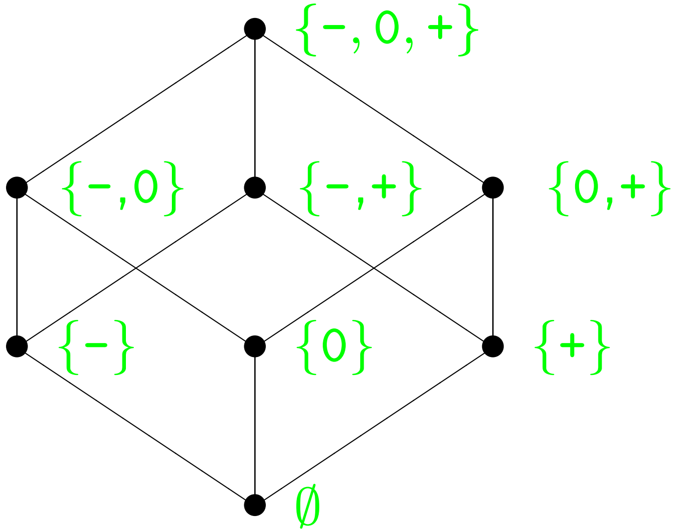
Monotone functions satisfying: $\gamma \circ \alpha \sqsupseteq \lambda l.l \qquad \alpha \circ \gamma = \lambda m.m$

Example (1):

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\mathsf{sign}}, \gamma_{\mathsf{sign}}, \mathcal{P}(\mathbf{Sign}))$$

where sign : $\mathbf{Z} \rightarrow \mathbf{Sign}$ is specified by:

$$\mathsf{sign}(z) = \begin{cases} - & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ + & \text{if } z > 0 \end{cases}$$



Is it a Galois insertion?

Example (2):

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\mathsf{signparity}}, \gamma_{\mathsf{signparity}}, \mathcal{P}(\mathbf{Sign} \times \textsf{\textbf{Parity}}))$$

where $\mathbf{Sign} = \{\textsf{-}, \textsf{0}, \textsf{+}\}$ and $\textsf{\textbf{Parity}} = \{\textsf{odd}, \textsf{even}\}$

and signparity : $\mathbf{Z} \to \mathbf{Sign} \times \textsf{\textbf{Parity}}$:

$$\mathsf{signparity}(z) = \begin{cases} (\mathsf{sign}(z), \mathsf{odd}) & \text{if } z \text{ is odd} \\ (\mathsf{sign}(z), \mathsf{even}) & \text{if } z \text{ is even} \end{cases}$$

Is it a Galois insertion?

Properties of Galois Insertions

Lemma: For a Galois connection (L, α, γ, M) the following claims are equivalent:

(i) (L, α, γ, M) is a Galois insertion;

(ii) α is surjective: $\forall m \in M : \exists l \in L : \alpha(l) = m$;

(iii) γ is injective: $\forall m_1, m_2 \in M : \gamma(m_1) = \gamma(m_2) \Rightarrow m_1 = m_2$; and

(iv) γ is an order-similarity: $\forall m_1, m_2 \in M : \gamma(m_1) \sqsubseteq \gamma(m_2) \Leftrightarrow m_1 \sqsubseteq m_2$.

Corollary: A Galois connection specified by an *extraction* function $\eta : V \to D$ is a Galois insertion if and only if η is surjective.

Example (1) reconsidered:

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\mathsf{sign}}, \gamma_{\mathsf{sign}}, \mathcal{P}(\mathbf{Sign}))$$

$$\mathsf{sign}(z) = \begin{cases} \mathsf{-} & \text{if } z < 0 \\ \mathsf{0} & \text{if } z = 0 \\ \mathsf{+} & \text{if } z > 0 \end{cases}$$

is a Galois insertion because sign is surjective.

Example (2) reconsidered:

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\mathsf{signparity}}, \gamma_{\mathsf{signparity}}, \mathcal{P}(\mathbf{Sign} \times \mathbf{Parity}))$$

$$\mathsf{signparity}(z) = \begin{cases} (\mathsf{sign}(z), \mathsf{odd}) & \text{if } z \text{ is odd} \\ (\mathsf{sign}(z), \mathsf{even}) & \text{if } z \text{ is even} \end{cases}$$

is not a Galois insertion because signparity is not surjective.

Reduction Operators

Given a Galois connection (L, α, γ, M) it is always possible to obtain a Galois insertion by enforcing that the concretisation function γ is injective.

Idea: remove the superfluous elements from M using a *reduction operator*

$$\varsigma : M \to M$$

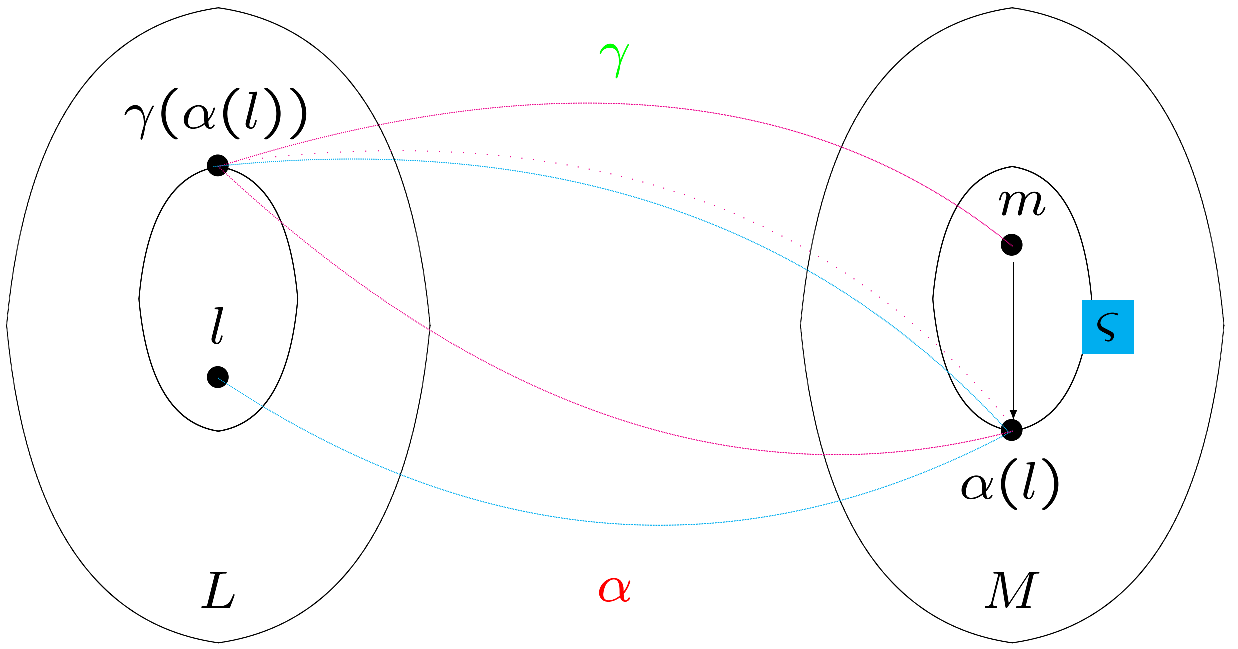
defined from the Galois connection.

