Program Analysis
Lecture 9

Rayna Dimitrova

WS 2016/2017
We interpret a function $f : \mathbb{Z}^n \to \mathbb{Z}$ on sets of integers by defining the function $f_P : \mathcal{P}(\mathbb{Z})^n \to \mathcal{P}(\mathbb{Z})$ such that for $Z_1, \ldots, Z_n \subseteq \mathbb{Z}$:

$$f_P(Z_1, \ldots, Z_n) := \{ f(z_1, \ldots, z_n) \mid z_1 \in Z_1, \ldots, z_n \in Z_n \}$$

We interpret a relation $r : \mathbb{Z}^n \to \mathbb{B}$ on sets of integers by defining the relation $r_P : \mathcal{P}(\mathbb{Z})^n \to \mathcal{P}(\mathbb{B})$ such that for $Z_1, \ldots, Z_n \subseteq \mathbb{Z}$:

$$r_P(Z_1, \ldots, Z_n) := \{ r(z_1, \ldots, z_n) \mid z_1 \in Z_1, \ldots, z_n \in Z_n \}$$
Approximating functions and relations

Let $\mathcal{P}(\mathbb{Z}) \xleftarrow{\gamma_\eta} \mathcal{P}(A)$ be the Galois connection for $(\mathcal{P}(\mathbb{Z}), \subseteq)$ and $(\mathcal{P}(A), \subseteq)$ defined by a given extraction function $\eta : \mathbb{Z} \to A$.

Suppose that we are given

- for every $op \in \{+, -, *, /\}$, a function $\mathcal{I}^\#(op) : \mathcal{P}(A) \times \mathcal{P}(A) \to \mathcal{P}(A)$ that is an upper approximation to the function $\mathcal{I}(op)_\mathcal{P} : \mathcal{P}(\mathbb{Z}) \times \mathcal{P}(\mathbb{Z}) \to \mathcal{P}(\mathbb{Z})$, that is, for all $A_1, A_2 \subseteq A$

  $$\alpha_\eta(\mathcal{I}(op)_\mathcal{P}(\gamma_\eta(A_1), \gamma_\eta(A_2))) \subseteq \mathcal{I}^\#(op)(A_1, A_2)$$

- for every $rel \in \{<, =\}$, a relation $\mathcal{I}^\#(rel) : \mathcal{P}(A) \times \mathcal{P}(A) \to \mathcal{P}(\mathbb{B})$ that is an upper approximation to the relation $\mathcal{I}(rel)_\mathcal{P} : \mathcal{P}(\mathbb{Z}) \times \mathcal{P}(\mathbb{Z}) \to \mathcal{P}(\mathbb{B})$, that is, for all $A_1, A_2 \subseteq A$

  $$\mathcal{I}(rel)_\mathcal{P}(\gamma_\eta(A_1), \gamma_\eta(A_2)) \subseteq \mathcal{I}^\#(rel)(A_1, A_2)$$
Approximating functions and relations

Let $\mathcal{P}(\mathbb{Z}) \xleftrightarrow{\gamma_\eta} \mathcal{P}(A)$ be the Galois connection for $(\mathcal{P}(\mathbb{Z}), \subseteq)$ and $(\mathcal{P}(A), \subseteq)$ defined by a given extraction function $\eta : \mathbb{Z} \to A$.

If $f_\mathcal{P} : \mathcal{P}(\mathbb{Z})^n \to \mathcal{P}(\mathbb{Z})$ and $r_\mathcal{P} : \mathcal{P}(\mathbb{Z})^n \to \mathcal{P}(\mathbb{B})$, we can define $f^\# : \mathcal{P}(A)^n \to \mathcal{P}(A)$ and $r^\# : \mathcal{P}(A)^n \to \mathcal{P}(\mathbb{B})$ using the abstraction and concretization functions, where for $A_1, \ldots, A_n \subseteq A$ we let

$$f^\#(A_1, \ldots, A_n) = \alpha_\eta(f_\mathcal{P}(\gamma_\eta(A_1), \ldots, \gamma_\eta(A_n)))$$
$$r^\#(A_1, \ldots, A_n) = r_\mathcal{P}(\gamma_\eta(A_1), \ldots, \gamma_\eta(A_n))$$

It is easy to see that

$f^\#$ is an upper approximation to $f_\mathcal{P}$

$r^\#$ is an upper approximation to $r_\mathcal{P}$
Example: upper approximation of arithmetic operations

