Principles of Program Analysis:

Abstract Interpretation

A Mundane Approach to Semantic Correctness

Semantics:

\[ p \vdash v_1 \sim v_2 \]

where \( v_1, v_2 \in V \).

Note: \( \sim \) might be deterministic.

Program analysis:

\[ p \vdash l_1 \triangleright l_2 \]

where \( l_1, l_2 \in L \).

Note: \( \triangleright \) 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: \( V = \text{State} \)

Transitions:

\[
S_\star \vdash \sigma_1 \sim \sigma_2
\]

iff

\[
\langle S_\star, \sigma_1 \rangle \rightarrow^* \sigma_2
\]

Constant Propagation Analysis:

Properties:

\[
L = \hat{\text{State}}_{\text{CP}} = (\text{Var}_\star \rightarrow \mathbb{Z}^\top)_\perp
\]

Transitions:

\[
S_\star \vdash \hat{\sigma}_1 \triangleright \hat{\sigma}_2
\]

iff

\[
\hat{\sigma}_1 = \iota
\]

\[
\hat{\sigma}_2 = \bigsqcup \{ \text{CP}\bullet(\ell) \mid \ell \in \text{final}(S_\star) \}
\]

\[
(\text{CP}_\circ, \text{CP}_\bullet) \models \text{CP}=(S_\star)
\]
Example: Control Flow Analysis

Structural Operational Semantics:

Values: $V = \text{Val}$

Transitions:

\begin{align*}
\mathbf{e} \vdash v_1 \mapsto v_2 \\
\text{iff}
\end{align*}

\begin{align*}
\text{[ ]} \vdash (\mathbf{e} v^{l_1}_{l_1})^l_2 \rightarrow^* v^{l_2}_{l_2}
\end{align*}

Pure 0-CFA Analysis:

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

Transitions:

\begin{align*}
\mathbf{e} \vdash (\widehat{\rho}_1, \widehat{v}_1) \triangleright (\widehat{\rho}_2, \widehat{v}_2) \\
\text{iff}
\end{align*}

\begin{align*}
\widehat{C}(l_1) &= \widehat{v}_1 \\
\widehat{C}(l_2) &= \widehat{v}_2 \\
\widehat{\rho}_1 &= \widehat{\rho}_2 = \widehat{\rho} \\
(\widehat{C}, \widehat{\rho}) \models (\mathbf{e} c^{l_1})^l_2
\end{align*}

for some placeholder constant $c$
Correctness Relations

\[ R : V \times L \rightarrow \{ \text{true, false} \} \]

Idea: \( v \ R \ l \) means that the value \( v \) is described by the property \( l \).

Correctness criterion: \( R \) is preserved under computation:

\[
\begin{align*}
p \vdash v_1 \quad &\sim\!

\vdash \quad v_2 \\
\vdash \quad &\vdash \quad v_2 \\
R \quad &\Rightarrow \quad R \\
\vdash \quad &\vdash \quad R \\
\vdash \quad &\vdash \quad R \\
p \vdash l_1 \quad &\triangleright \quad l_2
\end{align*}
\]

logical relation:

\[(p \vdash \sim) \quad (R \rightarrow R) \quad (p \vdash \triangleright)\]
Admissible Correctness Relations

\[ v R l_1 \land l_1 \sqsubseteq l_2 \Rightarrow v R l_2 \]

\[(\forall l \in L' \subseteq L : v R l) \Rightarrow v R (\bigcap L') \text{ (\{l | v R l\} is a Moore family)}\]

Two consequences:

\[ v R \top \]

\[ v R l_1 \land v R l_2 \Rightarrow v R (l_1 \cap l_2) \]

Assumption: \((L, \sqsubseteq)\) is a complete lattice.
Example: Data Flow Analysis

Correctness relation

\[ R_{CP} : \text{State} \times \hat{\text{State}}_{CP} \rightarrow \{true, false\} \]

is defined by

\[ \sigma \ R_{CP} \hat{\sigma} \iff \forall x \in FV(S_\tau) : (\hat{\sigma}(x) = \top \lor \sigma(x) = \hat{\sigma}(x)) \]
Example: Control Flow Analysis

Correctness relation

\[ R_{CFA} : \text{Val} \times (\hat{\text{Env}} \times \hat{\text{Val}}) \to \{\text{true, false}\} \]

is defined by

\[ v \ R_{CFA} (\hat{\rho}, \hat{v}) \iff v \ \mathcal{V} (\hat{\rho}, \hat{v}) \]

where \( \mathcal{V} \) is given by:

\[ v \ \mathcal{V} (\hat{\rho}, \hat{v}) \iff \begin{cases} \text{true} & \text{if } v = c \\ t \in \hat{v} \land \forall x \in \text{dom}(\rho) : \rho(x) \ \mathcal{V} (\hat{\rho}, \hat{\rho}(x)) & \text{if } v = \text{close } t \ \text{in } \rho \end{cases} \]
Representation Functions

\[ \beta : V \rightarrow L \]

Idea: \( \beta \) maps a value to the best property describing it.

Correctness criterion:

\[
\begin{align*}
p & \vdash v_1 & \sim & v_2 \\
\beta & \downarrow & \Rightarrow & \beta \\
\sqcap & \downarrow & \sqcap & \\
p & \vdash l_1 & \triangleright & l_2
\end{align*}
\]
Equivalence of Correctness Criteria

Given a representation function $\beta$ we define a correctness relation $R_{\beta}$ by

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

Given a correctness relation $R$ we define a representation function $\beta_R$ by

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

**Lemma:**

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

(ii) Given an admissible correctness relation $R : V \times L \to \{true, false\}$, then $\beta_R$ is well-defined and $R_{\beta_R} = R$. 
Equivalence of Criteria: \( R \) is \textit{generated} by \( \beta \)
Example: Data Flow Analysis

Representation function

\[ \beta_{\text{CP}} : \text{State} \rightarrow \hat{\text{State}}_{\text{CP}} \]

is defined by

\[ \beta_{\text{CP}}(\sigma) = \lambda x.\sigma(x) \]

\( R_{\text{CP}} \) is generated by \( \beta_{\text{CP}} \):

\[ \sigma \ R_{\text{CP}} \hat{\sigma} \quad \text{iff} \quad \beta_{\text{CP}}(\sigma) \sqsubseteq_{\text{CP}} \hat{\sigma} \]
Example: Control Flow Analysis