Proposition: Let (L, α, γ, M) be a Galois connection and define the reduction operator $\varsigma : M \to M$ by

$$\varsigma(m) = \bigsqcap \{m' \mid \gamma(m) = \gamma(m')\}$$

Then $\varsigma[M] = (\{\varsigma(m) \mid m \in M\}, \sqsubseteq_M)$ is a complete lattice and $(L, \alpha, \gamma, \varsigma[M])$ is a Galois insertion.

The reduction operator $\varsigma : M \to M$



Reduction operators from extraction functions

Assume that the Galois connection $(\mathcal{P}(V), \alpha_\eta, \gamma_\eta, \mathcal{P}(D))$ is given by an extraction function $\eta : V \to D$.

Then the reduction operator ς_η is given by

$$\varsigma_\eta(D') = D' \cap \eta[V]$$

where $\eta[V] = \{d \in D \mid \exists v \in V : \eta(v) = d\}$.

Since $\varsigma_\eta[\mathcal{P}(D)]$ is isomorphic to $\mathcal{P}(\eta[V])$ the resulting Galois insertion is isomorphic to

$$(\mathcal{P}(V), \alpha_\eta, \gamma_\eta, \mathcal{P}(\eta[V]))$$

Systematic Design of Galois Connections

The "functional composition" (or "sequential composition") of two Galois connections is also a Galois connection:

$$L_0 \underset{\alpha_1}{\overset{\gamma_1}{\rightleftarrows}} L_1 \underset{\alpha_2}{\overset{\gamma_2}{\rightleftarrows}} L_2 \underset{\alpha_3}{\overset{\gamma_3}{\rightleftarrows}} \cdots \underset{\alpha_k}{\overset{\gamma_k}{\rightleftarrows}} L_k$$

A catalogue of techniques for combining Galois connections:

- independent attribute method
- direct product
- reduced product
- total function space
- relational method
- direct tensor product
- reduced tensor product
- monotone function space

Running Example: Array Bound Analysis

Approximation of the difference in magnitude between two numbers (typically the index and the bound):

- a Galois connection for approximating pairs (z_1, z_2) of integers by their difference $|z_1| - |z_2|$
- a Galois connection for approximating integers using a finite lattice $\{$<-1, -1, 0, +1, >+1$\}$
- a Galois connection for their functional composition

Example: Difference in Magnitude

$$(\mathcal{P}(\mathbf{Z} \times \mathbf{Z}), \alpha_{\mathsf{diff}}, \gamma_{\mathsf{diff}}, \mathcal{P}(\mathbf{Z}))$$

where the extraction function diff $: \mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}$ calculates the difference in magnitude:

$$\mathsf{diff}(z_1, z_2) = |z_1| - |z_2|$$

The abstraction and concretisation functions are

$$\begin{aligned} \alpha_{\mathsf{diff}}(ZZ) &= \{|z_1| - |z_2| \mid (z_1, z_2) \in ZZ\} \\ \gamma_{\mathsf{diff}}(Z) &= \{(z_1, z_2) \mid |z_1| - |z_2| \in Z\} \end{aligned}$$

for $ZZ \subseteq \mathbf{Z} \times \mathbf{Z}$ and $Z \subseteq \mathbf{Z}$.

Example: Finite Approximation

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\textsf{range}}, \gamma_{\textsf{range}}, \mathcal{P}(\textbf{Range}))$$

where **Range** $= \{$<-1, -1, 0, +1, >+1$\}$
and the extraction function range : $\mathbf{Z} \rightarrow$ **Range** is

$$\textsf{range}(z) = \begin{cases} \texttt{<-1} & \text{if } z < -1 \\ \texttt{-1} & \text{if } z = -1 \\ \texttt{0} & \text{if } z = 0 \\ \texttt{+1} & \text{if } z = 1 \\ \texttt{>+1} & \text{if } z > 1 \end{cases}$$

The abstraction and concretisation functions are

$$\begin{aligned} \alpha_{\textsf{range}}(Z) &= \{\textsf{range}(z) \mid z \in Z\} \\ \gamma_{\textsf{range}}(R) &= \{z \mid \textsf{range}(z) \in R\} \end{aligned}$$

for $Z \subseteq \mathbf{Z}$ and $R \subseteq$ **Range**.

Example: Functional Composition

$$(\mathcal{P}(\mathbf{Z} \times \mathbf{Z}), \alpha_{\mathsf{R}}, \gamma_{\mathsf{R}}, \mathcal{P}(\textbf{Range}))$$

where

$$\begin{aligned} \alpha_{\mathsf{R}} &= \alpha_{\mathsf{range}} \circ \alpha_{\mathsf{diff}} \\ \gamma_{\mathsf{R}} &= \gamma_{\mathsf{diff}} \circ \gamma_{\mathsf{range}} \end{aligned}$$

The explicit formulae for the abstraction and concretisation functions

$$\begin{aligned} \alpha_{\mathsf{R}}(ZZ) &= \{\mathsf{range}(|z_1|-|z_2|) \mid (z_1, z_2) \in ZZ\} \\ \gamma_{\mathsf{R}}(R) &= \{(z_1, z_2) \mid \mathsf{range}(|z_1|-|z_2|) \in R\} \end{aligned}$$

correspond to the extraction function $\mathsf{range} \circ \mathsf{diff}$.