Consider $\mathcal{P}(\mathbb{Z}) \xleftrightarrow{\gamma_\eta \alpha_\eta} \mathcal{P}(\{+, -, 0\})$ where $\eta = \text{sign}$

<table>
<thead>
<tr>
<th></th>
<th>$+$</th>
<th>$-$</th>
<th>$0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${+}$</td>
<td>${+}$</td>
<td>${+}$</td>
<td>${+}$</td>
</tr>
<tr>
<td>${-}$</td>
<td>${+, -, 0}$</td>
<td>${-}$</td>
<td>${-}$</td>
</tr>
<tr>
<td>${0}$</td>
<td>${+}$</td>
<td>${-}$</td>
<td>${0}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$+$</th>
<th>$-$</th>
<th>$0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${+}$</td>
<td>${+}$</td>
<td>${-}$</td>
<td>${0}$</td>
</tr>
<tr>
<td>${-}$</td>
<td>${-}$</td>
<td>${+}$</td>
<td>${0}$</td>
</tr>
<tr>
<td>${0}$</td>
<td>${0}$</td>
<td>${0}$</td>
<td>${0}$</td>
</tr>
</tbody>
</table>

$\{+, 0\} *\# \{-\} = \{+\} *\# \{-\} \cup \{0\} *\# \{-\} = \{-\} \cup \{0\} = \{-, 0\}$
Example: upper approximation of relations

Consider $\mathcal{P}(\mathbb{Z}) \xymatrix{\nearrow \gamma_{\eta} \ar@{<=>}[<,>][rr]_{\alpha_{\eta}} & \mathcal{P}(\{+,-,0\})}$ where $\eta = \text{sign}$

<table>
<thead>
<tr>
<th>$=$</th>
<th>${+}$</th>
<th>${-}$</th>
<th>${0}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${+}$</td>
<td>${\text{true, false}}$</td>
<td>${\text{false}}$</td>
<td>${\text{false}}$</td>
</tr>
<tr>
<td>${-}$</td>
<td>${\text{false}}$</td>
<td>${\text{true, false}}$</td>
<td>${\text{false}}$</td>
</tr>
<tr>
<td>${0}$</td>
<td>${\text{false}}$</td>
<td>${\text{false}}$</td>
<td>${\text{true}}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$&gt;$</th>
<th>${+}$</th>
<th>${-}$</th>
<th>${0}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${+}$</td>
<td>${\text{true, false}}$</td>
<td>${\text{true}}$</td>
<td>${\text{true}}$</td>
</tr>
<tr>
<td>${-}$</td>
<td>${\text{false}}$</td>
<td>${\text{true, false}}$</td>
<td>${\text{false}}$</td>
</tr>
<tr>
<td>${0}$</td>
<td>${\text{false}}$</td>
<td>${\text{true}}$</td>
<td>${\text{false}}$</td>
</tr>
</tbody>
</table>

$\{+, 0\} =$ $\# \{0\} = \{+\} =$ $\# \{0\} \cup \{0\} =$ $\# \{0\}$$\Rightarrow \{false\} \cup \{true\}$$\Rightarrow \{true, false\}$
3-valued approximation of Boolean operators

We approximate Boolean operators in the 3-valued logic

$$(\mathcal{P}(B) \setminus \{\emptyset\}, \neg_3, \lor_3)$$

where we interpret elements of $\mathcal{P}(B) \setminus \{\emptyset\}$ numerically as follows

- $\{\text{false}\} = 0$ (definitely false)
- $\{\text{true}\} = 1$ (definitely true)
- $\{\text{true, false}\} = 1/2$ (unknown value)

The approximations of $\neg$ and $\lor$ are defined by

$$
\begin{array}{c|c|c|c}
\lor_3 & 0 & 1 & 1/2 \\
\hline
0 & 0 & 1 & 1/2 \\
1 & 1 & 1 & 1 \\
1/2 & 1/2 & 1 & 1/2 \\
\end{array}
$$

$$
\begin{array}{c|c}
\neg_3 & 0 & 1 \\
\hline
0 & 1 \\
1 & 0 \\
1/2 & 1/2 \\
\end{array}
$$
Derivation of an abstract semantics

Let $\mathcal{P}(\text{States}) \xrightleftharpoons[\alpha]{\gamma} \mathcal{P}((\text{Vars} \to A))$ be the Galois connection defined by lifting a given extraction function $\eta : \mathbb{Z} \to A$. Then,

$$\alpha(S) = \{ \eta \circ \sigma \mid \sigma \in S \} \text{ for } S \subseteq \text{States}$$

We define the set of abstract states $A\text{States} := (\text{Vars} \to A)$.