Representation function

\[ \beta_{\text{CFA}} : \text{Val} \to \hat{\text{Env}} \times \hat{\text{Val}} \]

is defined by

\[
\beta_{\text{CFA}}(v) = \begin{cases}
(\lambda x. \emptyset, \emptyset) & \text{if } v = c \\
(\beta_{\text{CFA}}^E(\rho), \{t\}) & \text{if } v = \text{close} \ t \ \text{in} \ \rho
\end{cases}
\]

\[ \beta_{\text{CFA}}^E(\rho)(x) = \bigcup \{ \hat{\rho}_y(x) \mid \beta_{\text{CFA}}(\rho(y)) = (\hat{\rho}_y, \hat{v}_y) \ \text{and} \ y \in \text{dom}(\rho) \} \]

\[
\bigcup \left\{ \{ \hat{v}_x \} \mid x \in \text{dom}(\rho) \ \text{and} \ \beta_{\text{CFA}}(\rho(x)) = (\hat{\rho}_x, \hat{v}_x) \right\}
\]

\[ \emptyset \quad \text{otherwise} \]

\[ R_{\text{CFA}} \text{ is generated by } \beta_{\text{CFA}}: \]

\[ v \ R_{\text{CFA}} (\hat{\rho}, \hat{v}) \quad \text{iff} \quad \beta_{\text{CFA}}(v) \sqsubseteq_{\text{CFA}} (\hat{\rho}, \hat{v}) \]
A Modest Generalisation

Semantics:

\[ p \vdash v_1 \sim v_2 \]

where \( v_1 \in V_1, v_2 \in V_2 \)

Program analysis:

\[ p \vdash l_1 \triangleright l_2 \]

where \( l_1 \in L_1, l_2 \in L_2 \)

logical relation:

\[(p \vdash \cdot \sim \cdot) (R_1 \Rightarrow R_2) (p \vdash \cdot \triangleright \cdot)\]
Higher-Order Formulation

Assume that

- \( R_1 \) is an admissible correctness relation for \( V_1 \) and \( L_1 \) that is \textit{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 \textit{generated by} the representation function \( \beta_2 : V_2 \to L_2 \)

Then the relation \( R_1 \xrightarrow{\sim} R_2 \) is an admissible correctness relation for \( V_1 \to V_2 \) and \( L_1 \to L_2 \) that is \textit{generated by} the representation function \( \beta_1 \xrightarrow{\sim} \beta_2 \) defined by

\[
(\beta_1 \xrightarrow{\sim} \beta_2)(\sim) = \lambda l_1. \bigcup \{ \beta_2(v_2) \mid \beta_1(v_1) \sqsubseteq l_1 \wedge v_1 \sim v_2 \}
\]
Example:

Semantics:

\[ \text{plus} \vdash (z_1, z_2) \leadsto z_1 + z_2 \]

where \( z_1, z_2 \in \mathbb{Z} \)

Program analysis:

\[ \text{plus} \vdash \mathbb{Z} \rightarrow \{ z_1 + z_2 \mid (z_1, z_2) \in \mathbb{Z} \} \]

where \( \mathbb{Z} \subseteq \mathbb{Z} \times \mathbb{Z} \)

<table>
<thead>
<tr>
<th>Correctness relations</th>
<th>Representation functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
<td>( R_\mathbb{Z} )</td>
</tr>
<tr>
<td>( \beta_\mathbb{Z} )</td>
<td>( { z } )</td>
</tr>
<tr>
<td>argument</td>
<td>( R_{\mathbb{Z} \times \mathbb{Z}} )</td>
</tr>
<tr>
<td>( \beta_{\mathbb{Z} \times \mathbb{Z}} )</td>
<td>( { (z_1, z_2) } )</td>
</tr>
<tr>
<td>plus</td>
<td>( (\text{plus} \vdash \cdot \leadsto \cdot) )</td>
</tr>
<tr>
<td>( (R_{\mathbb{Z} \times \mathbb{Z}} \rightarrow R_\mathbb{Z}) )</td>
<td>( (\beta_{\mathbb{Z} \times \mathbb{Z}} \rightarrow \beta_\mathbb{Z})(\text{plus} \vdash \cdot \leadsto \cdot) )</td>
</tr>
<tr>
<td>( (\text{plus} \vdash \cdot \triangleright \cdot) )</td>
<td>( (\text{plus} \vdash \cdot \triangleright \cdot) )</td>
</tr>
</tbody>
</table>
Approximation of Fixed Points

- Fixed points
- Widening
- Narrowing

Example: lattice of intervals for *Array Bound Analysis*
The complete lattice $\text{Interval} = (\text{Interval}, \sqsubseteq)$
Fixed points

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

- $l$ is a \textit{fixed point} iff $f(l) = l$
- $f$ is \textit{reductive} at $l$ iff $f(l) \sqsubseteq l$
- $f$ is \textit{extensive} at $l$ iff $f(l) \supseteq l$

\textbf{Tarski’s Theorem} ensures that

- $lfp(f) = \bigcap \text{Fix}(f) = \bigcap \text{Red}(f) \in \text{Fix}(f) \subseteq \text{Red}(f)$
- $gfp(f) = \bigcup \text{Fix}(f) = \bigcup \text{Ext}(f) \in \text{Fix}(f) \subseteq \text{Ext}(f)$
Fixed points of $f$

- $\text{Red}(f)$
- $\text{Fix}(f)$
- $\text{Ext}(f)$

- $\top$
- $f^n(\top)$
- $\bigcap_n f^n(\top)$
- $\text{gfp}(f)$
- $\text{lfp}(f)$
- $\bigcup_n f^n(\bot)$
- $f^n(\bot)$
- $\bot$
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 \((f^n_{\nabla})_n\) 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 \(\nabla\): an upper bound operator satisfying a finiteness condition.
Upper bound operators

\[ \uplus : L \times L \to L \] is an upper bound operator iff

\[ l_1 \sqsubseteq l_1 \uplus l_2 \sqsupseteq l_2 \]

for all \( l_1, l_2 \in L \).