Approximation of Pairs

Independent Attribute Method

Let $(L_1, \alpha_1, \gamma_1, M_1)$ and $(L_2, \alpha_2, \gamma_2, M_2)$ be Galois connections.

The *independent attribute method* gives a Galois connection

$$(L_1 \times L_2, \alpha, \gamma, M_1 \times M_2)$$

where

$$\begin{aligned} \alpha(l_1, l_2) &= (\alpha_1(l_1), \alpha_2(l_2)) \\ \gamma(m_1, m_2) &= (\gamma_1(m_1), \gamma_2(m_2)) \end{aligned}$$

Example: Detection of Signs Analysis

Given

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\mathsf{sign}}, \gamma_{\mathsf{sign}}, \mathcal{P}(\mathbf{Sign}))$$

using the extraction function sign.

The independent attribute method gives

$$(\mathcal{P}(\mathbf{Z}) \times \mathcal{P}(\mathbf{Z}), \alpha_{\mathsf{SS}}, \gamma_{\mathsf{SS}}, \mathcal{P}(\mathbf{Sign}) \times \mathcal{P}(\mathbf{Sign}))$$

where

$$\begin{array}{rcl} \alpha_{\mathsf{SS}}(Z_1, Z_2) & = & (\{\mathsf{sign}(z) \mid z \in Z_1\}, \{\mathsf{sign}(z) \mid z \in Z_2\}) \\ \gamma_{\mathsf{SS}}(S_1, S_2) & = & (\{z \mid \mathsf{sign}(z) \in S_1\}, \{z \mid \mathsf{sign}(z) \in S_2\}) \end{array}$$

Motivating the Relational Method

The independent attribute method often leads to imprecision!

Semantics: The expression `(x,-x)` may have a value in

$$\{(z, -z) \mid z \in \mathbf{Z}\}$$

Analysis: When we use $\mathcal{P}(\mathbf{Z}) \times \mathcal{P}(\mathbf{Z})$ to represent sets of pairs of integers we cannot do better than representing $\{(z, -z) \mid z \in \mathbf{Z}\}$ by

$$(\mathbf{Z}, \mathbf{Z})$$

Hence the best property describing it will be

$$\alpha_{\mathsf{SS}}(\mathbf{Z}, \mathbf{Z}) = (\{\text{-}, 0, \text{+}\}, \{\text{-}, 0, \text{+}\})$$

Relational Method

Let $(\mathcal{P}(V_1), \alpha_1, \gamma_1, \mathcal{P}(D_1))$ and $(\mathcal{P}(V_2), \alpha_2, \gamma_2, \mathcal{P}(D_2))$ be Galois connections.

The *relational method* will give rise to the Galois connection

$$(\mathcal{P}(V_1 \times V_2), \alpha, \gamma, \mathcal{P}(D_1 \times D_2))$$

where

$$\begin{aligned} \alpha(VV) &= \bigcup\{\alpha_1(\{v_1\}) \times \alpha_2(\{v_2\}) \mid (v_1, v_2) \in VV\} \\ \gamma(DD) &= \{(v_1, v_2) \mid \alpha_1(\{v_1\}) \times \alpha_2(\{v_2\}) \subseteq DD\} \end{aligned}$$

Generalisation to arbitrary complete lattices: use *tensor products*.

Relational Method from Extraction Functions

Assume that the Galois connections $(\mathcal{P}(V_i), \alpha_i, \gamma_i, \mathcal{P}(D_i))$ are given by *extraction functions* $\eta_i : V_i \to D_i$ as in

$$\begin{aligned} \alpha_i(V_i') &= \{\eta_i(v_i) \mid v_i \in V_i'\} \\ \gamma_i(D_i') &= \{v_i \mid \eta_i(v_i) \in D_i'\} \end{aligned}$$

Then the Galois connection $(\mathcal{P}(V_1 \times V_2), \alpha, \gamma, \mathcal{P}(D_1 \times D_2))$ has

$$\begin{aligned} \alpha(VV) &= \{(\eta_1(v_1), \eta_2(v_2)) \mid (v_1, v_2) \in VV\} \\ \gamma(DD) &= \{(v_1, v_2) \mid (\eta_1(v_1), \eta_2(v_2)) \in DD\} \end{aligned}$$

which also can be obtained directly from the extraction function $\eta : V_1 \times V_2 \to D_1 \times D_2$ defined by

$$\eta(v_1, v_2) = (\eta_1(v_1), \eta_2(v_2))$$

Example: Detection of Signs Analysis

Using the relational method we get a Galois connection

$$(\mathcal{P}(\mathbf{Z} \times \mathbf{Z}), \alpha_{\mathsf{SS'}}, \gamma_{\mathsf{SS'}}, \mathcal{P}(\mathbf{Sign} \times \mathbf{Sign}))$$

where

$$\begin{array}{rcl} \alpha_{\mathsf{SS'}}(ZZ) & = & \{(\mathsf{sign}(z_1), \mathsf{sign}(z_2)) \mid (z_1, z_2) \in ZZ\} \\ \gamma_{\mathsf{SS'}}(SS) & = & \{(z_1, z_2) \mid (\mathsf{sign}(z_1), \mathsf{sign}(z_2)) \in SS\} \end{array}$$

corresponding to an extraction function $\mathsf{twosigns} : \mathbf{Z} \times \mathbf{Z} \to \mathbf{Sign} \times \mathbf{Sign}$ defined by

$$\mathsf{twosigns}(z_1, z_2) = (\mathsf{sign}(z_1), \mathsf{sign}(z_2))$$

Advantages of the Relational Method

Semantics: The expression `(x,-x)` may have a value in

$$\{(z, -z) \mid z \in \mathbf{Z}\}$$

In the present setting $\{(z, -z) \mid z \in \mathbf{Z}\}$ is an element of $\mathcal{P}(\mathbf{Z} \times \mathbf{Z})$.

Analysis: The best "relational" property describing it is

$$\alpha_{\mathsf{SS'}}(\{(z, -z) \mid z \in \mathbf{Z}\}) = \{(\mathsf{-}, \mathsf{+}), (\mathsf{0}, \mathsf{0}), (\mathsf{+}, \mathsf{-})\}$$

whereas the best "independent attribute" property was

$$\alpha_{\mathsf{SS}}(\mathbf{Z}, \mathbf{Z}) = (\{\mathsf{-}, \mathsf{0}, \mathsf{+}\}, \{\mathsf{-}, \mathsf{0}, \mathsf{+}\})$$

Function Spaces

Total Function Space

Let (L, α, γ, M) be a Galois connection and let S be a set.