Suppose that we are given functions $\mathcal{I}^\#(op)$ for every $op \in \{+, -, \ast, /\}$ and relations $\mathcal{I}^\#(rel)$ for every $rel \in \{<, =\}$ which are upper approximations as required earlier.

We will define an abstract transition relation

$$\Rightarrow \subseteq (\text{Progs} \times \mathcal{P}(A\text{States})) \times ((\text{Progs} \times \mathcal{P}(A\text{States})) \cup \mathcal{P}(A\text{States}))$$

and abstract predicate transformers

$$\text{post}^\# : \mathcal{P}(A\text{States}) \to \mathcal{P}(A\text{States}) \text{ and}$$

$$\text{post}^\#_{c,c'} : \mathcal{P}(A\text{States}) \to \mathcal{P}(A\text{States})$$
Abstract semantics of expressions

\[ \mathcal{A}^\# : \text{AExp} \rightarrow (\text{AStates} \rightarrow \mathcal{P}(A)) \]

\[ \mathcal{A}^\#[x] \rho = \{ \rho(x) \} \]

\[ \mathcal{A}^\#[k] \rho = \{ \eta(N[k]) \} \]

\[ \mathcal{A}^\#[a_1 \text{ op } a_2] \rho = \mathcal{I}^\#(\text{op})(\mathcal{A}^\#[a_1] \rho, \mathcal{A}^\#[a_2] \rho) \]

\[ \mathcal{B}^\# : \text{BExp} \rightarrow (\text{AStates} \rightarrow \mathcal{P}(\mathbb{B})) \]

\[ \mathcal{B}^\#[\text{true}] \rho = \{ \text{true} \} \]

\[ \mathcal{B}^\#[\neg b] \rho = \neg_3 \mathcal{B}^\#[b] \rho \]

\[ \mathcal{B}^\#[b_1 \lor b_2] \rho = \mathcal{B}^\#[b_1] \rho \lor_3 \mathcal{B}^\#[b_2] \rho \]

\[ \mathcal{B}^\#[a_1 \text{ rel } a_2] \rho = \mathcal{I}^\#(\text{rel})(\mathcal{A}^\#[a_1] \rho, \mathcal{A}^\#[a_2] \rho) \]
Example: abstract semantics of expressions

Consider $\mathcal{P}(\mathbb{Z}) \xrightarrow{\gamma_\eta} \mathcal{P}\{+, -, 0\}$ where $\eta = \text{sign}$ and the abstract state $\rho$ where $\rho(x) = +$ and $\rho(y) = -$. Then

$A^\#[2 \times x + y] \rho = \{+, -, 0\}$

$B^\#[\neg(x + 1 > y)] \rho = \{false\}$
Abstract transition relation

We now derive the abstract transition relation

$$\Rightarrow \subseteq (Progs \times \mathcal{P}(AStates)) \times ((Progs \times \mathcal{P}(AStates)) \cup \mathcal{P}(AStates))$$

by a set of rules that are abstract counterparts of the rules for $\rightarrow$. 
Abstract transition relation

**SKIP**

\[(\text{[skip]}^l, \text{abs}) \Rightarrow \text{abs}\]

**ASSIGN**

\[(\text{[x := a]}^l, \text{abs}) \Rightarrow \{\rho[x \mapsto d] \mid \rho \in \text{abs}, d \in A\#[[a] \rho}\}\]

**READ**

\[(\text{[read(x)}]^l, \text{abs}) \Rightarrow \{\rho[x \mapsto \eta(z)] \mid \rho \in \text{abs}, z \in \mathbb{Z}\}\]

**PRINT**

\[(\text{[print(x)}]^l, \text{abs}) \Rightarrow \text{abs}\]

**SEQ1**

\[(c_1, \text{abs}) \Rightarrow \text{abs}'\]

\[(c_1; c_2, \text{abs}) \Rightarrow (c_2, \text{abs}')\]

**SEQ2**

\[(c_1, \text{abs}) \Rightarrow (c_1', \text{abs}')\]

\[(c_1; c_2, \text{abs}) \Rightarrow (c_1'; c_2, \text{abs}')\]
Abstract transition relation