Let \( (l_n) \) be a sequence of elements of \( L \). Define the sequence \( (l_n) \) by:

\[
l_n = \begin{cases} 
l_n & \text{if } n = 0 \\
\uplus_{n-1} l_n & \text{if } n > 0
\end{cases}
\]

**Fact:** If \( (l_n) \) is a sequence and \( \uplus \) is an upper bound operator then \( (l_n) \) is an ascending chain; furthermore \( l_n \sqsubseteq \bigcup \{l_0, l_1, \ldots, l_n\} \) for all \( n \).
Example:

Let \( \text{int} \) be an arbitrary but fixed element of \textbf{Interval}.

An upper bound operator:

\[
\text{int}_1 \uparrow^{\text{int}} \text{int}_2 = \begin{cases} 
\text{int}_1 \cup \text{int}_2 & \text{if } \text{int}_1 \sqsubseteq \text{int} \lor \text{int}_2 \sqsubseteq \text{int}_1 \\
[-\infty, \infty] & \text{otherwise}
\end{cases}
\]

Example: \([1, 2] \uparrow^{[0,2]} [2, 3] = [1, 3] \) and \([2, 3] \uparrow^{[0,2]} [1, 2] = [-\infty, \infty] \).

Transformation of: \([0, 0], [1, 1], [2, 2], [3, 3], [4, 4], [5, 5], \ldots\)

If \( \text{int} = [0, \infty] \): \([0, 0], [0, 1], [0, 2], [0, 3], [0, 4], [0, 5], \ldots\)

If \( \text{int} = [0, 2] \): \([0, 0], [0, 1], [0, 2], [0, 3], [-\infty, \infty], [-\infty, \infty], \ldots\)
Widening operators

An operator $\nabla : L \times L \rightarrow L$ 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 $f : L \rightarrow L$ and a widening operator $\nabla$, define the sequence $(f^n \nabla)_n$ by

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

One can show that:

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

\[ \text{lfp}_\nabla(f) = f^m \]
The widening operator $\nabla$ applied to $f$

\[
\begin{align*}
Red(f) & \rightarrow \cdots \rightarrow \nabla f^1 \\
\nabla f^2 & \rightarrow \cdots \rightarrow \nabla f^m = \nabla f^{m+1} = lfp_f
\end{align*}
\]
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 $\nabla$ based on $K$.

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

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

where

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

PPA Section 4.2  © F.Nielson & H.Riis Nielson & C.Hankin (Dec. 2004)
Example (cont.) — formalisation:

Let \( z_i \in \mathbb{Z}' = \mathbb{Z} \cup \{-\infty, \infty\} \) and write:

\[
\begin{align*}
LB_K(z_1, z_3) &= \begin{cases} 
z_1 & \text{if } z_1 \leq z_3 \\
k & \text{if } z_3 < z_1 \land k = \max\{k \in K \mid k \leq z_3\} \\
-\infty & \text{if } z_3 < z_1 \land \forall k \in K : z_3 < k
\end{cases} \\
UB_K(z_2, z_4) &= \begin{cases} 
z_2 & \text{if } z_4 \leq z_2 \\
k & \text{if } z_2 < z_4 \land k = \min\{k \in K \mid z_4 \leq k\} \\
\infty & \text{if } z_2 < z_4 \land \forall k \in K : k < z_4
\end{cases}
\end{align*}
\]

\[\text{int}_1 \triangledown \text{int}_2 = \begin{cases} 
\bot & \text{if } \text{int}_1 = \text{int}_2 = \bot \\
[ LB_K(\inf(\text{int}_1), \inf(\text{int}_2)), UB_K(\sup(\text{int}_1), \sup(\text{int}_2)) ] & \text{otherwise}
\end{cases}\]
Example (cont.):

Consider the ascending chain \((\text{int}_n)_n\)

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

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

Then \((\text{int}^\nabla_n)_n\) is the chain

\[ [0, 1], [0, 3], [0, 3], [0, 5], [0, 5], [0, \infty], [0, \infty], \ldots \]

which eventually stabilises.
Narrowing Operators

Status: Widening gives us an upper approximation $lfp_\nabla(f)$ of the least fixed point of $f$.

Observation: $f(lfp_\nabla(f)) \subseteq lfp_\nabla(f)$ so the approximation can be improved by considering the iterative sequence $(f^n(lfp_\nabla(f)))_n$.

It will satisfy $f^n(lfp_\nabla(f)) \supseteq lfp(f)$ 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 $\Delta : L \times L \rightarrow L$ is a *narrowing operator* iff

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

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

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

We construct the sequence \( ([f]_\Delta^n)_n \)

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

One can show that:

- \( ([f]_\Delta^n)_n \) is a descending chain where all elements satisfy \( \text{lfp}(f) \sqsubseteq [f]_\Delta^n \)

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

\[
\triangle \text{lfp}(f) = [f]_{\Delta}^{m'}
\]
The narrowing operator $\triangle$ applied to $f$

\[
[f]_\triangle = \text{lfp}_\triangledown (f)
\]

\[
[f]_\triangle^0 = \text{lfp}_\triangledown (f)
\]

\[
[f]_\triangle^1
\]

\[
[f]_\triangle^{m'-1}
\]

\[
[f]_\triangle^m = [f]_\triangle^{m'+1} = \text{lfp}_\triangledown
\]
Example:

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

- those with elements of the form \([-\infty, z], z \in \mathbb{Z}\)
- those with elements of the form \([z, \infty], z \in \mathbb{Z}\)

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

\([z_1, \infty], [z_2, \infty], [z_3, \infty], \ldots\)

(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

$$\begin{align*}
\text{int}_1 \triangle \text{int}_2 &= \begin{cases} 
\bot & \text{if } \text{int}_1 = \bot \lor \text{int}_2 = \bot \\
[z_1, z_2] & \text{otherwise}
\end{cases}
\end{align*}$$

where

$$
\begin{align*}
z_1 &= \begin{cases}
\inf(\text{int}_1) & \text{if } N < \inf(\text{int}_2) \land \sup(\text{int}_2) = \infty \\
\inf(\text{int}_2) & \text{otherwise}
\end{cases} \\
z_2 &= \begin{cases}
\sup(\text{int}_1) & \text{if } \inf(\text{int}_2) = -\infty \land \sup(\text{int}_2) < -N \\
\sup(\text{int}_2) & \text{otherwise}
\end{cases}
\end{align*}
$$
Example (cont.):

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

\[ [0, \infty], [1, \infty], [2, \infty], [3, \infty], [4, \infty], [5, \infty], \ldots \]

and assume that \( N = 3 \).