The Galois connection for the *total function space*

$$(S \to L, \alpha', \gamma', S \to M)$$

is defined by

$$\alpha'(f) = \alpha \circ f \qquad\qquad \gamma'(g) = \gamma \circ g$$

Do we need to assume that S is non-empty?

Monotone Function Space

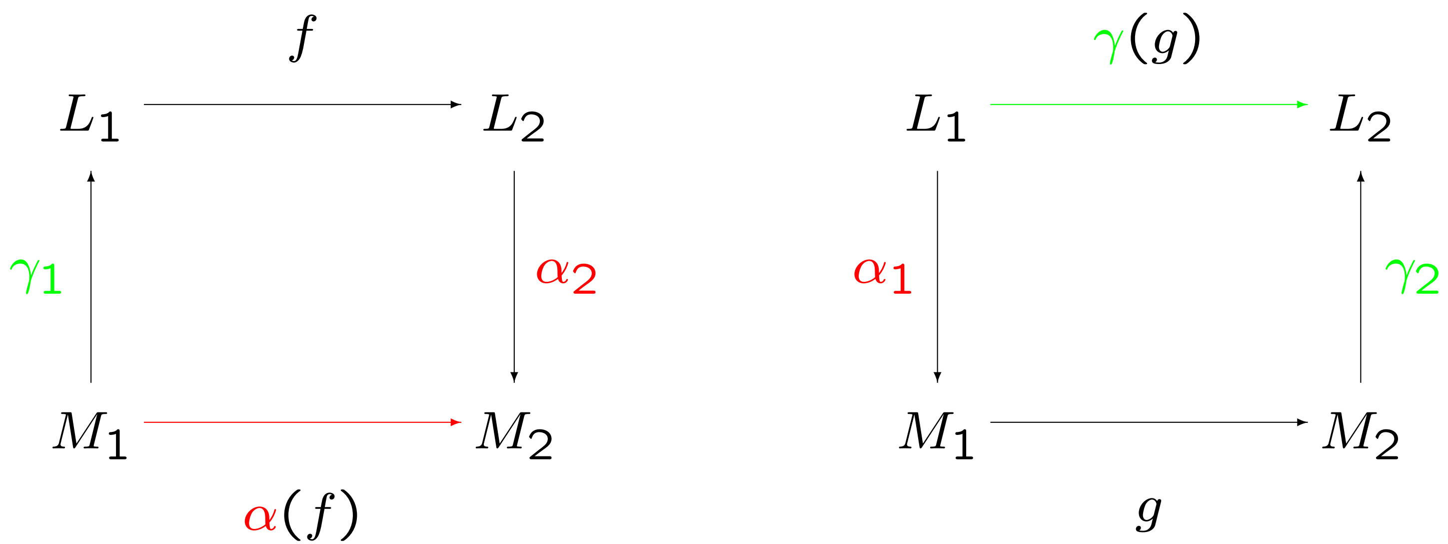
Let $(L_1, \alpha_1, \gamma_1, M_1)$ and $(L_2, \alpha_2, \gamma_2, M_2)$ be Galois connections.

The Galois connection for the *monotone function space*

$$(L_1 \rightarrow L_2, \alpha, \gamma, M_1 \rightarrow M_2)$$

is defined by

$$\alpha(f) = \alpha_2 \circ f \circ \gamma_1 \qquad\qquad \gamma(g) = \gamma_2 \circ g \circ \alpha_1$$



Performing Analyses Simultaneously

Direct Product

Let $(L, \alpha_1, \gamma_1, M_1)$ and $(L, \alpha_2, \gamma_2, M_2)$ be Galois connections.

The *direct product* is the Galois connection

$$(L, \alpha, \gamma, M_1 \times M_2)$$

defined by

$$\begin{aligned} \alpha(l) &= (\alpha_1(l), \alpha_2(l)) \\ \gamma(m_1, m_2) &= \gamma_1(m_1) \sqcap \gamma_2(m_2) \end{aligned}$$

Example:

Combining the detection of signs analysis for pairs of integers with the analysis of difference in magnitude.

We get the Galois connection

$$(\mathcal{P}(\mathbf{Z} \times \mathbf{Z}), \alpha_{\mathsf{SSR}}, \gamma_{\mathsf{SSR}}, \mathcal{P}(\mathbf{Sign} \times \mathbf{Sign}) \times \mathcal{P}(\mathbf{Range}))$$

where

$$\begin{array}{rcl} \alpha_{\mathsf{SSR}}(ZZ) & = & (\{(\mathsf{sign}(z_1), \mathsf{sign}(z_2)) \mid (z_1, z_2) \in ZZ\}, \\ & & \{\mathsf{range}(|z_1| - |z_2|) \mid (z_1, z_2) \in ZZ\}) \end{array}$$

$$\begin{array}{rcl} \gamma_{\mathsf{SSR}}(SS, R) & = & \{(z_1, z_2) \mid (\mathsf{sign}(z_1), \mathsf{sign}(z_2)) \in SS\} \\ & \cap & \{(z_1, z_2) \mid \mathsf{range}(|z_1| - |z_2|) \in R\} \end{array}$$

Motivating the Direct Tensor Product

The expression `(x, 3*x)` may have a value in

$$\{(z, 3 * z) \mid z \in \mathbf{Z}\}$$

which is described by

$$\alpha_{\mathsf{SSR}}(\{(z, 3 * z) \mid z \in \mathbf{Z}\}) = (\{(\mathsf{-}, \mathsf{-}), (\mathsf{0}, \mathsf{0}), (\mathsf{+}, \mathsf{+})\}, \{\mathsf{0}, \mathsf{<\text{-}1}\})$$

But

- any pair described by $(\mathsf{0}, \mathsf{0})$ will have a difference in magnitude described by 0
- any pair described by $(\mathsf{-}, \mathsf{-})$ or $(\mathsf{+}, \mathsf{+})$ will have a difference in magnitude described by <-1

and the analysis cannot express this.