**IF**<sub>true</sub>

\[
\exists \rho \in \text{abs} : \text{true} \in \mathcal{B}^\# [b] \rho \\
(\text{if } [b]^l \text{ then } \{c_1\} \text{ else } \{c_2\}, \text{abs}) \Rightarrow \\
(c_1, \text{abs} \setminus \{\rho \in \text{abs} \mid \mathcal{B}^\# [b] \rho = \{\text{false}\}\})
\]

**IF**<sub>false</sub>

\[
\exists \rho \in \text{abs} : \text{false} \in \mathcal{B}^\# [b] \rho \\
(\text{if } [b]^l \text{ then } \{c_1\} \text{ else } \{c_2\}, \text{abs}) \Rightarrow \\
(c_2, \text{abs} \setminus \{\rho \in \text{abs} \mid \mathcal{B}^\# [b] \rho = \{\text{true}\}\})
\]

**WHILE**<sub>true</sub>

\[
\exists \rho \in \text{abs} : \text{true} \in \mathcal{B}^\# [b] \rho \\
(\text{while } [b]^l \text{ do } \{c\}, \text{abs}) \Rightarrow \\
(c; \text{while } [b]^l \text{ do } \{c\}, \text{abs} \setminus \{\rho \in \text{abs} \mid \mathcal{B}^\# [b] \rho = \{\text{false}\}\})
\]

**WHILE**<sub>false</sub>

\[
\exists \rho \in \text{abs} : \text{false} \in \mathcal{B}^\# [b] \rho \\
(\text{while } [b]^l \text{ do } \{c\}, \text{abs}) \Rightarrow \\
\text{abs} \setminus \{\rho \in \text{abs} \mid \mathcal{B}^\# [b] \rho = \{\text{true}\}\}
\]
Abstract predicate transformers

For all $c, c' \in Progs$ we define $\text{post}_c^\# : \mathcal{P}(AStates) \rightarrow \mathcal{P}(AStates)$ and $\text{post}_{c,c'}^\# : \mathcal{P}(AStates) \rightarrow \mathcal{P}(AStates)$ by

$$\text{post}_c^\#(abs) := \begin{cases} abs' & \text{if } (c, abs) \Rightarrow abs' \\ \emptyset & \text{otherwise} \end{cases}$$

$$\text{post}_{c,c'}^\#(abs) := \begin{cases} abs' & \text{if } (c, abs) \Rightarrow (c', abs') \\ \emptyset & \text{otherwise} \end{cases}$$

Theorem The family of functions $\text{post}_c^\#$ and $\text{post}_{c,c'}^\#$ is an abstract semantics, that is, for all $c, c' \in Progs$ we have that

$$\alpha \circ \text{post}_c \circ \gamma \subseteq \text{post}_c^\#$$
$$\alpha \circ \text{post}_{c,c'} \circ \gamma \subseteq \text{post}_{c,c'}^\#$$
Soundness of abstract semantics

**Theorem** The family of functions $\text{post}_c^\#$ and $\text{post}_{c,c'}^\#$ is an abstract semantics, that is, for all $c, c' \in \text{Progs}$ we have that

$$
\alpha \circ \text{post}_c \circ \gamma \subseteq \text{post}_c^\#
$$

$$
\alpha \circ \text{post}_{c,c'} \circ \gamma \subseteq \text{post}_{c,c'}^\#
$$

**Lemma** (Soundness of abstract evaluation) Let $\eta : \mathbb{Z} \rightarrow A$ be the extraction function used to define the abstraction. Then:

(1) For every $a \in A\text{Exp}$ and $\sigma \in \text{States}$ we have

$$
\eta(\mathcal{A}[a]\sigma) \in \mathcal{A}^\#[a](\eta \circ \sigma)
$$

(2) For every $b \in B\text{Exp}$ and $\sigma \in \text{States}$ we have

$$
\mathcal{B}[b]\sigma \in \mathcal{B}^\#[b](\eta \circ \sigma)
$$
Soundness of abstract semantics

\[ \alpha(S) = \{ \eta \circ \sigma \mid \sigma \in S \} \quad \text{for } S \subseteq \text{States} \]

\[ \gamma(\text{\text{abs}}) = \{ \sigma \mid \eta \circ \sigma \in \text{\text{abs}} \} \quad \text{for } \text{\text{abs}} \subseteq \text{AStates} \]