Then the narrowing operator \( \Delta_N \) will give the sequence \( ([n, \infty]^\Delta)_n \)

\[ [0, \infty], [1, \infty], [2, \infty], [3, \infty], [3, \infty], [3, \infty], \ldots \]
Galois Connections

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

\[
\begin{array}{c}
L \xrightarrow{\gamma} M \\
\alpha \\
\end{array}
\]

\(\alpha: \text{abstraction function}\)

\(\gamma: \text{concretisation function}\)

is a **Galois connection** if and only if

\[\gamma \circ \alpha \sqsupseteq \lambda l.l\]

\[\alpha \circ \gamma \sqsubseteq \lambda m.m\]
Galois connections

\[ \gamma(m) \quad \alpha(l) \]

\[ L \quad M \]

\[ \gamma \circ \alpha \sqsubseteq \lambda l \cdot l \]

\[ \alpha \circ \gamma \sqsubseteq \lambda m \cdot m \]
Example:

Galois connection

\[(\mathcal{P}(\mathbb{Z}), \alpha_{\text{ZI}}, \gamma_{\text{ZI}}, \text{Interval})\]

with concretisation function

\[\gamma_{\text{ZI}}(\text{int}) = \{z \in \mathbb{Z} \mid \inf(\text{int}) \leq z \leq \sup(\text{int})\}\]

and abstraction function

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

Examples:

\[\gamma_{\text{ZI}}([0, 3]) = \{0, 1, 2, 3\}\]
\[\gamma_{\text{ZI}}([0, \infty]) = \{z \in \mathbb{Z} \mid z \geq 0\}\]
\[\alpha_{\text{ZI}}(\{0, 1, 3\}) = [0, 3]\]
\[\alpha_{\text{ZI}}(\{2 \cdot z \mid z > 0\}) = [2, \infty]\]
Adjunctions

\[ L \xrightarrow{\alpha} M \xleftarrow{\gamma} \]

is an \emph{adjunction} if and only if

\[ \alpha : L \to M \text{ and } \gamma : M \to L \text{ are total functions} \]

that satisfy

\[ \alpha(l) \sqsubseteq m \iff l \sqsubseteq \gamma(m) \]

for all \( l \in L \) and \( m \in M \).

**Proposition:** \((\alpha, \gamma)\) is an adjunction iff it is a Galois connection.
Galois connections from representation functions

A representation function $\beta : V \rightarrow L$ gives rise to a Galois connection $(\mathcal{P}(V), \alpha, \gamma, L)$

where

$\alpha(V') = \bigsqcup \{ \beta(v) \mid v \in V' \}$

$\gamma(l) = \{ v \in V \mid \beta(v) \sqsubseteq l \}$

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

This indeed defines an adjunction:

$\alpha(V') \sqsubseteq l \iff \bigsqcup \{ \beta(v) \mid v \in V' \} \sqsubseteq l$

$\iff \forall v \in V' : \beta(v) \sqsubseteq l$

$\iff V' \subseteq \gamma(l)$
Galois connections from extraction functions

An extraction function

\[ \eta : V \rightarrow D \]

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

It gives rise to a representation function \( \beta_\eta : V \rightarrow \mathcal{P}(D) \) (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

\[ \alpha_\eta(V') = \bigcup \{ \beta_\eta(v) \mid v \in V' \} = \{ \eta(v) \mid v \in V' \} \]

\[ \gamma_\eta(D') = \{ v \in V \mid \beta_\eta(v) \subseteq D' \} = \{ v \mid \eta(v) \in D' \} \]
Example:

Extraction function

\[ \text{sign} : \mathbb{Z} \rightarrow \text{Sign} \]

specified by

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

Galois connection

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

with

\[
\alpha_{\text{sign}}(Z) = \{\text{sign}(z) \mid z \in \mathbb{Z}\}
\]

\[
\gamma_{\text{sign}}(S) = \{z \in \mathbb{Z} \mid \text{sign}(z) \in S\}
\]
Properties of Galois Connections

**Lemma:** If \((L, \alpha, \gamma, M)\) is a Galois connection then:

- \(\alpha\) uniquely determines \(\gamma\) by \(\gamma(m) = \bigcup \{l \mid \alpha(l) \sqsubseteq m\}\)
- \(\gamma\) uniquely determines \(\alpha\) by \(\alpha(l) = \bigcap \{m \mid l \sqsubseteq \gamma(m)\}\)
- \(\alpha\) is completely additive and \(\gamma\) is completely multiplicative

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

**Lemma:**

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

**Fact:** If \((L, \alpha, \gamma, M)\) is a Galois connection then

- \(\alpha \circ \gamma \circ \alpha = \alpha\) and \(\gamma \circ \alpha \circ \gamma = \gamma\)
Example:

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

\begin{align*}
\gamma_{IS}(\{-, 0, +\}) &= [-\infty, \infty] & \gamma_{IS}(\{-, 0\}) &= [-\infty, 0] \\
\gamma_{IS}(\{-, +\}) &= [-\infty, \infty] & \gamma_{IS}(\{0, +\}) &= [0, \infty] \\
\gamma_{IS}(\{-\}) &= [-\infty, -1] & \gamma_{IS}(\{0\}) &= [0, 0] \\
\gamma_{IS}(\{+\}) &= [1, \infty] & \gamma_{IS}(\emptyset) &= \bot
\end{align*}

Does there exist an abstraction function

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

such that $(\text{Interval}, \alpha_{IS}, \gamma_{IS}, \mathcal{P}(\text{Sign}))$ is a Galois connection?
Example (cont.):

Is $\gamma_{IS}$ 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 $\gamma : M \rightarrow L$ is completely multiplicative if and only if the following hold:
- $\gamma : M \rightarrow L$ is monotone,
- $\gamma(T) = T$, and
- $\gamma(m_1 \sqcap m_2) = \gamma(m_1) \sqcap \gamma(m_2)$ whenever $m_1 \not\sqsubseteq m_2 \land m_2 \not\sqsubseteq m_1$

We calculate
\[
\begin{align*}
\gamma_{IS}(\{-,0\} \sqcap \{-,+,\}) &= \gamma_{IS}(\{-\}) = [-\infty, -1] \\
\gamma_{IS}(\{-,0\}) \sqcap \gamma_{IS}(\{-,+,\}) &= [-\infty, 0] \sqcap [-\infty, \infty] = [-\infty, 0]
\end{align*}
\]
showing that there is no Galois connection involving $\gamma_{IS}$. 