Direct Tensor Product

Let $(\mathcal{P}(V), \alpha_1, \gamma_1, \mathcal{P}(D_1))$ and $(\mathcal{P}(V), \alpha_2, \gamma_2, \mathcal{P}(D_2))$ be Galois connections.

The *direct tensor product* is the Galois connection

$$(\mathcal{P}(V), \alpha, \gamma, \mathcal{P}(D_1 \times D_2))$$

defined by

$$\begin{array}{rcl} \alpha(V') & = & \bigcup\{\alpha_1(\{v\}) \times \alpha_2(\{v\}) \mid v \in V'\} \\ \gamma(\mathit{DD}) & = & \{v \mid \alpha_1(\{v\}) \times \alpha_2(\{v\}) \subseteq \mathit{DD}\} \end{array}$$

Direct Tensor Product from Extraction Functions

Assume that the Galois connections $(\mathcal{P}(V), \alpha_i, \gamma_i, \mathcal{P}(D_i))$ are given by *extraction functions* $\eta_i : V \to D_i$ as in

$$\begin{aligned} \alpha_i(V') &= \{\eta_i(v) \mid v \in V'\} \\ \gamma_i(D_i') &= \{v \mid \eta_i(v) \in D_i'\} \end{aligned}$$

The Galois connection $(\mathcal{P}(V), \alpha, \gamma, \mathcal{P}(D_1 \times D_2))$ has

$$\begin{aligned} \alpha(V') &= \{(\eta_1(v), \eta_2(v)) \mid v \in V'\} \\ \gamma(\mathsf{DD}) &= \{v \mid (\eta_1(v), \eta_2(v)) \in \mathsf{DD}\} \end{aligned}$$

corresponding to the extraction function $\eta : V \to D_1 \times D_2$ defined by

$$\eta(v) = (\eta_1(v), \eta_2(v))$$

Example:

Using the direct tensor product to combine the detection of signs analysis for pairs of integers with the analysis of difference in magnitude.

$$(\mathcal{P}(\mathbf{Z} \times \mathbf{Z}), \alpha_{\mathsf{SSR}'}, \gamma_{\mathsf{SSR}'}, \mathcal{P}(\mathbf{Sign} \times \mathbf{Sign} \times \mathbf{Range}))$$

is given by

$$\begin{array}{rcl} \alpha_{\mathsf{SSR}'}(ZZ) & = & \{(\mathsf{sign}(z_1), \mathsf{sign}(z_2), \mathsf{range}(|z_1| - |z_2|)) \mid (z_1, z_2) \in ZZ\} \\ \gamma_{\mathsf{SSR}'}(SSR) & = & \{(z_1, z_2) \mid (\mathsf{sign}(z_1), \mathsf{sign}(z_2), \mathsf{range}(|z_1| - |z_2|)) \in SSR\} \end{array}$$

corresponding to twosignsrange $: \mathbf{Z} \times \mathbf{Z} \to \mathbf{Sign} \times \mathbf{Sign} \times \mathbf{Range}$ given by

$$\mathsf{twosignsrange}(z_1, z_2) = (\mathsf{sign}(z_1), \mathsf{sign}(z_2), \mathsf{range}(|z_1| - |z_2|))$$

Advantages of the Direct Tensor Product

The expression `(x,3*x)` may have a value in $\{(z, 3 * z) \mid z \in \mathbf{Z}\}$ which in the direct tensor product can be described by

$$\alpha_{\mathsf{SSR'}}(\{(z, 3 * z) \mid z \in \mathbf{Z}\}) = \{(\mathtt{-}, \mathtt{-}, \mathtt{<-1}), (\mathtt{0}, \mathtt{0}, \mathtt{0}), (\mathtt{+}, \mathtt{+}, \mathtt{<-1})\}$$

compared to the direct product that gave

$$\alpha_{\mathsf{SSR}}(\{(z, 3 * z) \mid z \in \mathbf{Z}\}) = (\{(\mathtt{-}, \mathtt{-}), (\mathtt{0}, \mathtt{0}), (\mathtt{+}, \mathtt{+})\}, \{\mathtt{0}, \mathtt{<-1}\})$$

Note that the Galois connection is *not* a Galois insertion because

$$\gamma_{\mathsf{SSR'}}(\emptyset) = \emptyset = \gamma_{\mathsf{SSR'}}(\{(\mathtt{0}, \mathtt{0}, \mathtt{<-1})\})$$

so $\gamma_{\mathsf{SSR'}}$ is not injective and hence we do not have a Galois insertion.

From Direct to Reduced

Reduced Product

Let $(L, \alpha_1, \gamma_1, M_1)$ and $(L, \alpha_2, \gamma_2, M_2)$ be Galois connections.

The *reduced product* is the Galois *insertion*

$$(L, \alpha, \gamma, \varsigma[M_1 \times M_2])$$

defined by

$$\begin{array}{rcl}
\alpha(l) & = & (\alpha_1(l), \alpha_2(l)) \\
\gamma(m_1, m_2) & = & \gamma_1(m_1) \sqcap \gamma_2(m_2) \\
\varsigma(m_1, m_2) & = & \bigsqcap \{(m_1', m_2') \mid \gamma_1(m_1) \sqcap \gamma_2(m_2) = \gamma_1(m_1') \sqcap \gamma_2(m_2')\}
\end{array}$$

Reduced Tensor Product

Let $(\mathcal{P}(V), \alpha_1, \gamma_1, \mathcal{P}(D_1))$ and $(\mathcal{P}(V), \alpha_2, \gamma_2, \mathcal{P}(D_2))$ be Galois connection.

The *reduced tensor product* is the Galois *insertion*

$$(\mathcal{P}(V), \alpha, \gamma, \varsigma[\mathcal{P}(D_1 \times D_2)])$$

defined by

$$\begin{array}{rcl} \alpha(V') & = & \bigcup\{\alpha_1(\{v\}) \times \alpha_2(\{v\}) \mid v \in V'\} \\ \gamma(DD) & = & \{v \mid \alpha_1(\{v\}) \times \alpha_2(\{v\}) \subseteq DD\} \\ \varsigma(DD) & = & \bigcap\{DD' \mid \gamma(DD) = \gamma(DD')\} \end{array}$$

Example: Array Bounds Analysis

The superfluous elements of $\mathcal{P}(\mathbf{Sign} \times \mathbf{Sign} \times \mathbf{Range})$ will be removed when we use a reduced tensor product:

The reduction operator $\varsigma_{\mathsf{SSR}'}$ amounts to

$$\varsigma_{\mathsf{SSR}'}(SSR) = \bigcap\{SSR' \mid \gamma_{\mathsf{SSR}'}(SSR) = \gamma_{\mathsf{SSR}'}(SSR')\}$$

where $SSR, SSR' \subseteq \mathbf{Sign} \times \mathbf{Sign} \times \mathbf{Range}$.