\[ \text{post}_c(P) = \{ \sigma' \mid \exists \sigma \in P : (c, \sigma) \rightarrow \sigma' \} \quad \text{for } P \subseteq \text{States} \]

\[ \text{post}_{c,c'}(P) = \{ \sigma' \mid \exists \sigma \in P : (c, \sigma) \rightarrow (c', \sigma') \} \quad \text{for } P \subseteq \text{States} \]
Soundness of abstract semantics

\[ \alpha(S) = \{ \eta \circ \sigma \mid \sigma \in S \} \quad \text{for } S \subseteq \text{States} \]
\[ \gamma(\text{abs}) = \{ \sigma \mid \eta \circ \sigma \in \text{abs} \} \quad \text{for } \text{abs} \subseteq A\text{States} \]

\[ \text{post}_c(P) = \{ \sigma' \mid \exists \sigma \in P : (c, \sigma) \rightarrow \sigma' \} \quad \text{for } P \subseteq \text{States} \]
\[ \text{post}_{c,c'}(P) = \{ \sigma' \mid \exists \sigma \in P : (c, \sigma) \rightarrow (c', \sigma') \} \quad \text{for } P \subseteq \text{States} \]

\[ \text{post}_c(\gamma(\text{abs})) = \{ \sigma' \mid \exists \sigma : \eta \circ \sigma \in \text{abs}, (c, \sigma) \rightarrow \sigma' \} \]
\[ \text{post}_{c,c'}(\gamma(\text{abs})) = \{ \sigma' \mid \exists \sigma : \eta \circ \sigma \in \text{abs}, (c, \sigma) \rightarrow (c', \sigma') \} \]
Soundness of abstract semantics

\[ \alpha(S) = \{ \eta \circ \sigma \mid \sigma \in S \} \quad \text{for } S \subseteq \text{States} \]
\[ \gamma(\text{abs}) = \{ \sigma \mid \eta \circ \sigma \in \text{abs} \} \quad \text{for } \text{abs} \subseteq A\text{States} \]

\[ \text{post}_c(P) = \{ \sigma' \mid \exists \sigma \in P : (c, \sigma) \rightarrow \sigma' \} \quad \text{for } P \subseteq \text{States} \]
\[ \text{post}_{c,c'}(P) = \{ \sigma' \mid \exists \sigma \in P : (c, \sigma) \rightarrow (c', \sigma') \} \quad \text{for } P \subseteq \text{States} \]

\[ \text{post}_c(\gamma(\text{abs})) = \{ \sigma' \mid \exists \sigma : \eta \circ \sigma \in \text{abs}, (c, \sigma) \rightarrow \sigma' \} \]
\[ \text{post}_{c,c'}(\gamma(\text{abs})) = \{ \sigma' \mid \exists \sigma : \eta \circ \sigma \in \text{abs}, (c, \sigma) \rightarrow (c', \sigma') \} \]

\[ \alpha(\text{post}_c(\gamma(\text{abs}))) = \{ \eta \circ \sigma' \mid \exists \sigma : \eta \circ \sigma \in \text{abs}, (c, \sigma) \rightarrow \sigma' \} \]
\[ \alpha(\text{post}_{c,c'}(\gamma(\text{abs}))) = \{ \eta \circ \sigma' \mid \exists \sigma : \eta \circ \sigma \in \text{abs}, (c, \sigma) \rightarrow (c', \sigma') \} \]
Soundness of abstract semantics

\[
\alpha(\text{post}_c(\gamma(\text{abs}))) = \{ \eta \circ \sigma' \mid \exists \sigma : \eta \circ \sigma \in \text{abs}, (c, \sigma) \rightarrow \sigma' \}
\]

\[
\alpha(\text{post}_{c,c'}(\gamma(\text{abs}))) = \{ \eta \circ \sigma' \mid \exists \sigma : \eta \circ \sigma \in \text{abs}, (c, \sigma) \rightarrow (c', \sigma') \}
\]

\[
\text{post}_c^\#(\text{abs}) := \begin{cases} 
\text{abs}' & \text{if } (c, \text{abs}) \Rightarrow \text{abs}' \\
\emptyset & \text{otherwise}
\end{cases}
\]

\[
\text{post}_{c,c'}^\#(\text{abs}) := \begin{cases} 
\text{abs}' & \text{if } (c, \text{abs}) \Rightarrow (c', \text{abs}') \\
\emptyset & \text{otherwise}
\end{cases}
\]

We show that

\[
\alpha(\text{post}_c(\gamma(\text{abs}))) \subseteq \text{post}_c^\#(\text{abs})
\]

\[
\alpha(\text{post}_{c,c'}(\gamma(\text{abs}))) \subseteq \text{post}_{c,c'}^\#(\text{abs})
\]

by induction on the inference in the semantics.
Soundness of abstract semantics

Proof of the theorem We give the proof only for the cases ASSIGN, SEQ₁ and SEQ₂. The proofs for the other cases are similar.