PPA Section 4.3 © F.Nielson & H.Riis Nielson & C.Hankin (Dec. 2004)
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

- \( R : V \times L \rightarrow \{ \text{true}, \text{false} \} \) is an admissible correctness relation
- \( (L, \alpha, \gamma, M) \) is a Galois connection

Then \( S : V \times M \rightarrow \{ \text{true}, \text{false} \} \) defined by

\[ v S m \iff v R (\gamma(m)) \]

is an admissible correctness relation between \( V \) and \( M \)
The mundane approach: representation functions

Assume

- $R : V \times L \rightarrow \{true, false\}$ is generated by $\beta : V \rightarrow L$
- $(L, \alpha, \gamma, M)$ is a Galois connection

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

$v S m \iff v R (\gamma(m))$

is generated by $\alpha \circ \beta : V \rightarrow M$
Galois Insertions

Monotone functions satisfying:

\[ \gamma \circ \alpha \sqsubseteq \lambda l.l \quad \alpha \circ \gamma \equiv \lambda m.m \]
Example (1):

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

where \(\text{sign} : \mathbb{Z} \rightarrow \text{Sign} \) is specified by:

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

Is it a Galois insertion?
Example (2):

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

where $\text{Sign} = \{-, 0, +\}$ and $\text{Parity} = \{\text{odd}, \text{even}\}$

and $\text{signparity} : \mathbb{Z} \rightarrow \text{Sign} \times \text{Parity}$:

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

Is it a Galois insertion?
Properties of Galois Insertions

**Lemma:** For a Galois connection \((L, \alpha, \gamma, M)\) the following claims are equivalent:

(i) \((L, \alpha, \gamma, M)\) is a Galois insertion;
(ii) \(\alpha\) is surjective: \(\forall m \in M : \exists l \in L : \alpha(l) = m\);
(iii) \(\gamma\) is injective: \(\forall m_1, m_2 \in M : \gamma(m_1) = \gamma(m_2) \Rightarrow m_1 = m_2\); and
(iv) \(\gamma\) 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 \(\eta\) is surjective.
Example (1) reconsidered:

\[
(P(Z), \alpha_{\text{sign}}, \gamma_{\text{sign}}, P(\text{Sign}))
\]

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

is a Galois insertion because \text{sign} is surjective.

Example (2) reconsidered:

\[
(P(Z), \alpha_{\text{signparity}}, \gamma_{\text{signparity}}, P(\text{Sign} \times \text{Parity}))
\]

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

is not a Galois insertion because \text{signparity} is not surjective.
Reduction Operators

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

Idea: remove the superfluous elements from \(M\) using a reduction operator

\[\varsigma : M \rightarrow M\]

defined from the Galois connection.

**Proposition:** Let \((L, \alpha, \gamma, M)\) be a Galois connection and define the reduction operator \(\varsigma : M \rightarrow M\) by

\[\varsigma(m) = \bigsqcap \{m' | \gamma(m) = \gamma(m')\}\]

Then \(\varsigma[M] = (\{\varsigma(m) | m \in M\}, \sqsubseteq_M)\) is a complete lattice and \((L, \alpha, \gamma, \varsigma[M])\) is a Galois insertion.
The reduction operator $\varsigma : M \rightarrow M$
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 \(\eta : V \rightarrow D\).

Then the reduction operator \(\varsigma_\eta\) 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:

\[
\begin{array}{c}
L_0 \xrightarrow{\gamma_1} L_1 \xrightarrow{\gamma_2} L_2 \xrightarrow{\gamma_3} \cdots \xrightarrow{\gamma_k} L_k \\
\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \cdots \quad \alpha_k
\end{array}
\]

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

\[(\mathcal{P}(\mathbb{Z} \times \mathbb{Z}), \alpha_{\text{diff}}, \gamma_{\text{diff}}, \mathcal{P}(\mathbb{Z}))\]

where the extraction function \( \text{diff} : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z} \) calculates the difference in magnitude:

\[
diff(z_1, z_2) = |z_1| - |z_2|
\]

The abstraction and concretisation functions are

\[
\alpha_{\text{diff}}(ZZ) = \{|z_1| - |z_2| \mid (z_1, z_2) \in ZZ\}
\]

\[
\gamma_{\text{diff}}(Z) = \{(z_1, z_2) \mid |z_1| - |z_2| \in Z\}
\]

for \( ZZ \subseteq \mathbb{Z} \times \mathbb{Z} \) and \( Z \subseteq \mathbb{Z} \).
Example: Finite Approximation

\[(\mathcal{P}(\mathbb{Z}), \alpha_{\text{range}}, \gamma_{\text{range}}, \mathcal{P}(\text{Range}))\]

where \(\text{Range} = \{-1, -1, 0, +1, >+1\}\)

and the extraction function \(\text{range} : \mathbb{Z} \rightarrow \text{Range}\) is

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

The abstraction and concretisation functions are

\[
\alpha_{\text{range}}(Z) = \{ \text{range}(z) \mid z \in Z \}
\]

\[
\gamma_{\text{range}}(R) = \{ z \mid \text{range}(z) \in R \}
\]

for \(Z \subseteq \mathbb{Z}\) and \(R \subseteq \text{Range}\).
Example: Functional Composition

\[(P(Z \times Z), \alpha_R, \gamma_R, P(\text{Range}))\]

where

\[\alpha_R = \alpha_{\text{range}} \circ \alpha_{\text{diff}}\]

\[\gamma_R = \gamma_{\text{diff}} \circ \gamma_{\text{range}}\]

The explicit formulae for the abstraction and concretisation functions

\[\alpha_R(ZZ) = \{\text{range}(|z_1| - |z_2|) \mid (z_1, z_2) \in ZZ\}\]

\[\gamma_R(R) = \{(z_1, z_2) \mid \text{range}(|z_1| - |z_2|) \in R\}\]

correspond to the extraction function \(\text{range} \circ \text{diff}\).
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

\[
\alpha(l_1, l_2) = (\alpha_1(l_1), \alpha_2(l_2))
\]

\[
\gamma(m_1, m_2) = (\gamma_1(m_1), \gamma_2(m_2))
\]
Example: Detection of Signs Analysis

Given

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

using the extraction function $\text{sign}$.