The singleton sets constructed from the following 16 elements

$$\begin{array}{llll}
(-, 0, \texttt{<-1}), & (-, 0, -1), & (-, 0, 0), & \\
(0, -, 0), & (0, -, +1), & (0, -, \texttt{>+1}), & \\
(0, 0, \texttt{<-1}), & (0, 0, -1), & (0, 0, +1), & (0, 0, \texttt{>+1}), \\
(0, +, 0), & (0, +, +1), & (0, +, \texttt{>+1}), & \\
(+, 0, \texttt{<-1}), & (+, 0, -1), & (+, 0, 0) &
\end{array}$$

will be mapped to the empty set (as they are useless).

Example (cont.): Array Bounds Analysis

The remaining 29 elements of $\mathbf{Sign} \times \mathbf{Sign} \times \mathbf{Range}$ are

$$
\begin{array}{lllll}
(-,-,\texttt{<-1}), & (-,-,\texttt{-1}), & (-,-,\texttt{0}), & (-,-,\texttt{+1}), & (-,-,\texttt{>+1}), \\
(-,0,\texttt{+1}), & (-,0,\texttt{>+1}), & & & \\
(-,+,\texttt{<-1}), & (-,+,\texttt{-1}), & (-,+,\texttt{0}), & (-,+,\texttt{+1}), & (-,+,\texttt{>+1}), \\
(0,-,\texttt{<-1}), & (0,-,\texttt{-1}), & (0,0,\texttt{0}), & (0,+,\texttt{<-1}), & (0,+,\texttt{-1}), \\
(+,-,\texttt{<-1}), & (+,-,\texttt{-1}), & (+,-,\texttt{0}), & (+,-,\texttt{+1}), & (+,-,\texttt{>+1}), \\
(+,0,\texttt{+1}), & (+,0,\texttt{>+1}), & & & \\
(+,+,\texttt{<-1}), & (+,+,\texttt{-1}), & (+,+,\texttt{0}), & (+,+,\texttt{+1}), & (+,+,\texttt{>+1})
\end{array}
$$

and they describe disjoint subsets of $\mathbf{Z} \times \mathbf{Z}$.

Any collection of properties can be descibed in 4 bytes.

Summary

The Array Bound Analysis has been designed from three simple Galois connections specified by extraction functions:

(i) an analysis approximating integers by their sign,

(ii) an analysis approximating pairs of integers by their difference in magnitude, and

(iii) an analysis approximating integers by their closeness to 0, 1 and -1.

These analyses have been combined using:

(iv) the relational product of analysis (i) with itself,

(v) the functional composition of analyses (ii) and (iii), and

(vi) the reduced tensor product of analyses (iv) and (v).

Induced Operations

Given: Galois connections $(L_i, \alpha_i, \gamma_i, M_i)$ so that M_i is more approximate than (i.e. is coarser than) L_i.

Aim: Replace an existing analysis over L_i with an analysis making use of the coarser structure of M_i.

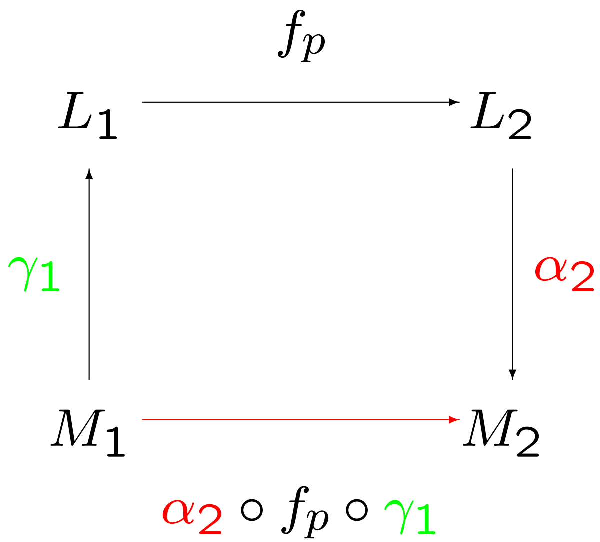
Methods:

- Inducing along the abstraction function: move the computations from L_i to M_i.
- Application to Data Flow Analysis.
- Inducing along the concretisation function: move a widening from M_i to L_i.

Inducing along the Abstraction Function

Given Galois connections $(L_i, \alpha_i, \gamma_i, M_i)$ so that M_i is more approximate than L_i.

Replace an existing analysis $f_p : L_1 \rightarrow L_2$ with a new and more approximate analysis $g_p : M_1 \rightarrow M_2$: take $g_p = \alpha_2 \circ f_p \circ \gamma_1$.



The analysis $\alpha_2 \circ f_p \circ \gamma_1$ is *induced* from f_p and the Galois connections.

Example:

A very precise analysis for `plus` based on $\mathcal{P}(\mathbf{Z})$ and $\mathcal{P}(\mathbf{Z} \times \mathbf{Z})$:

$$f_{\mathsf{plus}}(ZZ) = \{z_1 + z_2 \mid (z_1, z_2) \in ZZ\}$$

Two Galois connections

$$(\mathcal{P}(\mathbf{Z}), \alpha_{\mathsf{sign}}, \gamma_{\mathsf{sign}}, \mathcal{P}(\mathbf{Sign}))$$

$$(\mathcal{P}(\mathbf{Z} \times \mathbf{Z}), \alpha_{\mathsf{SS'}}, \gamma_{\mathsf{SS'}}, \mathcal{P}(\mathbf{Sign} \times \mathbf{Sign}))$$

An approximate analysis for `plus` based on $\mathcal{P}(\mathbf{Sign})$ and $\mathcal{P}(\mathbf{Sign} \times \mathbf{Sign})$:

$$g_{\mathsf{plus}} = \alpha_{\mathsf{sign}} \circ f_{\mathsf{plus}} \circ \gamma_{\mathsf{SS'}}$$