ASSIGN

Suppose that \( \eta \circ \sigma' \) is such that there exists \( \sigma \) where \( \eta \circ \sigma \in \text{abs} \) and \( ([x := a]^l, \sigma) \rightarrow \sigma' \). Then, \( \sigma' = \sigma[x \rightarrow A[a] \sigma] \).

By the Lemma we have \( \eta(A[a] \sigma) \in A^\# [a](\eta \circ \sigma) \). As also \( \eta \circ \sigma \in \text{abs} \), the definition of the transition relation \( \Rightarrow \) implies that

\[
(\eta \circ \sigma)[x \mapsto \eta(A[a] \sigma)] \in \text{post}^\#_c(\text{abs}).
\]

This is the same as

\[
\eta \circ (\sigma[x \mapsto A[a] \sigma]) \in \text{post}^\#_c(\text{abs}),
\]

which is

\[
\eta \circ \sigma' \in \text{post}^\#_c(\text{abs}).
\]
SEQ1

Suppose that \( \eta \circ \sigma' \) is such that there exists \( \sigma \) where \( \eta \circ \sigma \in \text{abs} \) and \( (c_1; c_2, \sigma) \rightarrow (c_2, \sigma') \) because \( (c_1, \sigma) \rightarrow \sigma' \).

Since \( \eta \circ \sigma \in \text{abs} \), we have \( \sigma \in \gamma(\text{abs}) \). As also \( (c_1, \sigma) \rightarrow \sigma' \), by the definition of post, we have

\[
\sigma' \in \text{post}_{c_1}(\gamma(\text{abs})).
\]

Applying the definition of \( \alpha \) we get

\[
\eta \circ \sigma' \in \alpha(\text{post}_{c_1}(\gamma(\text{abs}))).
\]

From this, and the induction hypothesis we get

\[
\eta \circ \sigma' \in \text{post}^\#_{c_1}(\text{abs}).
\]

By the rule \( \text{SEQ1} \) for the relation \( \Rightarrow \) we have

\[
(c_1; c_2, \text{abs}) \Rightarrow (c_2, \text{post}^\#_{c_1}(\text{abs})),
\]

which together with the above implies that \( \eta \circ \sigma' \in \text{post}^\#_{c_1; c_2, \text{abs}}(\text{abs}) \).
Proof continued

SEQ\textsubscript{2}

Suppose that $\eta \circ \sigma'$ is such that there exists $\sigma$ where $\eta \circ \sigma \in \text{abs}$ and $(c_1; c_2, \sigma) \to (c'_1; c_2, \sigma')$ because $(c_1, \sigma) \to (c'_1, \sigma')$.

Since $\eta \circ \sigma \in \text{abs}$, we have $\sigma \in \gamma(\text{abs})$. As also $(c_1, \sigma) \to (c'_1, \sigma')$, by the definition of post, we have

$$\sigma' \in \text{post}_{c_1, c'_1}(\gamma(\text{abs})).$$

Applying the definition of $\alpha$ we get

$$\eta \circ \sigma' \in \alpha(\text{post}_{c_1, c'_1}(\gamma(\text{abs}))).$$

From this, and the induction hypothesis we get

$$\eta \circ \sigma' \in \text{post}_{c_1, c'_1}^\#(\text{abs}).$$

By the rule SEQ\textsubscript{2} for the relation $\Rightarrow$ we have

$$(c_1; c_2, \text{abs}) \Rightarrow (c'_1; c_2, \text{post}_{c_1, c'_1}^\#(\text{abs})),$$

which together with the above implies that $\eta \circ \sigma' \in \text{post}_{c_1; c_2, c'_1; c_2}^\#(\text{abs})$. 
Two applications of Galois connections

- Application to dataflow analysis
- Widening operator induced by a Galois connection
Application to dataflow analysis

**Sets of States** analysis and **Constant Propagation** analysis

- Constant Propagation analysis is an upper approximation of Sets of States analysis (established via a Galois connection)

- Thus, correctness of Sets of States analysis implies correctness of Constant Propagation analysis
Sets of states analysis (lattice)

Approximates how sets of states are transformed into sets of states.