The independent attribute method gives

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

where

$$\alpha_{\text{SS}}(Z_1, Z_2) = (\{\text{sign}(z) \mid z \in Z_1\}, \{\text{sign}(z) \mid z \in Z_2\})$$

$$\gamma_{\text{SS}}(S_1, S_2) = (\{z \mid \text{sign}(z) \in S_1\}, \{z \mid \text{sign}(z) \in S_2\})$$
Motivating the Relational Method

The independent attribute method often leads to imprecision!

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

\[ \{(z, -z) \mid z \in \mathbb{Z}\} \]

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

\[(\mathbb{Z}, \mathbb{Z})\]

Hence the best property describing it will be

\[ \alpha_{SS}(\mathbb{Z}, \mathbb{Z}) = (\{-, 0, +\}, \{-, 0, +\}) \]
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

\[
\alpha(\mathcal{V}\mathcal{V}) = \bigcup \{ \alpha_1(\{v_1\}) \times \alpha_2(\{v_2\}) \mid (v_1, v_2) \in \mathcal{V}\mathcal{V} \}
\]

\[
\gamma(\mathcal{D}\mathcal{D}) = \{(v_1, v_2) \mid \alpha_1(\{v_1\}) \times \alpha_2(\{v_2\}) \subseteq \mathcal{D}\mathcal{D} \}
\]

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 \(\eta_i : V_i \rightarrow D_i\) as in

\[
\alpha_i(V'_i) = \{\eta_i(v_i) \mid v_i \in V'_i\}
\]

\[
\gamma_i(D'_i) = \{v_i \mid \eta_i(v_i) \in D'_i\}
\]

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

\[
\alpha(VV) = \{(\eta_1(v_1), \eta_2(v_2)) \mid (v_1, v_2) \in VV\}
\]

\[
\gamma(DD) = \{(v_1, v_2) \mid (\eta_1(v_1), \eta_2(v_2)) \in DD\}
\]

which also can be obtained directly from the extraction function \(\eta : V_1 \times V_2 \rightarrow D_1 \times D_2\) 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

\[(\mathcal{P}(\mathbb{Z} \times \mathbb{Z}), \alpha_{SS'}, \gamma_{SS'}, \mathcal{P}(\text{Sign} \times \text{Sign}))\]

where

\[\alpha_{SS'}(ZZ) = \{(\text{sign}(z_1), \text{sign}(z_2)) \mid (z_1, z_2) \in ZZ\}\]
\[\gamma_{SS'}(SS) = \{(z_1, z_2) \mid (\text{sign}(z_1), \text{sign}(z_2)) \in SS\}\]

corresponding to an extraction function \(\text{twosigns} : \mathbb{Z} \times \mathbb{Z} \to \text{Sign} \times \text{Sign}\) defined by

\[\text{twosigns}(z_1, z_2) = (\text{sign}(z_1), \text{sign}(z_2))\]
Advantages of the Relational Method

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

\[
\{(z, -z) \mid z \in \mathbb{Z}\}
\]

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

Analysis: The best “relational” property describing it is

\[
\alpha_{SS}(\{(z, -z) \mid z \in \mathbb{Z}\}) = \{(-, +), (0, 0), (+, -)\}
\]

whereas the best “independent attribute” property was

\[
\alpha_{SS}(\mathbb{Z}, \mathbb{Z}) = (\{-, 0, +\}, \{-, 0, +\})
\]
Function Spaces

Total Function Space

Let \((L, \alpha, \gamma, M)\) 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

\[
\alpha'(f) = \alpha \circ f \quad \quad \gamma'(g) = \gamma \circ g
\]

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

\[
\alpha(f) = \alpha_2 \circ f \circ \gamma_1 \quad \gamma(g) = \gamma_2 \circ g \circ \alpha_1
\]
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 \textit{direct product} is the Galois connection

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

defined by

\[
\alpha(l) = (\alpha_1(l), \alpha_2(l))
\]

\[
\gamma(m_1, m_2) = \gamma_1(m_1) \cap \gamma_2(m_2)
\]
Example:

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

We get the Galois connection

$$(\mathcal{P}(\mathbb{Z} \times \mathbb{Z}), \alpha_{SSR}, \gamma_{SSR}, \mathcal{P}(\text{Sign} \times \text{Sign}) \times \mathcal{P}(<\text{Range}>))$$

where

$$\alpha_{SSR}(\mathbb{Z}\mathbb{Z}) = \{(\text{sign}(z_1), \text{sign}(z_2)) \mid (z_1, z_2) \in \mathbb{Z}\mathbb{Z}\},$$

$$\gamma_{SSR}(SS, R) = \{(z_1, z_2) \mid (\text{sign}(z_1), \text{sign}(z_2)) \in SS\} \cap \{(z_1, z_2) \mid \text{range}(|z_1| - |z_2|) \in R\}$$
Motivating the Direct Tensor Product

The expression \((x, 3 \times x)\) may have a value in

\[ \{ (z, 3 \times z) \mid z \in \mathbb{Z} \} \]

which is described by

\[ \alpha_{SSR}(\{ (z, 3 \times z) \mid z \in \mathbb{Z} \}) = \{ (-,-),(0,0),(+,+),\{0,<-1}\} \]

But

- any pair described by \((0,0)\) will have a difference in magnitude described by 0
- any pair described by \((-,-)\) or \((+,+)\) 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

\[
\alpha(V') = \bigcup \{ \alpha_1(\{v\}) \times \alpha_2(\{v\}) \mid v \in V' \}
\]

\[
\gamma(DD) = \{ v \mid \alpha_1(\{v\}) \times \alpha_2(\{v\}) \subseteq DD \}
\]
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 \(\eta_i : V \rightarrow D_i\) as in

\[
\begin{align*}
\alpha_i(V') &= \{\eta_i(v) \mid v \in V'\} \\
\gamma_i(D'_i) &= \{v \mid \eta_i(v) \in D'_i\}
\end{align*}
\]

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

\[
\begin{align*}
\alpha(V') &= \{(\eta_1(v), \eta_2(v)) \mid v \in V'\} \\
\gamma(DD) &= \{v \mid (\eta_1(v), \eta_2(v)) \in DD\}
\end{align*}
\]

corresponding to the extraction function \(\eta : V \rightarrow D_1 \times D_2\) 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.