Example (cont.):

We calculate

$$
\begin{aligned}
g_{\text{plus}}(SS) &= \alpha_{\text{sign}}(f_{\text{plus}}(\gamma_{\text{SS}'}(SS))) \\
&= \alpha_{\text{sign}}(f_{\text{plus}}(\{(z_1, z_2) \in \mathbf{Z} \times \mathbf{Z} \mid (\text{sign}(z_1), \text{sign}(z_2)) \in SS\})) \\
&= \alpha_{\text{sign}}(\{z_1 + z_2 \mid z_1, z_2 \in \mathbf{Z}, (\text{sign}(z_1), \text{sign}(z_2)) \in SS\}) \\
&= \{\text{sign}(z_1 + z_2) \mid z_1, z_2 \in \mathbf{Z}, (\text{sign}(z_1), \text{sign}(z_2)) \in SS\} \\
&= \bigcup\{s_1 \oplus s_2 \mid (s_1, s_2) \in SS\}
\end{aligned}
$$

where $\oplus : \mathbf{Sign} \times \mathbf{Sign} \to \mathcal{P}(\mathbf{Sign})$ is the "addition" operator on signs (so e.g. $\texttt{+} \oplus \texttt{+} = \{\texttt{+}\}$ and $\texttt{+} \oplus \texttt{-} = \{\texttt{-}, 0, \texttt{+}\}$).

The Mundane Correctness of f_p carries over to g_p

The correctness relation R_i for V_i and L_i:

$$R_i : V_i \times L_i \to \{\textit{true}, \textit{false}\} \text{ is } \textit{generated by } \beta_i : V_i \to L_i$$

Correctness of f_p means

$$(p \vdash \cdot \leadsto \cdot)\,(R_1 \twoheadrightarrow R_2)\, f_p$$

(with $R_1 \twoheadrightarrow R_2$ being generated by $\beta_1 \twoheadrightarrow \beta_2$).

The correctness relation S_i for V_i and M_i:

$$S_i : V_i \times M_i \to \{\textit{true}, \textit{false}\} \text{ is } \textit{generated by } \alpha_i \circ \beta_i : V_i \to M_i$$

One can prove that

$$(p \vdash \cdot \leadsto \cdot)\,(R_1 \twoheadrightarrow R_2)\, f_p \;\wedge\; \alpha_2 \circ f_p \circ \gamma_1 \sqsubseteq g_p$$
$$\Rightarrow\; (p \vdash \cdot \leadsto \cdot)\,(S_1 \twoheadrightarrow S_2)\, g_p$$

with $S_1 \twoheadrightarrow S_2$ being generated by $(\alpha_1 \circ \beta_1) \twoheadrightarrow (\alpha_2 \circ \beta_2)$.

Fixed Points in the Induced Analysis

Let $f_p = \mathit{lfp}(F)$ for a monotone function $F : (L_1 \to L_2) \to (L_1 \to L_2)$.

The Galois connections $(L_i, \alpha_i, \gamma_i, M_i)$ give rise to a Galois connection $(L_1 \to L_2, \alpha, \gamma, M_1 \to M_2)$.

Take $g_p = \mathit{lfp}(G)$ where $G : (M_1 \to M_2) \to (M_1 \to M_2)$ is an "upper approximation" to F: we demand that $\alpha \circ F \circ \gamma \sqsubseteq G$.

Then for all $m \in M_1 \to M_2$:

$$G(m) \sqsubseteq m \Rightarrow F(\gamma(m)) \sqsubseteq \gamma(m)$$

and $\mathit{lfp}(F) \sqsubseteq \gamma(\mathit{lfp}(G))$ and $\alpha(\mathit{lfp}(F)) \sqsubseteq \mathit{lfp}(G)$

Application to Data Flow Analysis

A *generalised Monotone Framework* consists of:

- the property space: a complete lattice $L = (L, \sqsubseteq)$;
- the set $\mathcal{F}$ of monotone functions from L to L.

An *instance* A of a generalised Monotone Framework consists of:

- a finite flow, $F \subseteq \mathbf{Lab} \times \mathbf{Lab}$;
- a finite set of extremal labels, $E \subseteq \mathbf{Lab}$;
- an extremal value, $\iota \in L$; and
- a mapping $f_{\cdot}$ from the labels $\mathbf{Lab}$ of F and E to monotone transfer functions from L to L.

Application to Data Flow Analysis

Let (L, α, γ, M) be a Galois connection.

Consider an instance B of the generalised Monotone Framework M that satisfies

- the mapping $g_\cdot$ from the labels $\mathbf{Lab}$ of F and E to monotone transfer functions of $M \to M$ satisfies $g_\ell \;\sqsupseteq\; \alpha \circ f_\ell \circ \gamma$ for all ℓ; and
- the extremal value $\jmath$ satisfies $\gamma(\jmath) = \iota$;

and otherwise B is as A.

One can show that a solution to the B-constraints gives rise to a solution to the A-constraints:

$$(B_\circ, B_\bullet) \models \mathsf{B}^{\sqsupseteq} \quad \text{implies} \quad (\gamma \circ B_\circ, \gamma \circ B_\bullet) \models \mathsf{A}^{\sqsupseteq}$$

The Mundane Approach to Semantic Correctness

Here $F = \mathit{flow}(S_\star)$ and $E = \{\mathit{init}(S_\star)\}$.

Correctness of every solution to $\mathsf{A}^{\sqsupseteq}$ amounts to:

> Assume $(A_\circ, A_\bullet) \models \mathsf{A}^{\sqsupseteq}$ and $\langle S_\star, \sigma_1 \rangle \to^* \sigma_2$.
>
> Then $\beta(\sigma_1) \sqsubseteq \iota$ implies $\beta(\sigma_2) \sqsubseteq \bigsqcup\{A_\bullet(\ell) \mid \ell \in \mathit{final}(S_\star)\}$.

where $\beta : \mathbf{State} \to L$.

One can then prove the correctness result for B:

> Assume $(B_\circ, B_\bullet) \models \mathsf{B}^{\sqsupseteq}$ and $\langle S_\star, \sigma_1 \rangle \to^* \sigma_2$.
>
> Then $(\alpha \circ \beta)(\sigma_1) \sqsubseteq \jmath$ implies $(\alpha \circ \beta)(\sigma_2) \sqsubseteq \bigsqcup\{B_\bullet(\ell) \mid \ell \in \mathit{final}(S_\star)\}$.