\[(P(Z \times Z), \alpha_{SSR}', \gamma_{SSR}', P(\text{Sign} \times \text{Sign} \times \text{Range}))\]

is given by

\[
\begin{align*}
\alpha_{SSR}'(ZZ) &= \{(\text{sign}(z_1), \text{sign}(z_2), \text{range}(|z_1| - |z_2|)) \mid (z_1, z_2) \in ZZ\} \\
\gamma_{SSR}'(SSR) &= \{(z_1, z_2) \mid (\text{sign}(z_1), \text{sign}(z_2), \text{range}(|z_1| - |z_2|)) \in SSR\}
\end{align*}
\]

corresponding to \texttt{twosignsrange} : \(Z \times Z \rightarrow \text{Sign} \times \text{Sign} \times \text{Range}\) given by

\[
\text{twosignsrange}(z_1, z_2) = (\text{sign}(z_1), \text{sign}(z_2), \text{range}(|z_1| - |z_2|))
\]
Advantages of the Direct Tensor Product

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

\[
\alpha_{SSR}'(\{(z, 3 \cdot z) \mid z \in \mathbb{Z}\}) = \{(-, -, <-1), (0, 0, 0), (+, +, <-1)\}
\]

compared to the direct product that gave

\[
\alpha_{SSR}(\{(z, 3 \cdot z) \mid z \in \mathbb{Z}\}) = \{(0, 0), (+), \{0, <-1\}\}
\]

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

\[
\gamma_{SSR}'(\emptyset) = \emptyset = \gamma_{SSR}'(\{(0, 0, <-1)\})
\]

so \(\gamma_{SSR}'\) 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 \emph{insertion}
\[ (L, \alpha, \gamma, \varsigma[M_1 \times M_2]) \]
defined by
\[
\alpha(l) = (\alpha_1(l), \alpha_2(l)) \\
\gamma(m_1, m_2) = \gamma_1(m_1) \sqcap \gamma_2(m_2) \\
\varsigma(m_1, m_2) = \bigcap\{(m'_1, m'_2) \mid \gamma_1(m_1) \sqcap \gamma_2(m_2) = \gamma_1(m'_1) \sqcap \gamma_2(m'_2)\}
\]
Reduced Tensor Product

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

The \textit{reduced tensor product} is the Galois \textit{insertion}

\[(P(V), \alpha, \gamma, \varsigma[P(D_1 \times D_2)])\]

defined by

\[
\begin{align*}
\alpha(V') &= \bigcup \{\alpha_1(\{v\}) \times \alpha_2(\{v\}) \mid v \in V'\} \\
\gamma(DD) &= \{v \mid \alpha_1(\{v\}) \times \alpha_2(\{v\}) \subseteq DD\} \\
\varsigma(DD) &= \bigcap \{DD' \mid \gamma(DD) = \gamma(DD')\}
\end{align*}
\]
Example: Array Bounds Analysis

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

The reduction operator \( \varsigma_{SSR'} \) amounts to

\[
\varsigma_{SSR'}(SSR) = \bigcap \{ SS'R' \mid \gamma_{SSR'}(SSR) = \gamma_{SSR'}(SS'R') \}
\]

where \( SSR, SS'R' \subseteq \text{Sign} \times \text{Sign} \times \text{Range} \).

The singleton sets constructed from the following 16 elements

\((-,0,<-1), (-,0,-1), (-,0,0), (0,-,0), (0,-,+1), (0,-,>+1), (0,0,<-1), (0,0,0), (0,0,-1), (0,0,1), (0,0,+1), (0,0,>+1), (0,+,-), (0,+,-1), (0,+,-1), (0,+,-1)\)

will be mapped to the empty set (as they are useless).
Example (cont.): Array Bounds Analysis

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

$$(-, -, <1), (-, -, 1), (-, 0, (, -, +1), (-, -, >1),$$

$$(-, 0, +1), (-, 0, >1),$$

$$(-, +, <1), (-, +, 1), (-, +, 0), (-, +, +1), (-, +, >1),$$

$$(0, -, <1), (0, -, 1), (0, 0, 0), (0, +, <1), (0, +, 1),$$

$$(+, -, <1), (+, -, 1), (+, -), (+, +, 1), (+, -, >1),$$

$$(+, 0, +1), (+, 0, >1),$$

$$(+, +, <1), (+, +, 1), (+, +, 0), (+, +, +1), (+, +, >1)$$

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

Any collection of properties can be described 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 $\mathcal{P}(\mathbb{Z})$ and $\mathcal{P}(\mathbb{Z} \times \mathbb{Z})$:

$$f_{\text{plus}}(\mathbb{Z} \times \mathbb{Z}) = \{ z_1 + z_2 | (z_1, z_2) \in \mathbb{Z} \times \mathbb{Z} \}$$

Two Galois connections

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

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

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

$$g_{\text{plus}} = \alpha_{\text{sign}} \circ f_{\text{plus}} \circ \gamma_{SS}'$$
Example (cont.):

We calculate

\[ g_{\text{plus}}(SS) = \alpha_{\text{sign}}(f_{\text{plus}}(\gamma_{SS'}(SS))) \]
\[ = \alpha_{\text{sign}}(f_{\text{plus}}(\{(z_1, z_2) \in \mathbb{Z} \times \mathbb{Z} \mid (\text{sign}(z_1), \text{sign}(z_2)) \in SS\})) \]
\[ = \alpha_{\text{sign}}(\{z_1 + z_2 \mid z_1, z_2 \in \mathbb{Z}, (\text{sign}(z_1), \text{sign}(z_2)) \in SS\}) \]
\[ = \{\text{sign}(z_1 + z_2) \mid z_1, z_2 \in \mathbb{Z}, (\text{sign}(z_1), \text{sign}(z_2)) \in SS\} \]
\[ = \bigcup\{s_1 \oplus s_2 \mid (s_1, s_2) \in SS\} \]

where \( \oplus : \text{Sign} \times \text{Sign} \rightarrow \mathcal{P}(\text{Sign}) \) is the “addition” operator on signs (so e.g. \( + \oplus + = \{+\} \) and \( + \oplus - = \{-, 0, +\} \)).
The Mundane Correctness of $f_p$ carries over to $g_p$

The correctness relation $R_i$ for $V_i$ and $L_i$:

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

Correctness of $f_p$ means

$$(p \vdash \cdot \leadsto \cdot) (R_1 \rightarrow R_2) f_p$$

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

The correctness relation $S_i$ for $V_i$ and $M_i$:

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

One can prove that

$$(p \vdash \cdot \leadsto \cdot) (R_1 \rightarrow R_2) f_p \land \alpha_2 \circ f_p \circ \gamma_1 \sqsubseteq g_p$$

$$\Rightarrow (p \vdash \cdot \leadsto \cdot) (S_1 \rightarrow S_2) g_p$$

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

PPA Section 4.5 © F.Nielson & H.Riis Nielson & C.Hankin (Dec. 2004) 89
Fixed Points in the Induced Analysis

Let \( f_p = \text{lfp}(F) \) 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 \( g_p = \text{lfp}(G) \) 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 \( \text{lfp}(F') \sqsubseteq \gamma(\text{lfp}(G)) \) and \( \alpha(\text{lfp}(F)) \sqsubseteq \text{lfp}(G) \)
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, $F \subseteq \text{Lab} \times \text{Lab}$;
- a finite set of extremal labels, $E \subseteq \text{Lab}$;
- an extremal value, $\iota \in L$; and
- a mapping $f$, from the labels $\text{Lab}$ of $F$ and $E$ to monotone transfer functions from $L$ to $L$. 
Application to Data Flow Analysis

Let \( (L, \alpha, \gamma, M) \) be a Galois connection.

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

- the mapping \( g \) from the labels \( \text{Lab} \) of \( F \) and \( E \) to monotone transfer functions of \( M \to M \) satisfies \( g_\ell \sqsupseteq \alpha \circ f_\ell \circ \gamma \) for all \( \ell \); and

- the extremal value \( \iota \) satisfies \( \gamma(\iota) = \iota \);

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:

\[ (B_\circ, B_\bullet) \models B\sqsupseteq \text{ implies } (\gamma \circ B_\circ, \gamma \circ B_\bullet) \models A\sqsupseteq \]
The Mundane Approach to Semantic Correctness

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

Correctness of every solution to $A^\exists$ amounts to:

Assume $(A_\circ, A_\bullet) \models A^\exists$ and $\langle S_\star, \sigma_1 \rangle \rightarrow^* \sigma_2$.

Then $\beta(\sigma_1) \sqsubseteq \iota$ implies $\beta(\sigma_2) \sqsubseteq \bigsqcup\{A_\bullet(\ell) \mid \ell \in \text{final}(S_\star)\}$.

where $\beta : \text{State} \rightarrow L$.

One can then prove the correctness result for $B$:

Assume $(B_\circ, B_\bullet) \models B^\exists$ and $\langle S_\star, \sigma_1 \rangle \rightarrow^* \sigma_2$.

Then $(\alpha \circ \beta)(\sigma_1) \sqsubseteq j$ implies $(\alpha \circ \beta)(\sigma_2) \sqsubseteq \bigsqcup\{B_\bullet(\ell) \mid \ell \in \text{final}(S_\star)\}$.
Sets of States Analysis

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

- the flow \(F\) is \(\text{flow}(S_\star)\);
- the set \(E\) of extremal labels is \(\{\text{init}(S_\star)\}\);
- the extremal value \(\iota\) is \(\text{State}\); and
- the transfer functions are given by \(f^{\text{SS}}\):

\[
[x := a]^\ell \quad f^{\text{SS}}_\ell(\Sigma) = \{\sigma[x \mapsto A[a]]\sigma \mid \sigma \in \Sigma\}
\]
\[
[\text{skip}]^\ell \quad f^{\text{SS}}_\ell(\Sigma) = \Sigma
\]
\[
[b]^\ell \quad f^{\text{SS}}_\ell(\Sigma) = \Sigma
\]

where \(\Sigma \subseteq \text{State}\).

Correctness: Assume \((\text{SS}_0, \text{SS}_\bullet) \models \text{SS}_\trianglerighteq\) and \(\langle S_\star, \sigma_1 \rangle \rightarrow^* \sigma_2\). Then \(\sigma_1 \in \text{State}\) implies \(\sigma_2 \in \bigcup\{\text{SS}_\bullet(\ell) \mid \ell \in \text{final}(S_\star)\}\).
Constant Propagation Analysis

Generalised Monotone Framework over $\text{State}_{\text{CP}} = ((\text{Var} \rightarrow \mathbb{Z}^\top)_{\bot}, \sqsubseteq)$. Instance $\text{CP}$ for $S_*$:

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

\[
\begin{align*}
[x := a]^{\ell} : f^\text{CP}_{\ell}(\hat{\sigma}) &= \begin{cases} \bot & \text{if } \hat{\sigma} = \bot \\ \hat{\sigma}[x \mapsto A^\text{CP}[\llbracket a \rrbracket]\hat{\sigma}] & \text{otherwise} \end{cases} \\
[\text{skip}]^{\ell} : f^\text{CP}_{\ell}(\hat{\sigma}) &= \hat{\sigma} \\
[b]^{\ell} : f^\text{CP}_{\ell}(\hat{\sigma}) &= \hat{\sigma}
\end{align*}
\]
Galois Connection

The representation function $\beta_{CP} : \text{State} \rightarrow \widehat{\text{State}}_{CP}$ is defined by

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

This gives rise to a Galois connection

$$\left( \mathcal{P}(\text{State}), \alpha_{CP}, \gamma_{CP}, \widehat{\text{State}}_{CP} \right)$$

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

One can show that for all labels $\ell$

$$f_{\ell}^{CP} \sqsubseteq \alpha_{CP} \circ f_{\ell}^{SS} \circ \gamma_{CP}$$

as well as

$$\gamma_{CP}(\lambda x. T) = \text{State}$$

It follows that $\text{CP}$ is an upper approximation to the analysis induced from $\text{SS}$ 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 \((L, \alpha, \gamma, M)\).

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) \((L, \alpha, \gamma, M)\) is a Galois insertion and \( \nabla_M : M \times M \to M \) is a widening.
Precision of the Induced Widening Operator

**Lemma:** Let \((L, \alpha, \gamma, M)\) 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

\[
\text{lfp}_{\nabla_L}(f) = \gamma(\text{lfp}_{\nabla_M}(\alpha \circ f \circ \gamma))
\]

for all monotone functions \(f : L \to L\).
Precision of the Induced Widening Operator

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

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

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