Sets of States Analysis

Generalised Monotone Framework over $(\mathcal{P}(\mathbf{State}), \subseteq)$.
Instance SS for $S_\star$:

- the flow F is *flow*$(S_\star)$;
- the set E of extremal labels is $\{$*init*$(S_\star)\}$;
- the extremal value ι is $\mathbf{State}$; and
- the transfer functions are given by $f^{\mathsf{SS}}_{\cdot}$:

$$
\begin{array}{lrcl}
[x := a]^\ell & f^{\mathsf{SS}}_\ell(\Sigma) & = & \{\sigma[x \mapsto \mathcal{A}[\![a]\!]\sigma] \mid \sigma \in \Sigma\} \\
[\texttt{skip}]^\ell & f^{\mathsf{SS}}_\ell(\Sigma) & = & \Sigma \\
[b]^\ell & f^{\mathsf{SS}}_\ell(\Sigma) & = & \Sigma
\end{array}
$$

where $\Sigma \subseteq \mathbf{State}$.

Correctness: Assume $(\mathsf{SS}_\circ, \mathsf{SS}_\bullet) \models \mathsf{SS}^{\supseteq}$ and $\langle S_\star, \sigma_1\rangle \to^* \sigma_2$.
Then $\sigma_1 \in \mathbf{State}$ implies $\sigma_2 \in \bigcup\{\mathsf{SS}_\bullet(\ell) \mid \ell \in$ *final*$(S_\star)\}$.

Constant Propagation Analysis

Generalised Monotone Framework over $\widehat{\mathbf{State}}_{\mathsf{CP}} = ((\mathbf{Var} \to \mathbf{Z}^{\top})_{\bot}, \sqsubseteq)$.
Instance CP for $S_\star$:

- the flow F is *flow*$(S_\star)$;
- the set E of extremal labels is $\{$*init*$(S_\star)\}$;
- the extremal value ι is $\lambda x.\top$; and
- the transfer functions are given by the mapping $f^{\mathsf{CP}}_{\cdot}$:

$$
\begin{array}{lrcl}
[x := a]^{\ell}: & f^{\mathsf{CP}}_{\ell}(\widehat{\sigma}) & = & \begin{cases} \bot & \text{if } \widehat{\sigma} = \bot \\ \widehat{\sigma}[x \mapsto \mathcal{A}_{\mathsf{CP}}[\![a]\!]\widehat{\sigma}] & \text{otherwise} \end{cases} \\
[\mathtt{skip}]^{\ell}: & f^{\mathsf{CP}}_{\ell}(\widehat{\sigma}) & = & \widehat{\sigma} \\
[b]^{\ell}: & f^{\mathsf{CP}}_{\ell}(\widehat{\sigma}) & = & \widehat{\sigma}
\end{array}
$$

Galois Connection

The representation function $\beta_{\mathsf{CP}} : \mathbf{State} \to \widehat{\mathbf{State}}_{\mathsf{CP}}$ is defined by

$$\beta_{\mathsf{CP}}(\sigma) = \sigma$$

This gives rise to a Galois connection

$$(\mathcal{P}(\mathbf{State}), \alpha_{\mathsf{CP}}, \gamma_{\mathsf{CP}}, \widehat{\mathbf{State}}_{\mathsf{CP}})$$

where $\alpha_{\mathsf{CP}}(\Sigma) = \bigsqcup\{\beta_{\mathsf{CP}}(\sigma) \mid \sigma \in \Sigma\}$ and $\gamma_{\mathsf{CP}}(\widehat{\sigma}) = \{\sigma \mid \beta_{\mathsf{CP}}(\sigma) \sqsubseteq \widehat{\sigma}\}$.

One can show that for all labels ℓ

$f_\ell^{\mathsf{CP}} \sqsupseteq \alpha_{\mathsf{CP}} \circ f_\ell^{\mathsf{SS}} \circ \gamma_{\mathsf{CP}}$ as well as $\gamma_{\mathsf{CP}}(\lambda x.\top) = \mathbf{State}$

It follows that CP is an upper approximation to the analysis induced from SS and the Galois connection; therefore it is correct.

Inducing along the Concretisation Function

Given an upper bound operator

$$\nabla_M : M \times M \to M$$

and a Galois connection (L, α, γ, M).

Define an upper bound operator

$$\nabla_L : L \times L \to L$$

by

$$l_1 \ \nabla_L \ l_2 = \gamma(\ \alpha(l_1) \ \nabla_M \ \alpha(l_2) \ )$$

It defines a widening operator if one of the following conditions holds:

(i) M satisfies the Ascending Chain Condition, or

(ii) (L, α, γ, M) is a Galois insertion and $\nabla_M : M \times M \to M$ is a widening.

Precision of the Induced Widening Operator

Lemma: Let (L, α, γ, M) be a Galois insertion such that $\gamma(\bot_M) = \bot_L$ and let $\nabla_M : M \times M \to M$ be a widening operator.

Then the widening operator $\nabla_L : L \times L \to L$ defined by

$$l_1 \ \nabla_L \ l_2 = \gamma(\alpha(l_1) \ \nabla_M \ \alpha(l_2))$$

satisfies

$$\mathit{lfp}_{\nabla_L}(f) = \gamma(\mathit{lfp}_{\nabla_M}(\alpha \circ f \circ \gamma))$$

for all monotone functions $f : L \to L$.

Precision of the Induced Widening Operator

Corollary: Let M be of finite height, let (L, α, γ, M) be a Galois insertion (such that $\gamma(\bot_M) = \bot_L$), and let ∇_M equal the least upper bound operator $\sqcup_M$.

Then the above lemma shows that $\mathit{lfp}_{\nabla_L}(f) = \gamma(\mathit{lfp}(\alpha \circ f \circ \gamma))$.

This means that $\mathit{lfp}_{\nabla_L}(f)$ *equals* the result we would have obtained if we decided to work with $\alpha \circ f \circ \gamma : M \to M$ instead of the given $f : L \to L$; furthermore the number of iterations needed turn out to be the same. However, for all other operations the increased precision of L is available.