Forward may analysis with

- complete lattice \((L, \leq_L) := (\mathcal{P}(States), \subseteq)\)

- extremal value \(i^S := States\)
Sets of states analysis (transfer functions)

» for \([x := a]^b\) let

\[
f_b^S(P) := \{ \sigma[x \mapsto \mathcal{A}[a]\sigma] \mid \sigma \in P \}\]

» for \([\text{read}(x)]^b\) let

\[
f_b^S(P) := \{ \sigma[x \mapsto z] \mid \sigma \in P, z \in \mathbb{Z} \}\]

» for \([\text{print}(a)]^b\), for \([\text{skip}]^b\), and for condition \([e]^b\) let

\[
f_b^S(P) = P \text{ for all } P \subseteq \text{States}\]
Constant propagation analysis (lattice)

\[(M, \leq_M) := ((\text{Vars} \rightarrow (\mathbb{Z} \cup \{?\})) \cup \{\perp\}, \leq_M)\]

where the partial ordering \(\leq_M\) is defined by

- \(\perp \leq_M m\) for all \(m \in M\),
- \(m_1 \leq_M m_2\) if and only if for every \(v \in \text{Vars}\) we have that either \(m_1(v), m_2(v) \in \mathbb{Z}\) and \(m_1(v) = m_2(v)\), or \(m_2(v) = ?\)

extremal value \(i^{CP} := \top_M\)
Constant propagation analysis (transfer functions)

- for \([x := a]^b\) let

\[
f_{CP}^b(m) := \begin{cases} 
\bot & \text{if } m = \bot, \\
 m[x \mapsto \text{eval}(a, m)] & \text{otherwise}
\end{cases}
\]

- for \([\text{read}(x)]^b\) let

\[
f_{CP}^b(m) := \begin{cases} 
\bot & \text{if } m = \bot, \\
 m[x \mapsto ?] & \text{otherwise}
\end{cases}
\]

- for \([\text{print}(a)]^b\), for \([\text{skip}]^b\), and for condition \([e]^b\) let

\[
f_{CP}^b(m) := m \text{ for all } m \in M
\]
Galois connection relating the two analyses

We define a function $\beta : States \rightarrow M$ as $\beta(\sigma) := \sigma$.

The function $\beta$ defines a Galois connection $\mathcal{P}(States) \xleftarrow{\gamma} M$ where $\alpha$ and $\gamma$ are given by:

$$\alpha(P) := \bigsqcup \{ \beta(\sigma) \mid \sigma \in P \} \text{ for } P \in \mathcal{P}(States)$$

$$\gamma(m) := \{ \sigma \mid \beta(\sigma) \leq_M m \} \text{ for } m \in M$$

We can show that

- $\gamma(\top_M) = States$

- For all blocks $b$ it holds that $\alpha \circ f_b^S \circ \gamma \leq_M f_{b^{CP}}$.

This implies that Constant Propagation analysis is an upper approximation of the Sets of States analysis.
Galois connection relating the two analyses

We will show $\alpha \circ f_b^S \circ \gamma \leq_M f_b^{CP}$ only for the case $[x := a]^b$.

For $m = \bot$ the claim easily follows. Let $m \neq \bot$.

$$
\alpha(f_b^S(\gamma(m))) = \alpha(f_b^S(\{\sigma \mid \beta(\sigma) \leq_M m\})) = \alpha(f_b^S(\{\sigma \mid \sigma \leq_M m\})) = \alpha(\{\sigma[x \mapsto A[a]a] \mid \sigma \leq_M m\}) = \bigsqcup\{\beta(\sigma[x \mapsto A[a]a]) \mid \sigma \leq_M m\} = \bigsqcup\{\sigma[x \mapsto A[a]a] \mid \sigma \leq_M m\} \leq_M m[x \mapsto \bigsqcup\{A[a]a \sigma \mid \sigma \leq_M m\}]
$$

By induction on the structure of arithmetic expressions it is easy to show that $\bigsqcup\{A[a]a \sigma \mid \sigma \leq_M m\} \leq eval(a, m)$ (where $\bigsqcup$ and $\leq$ are from the lattice $(\mathbb{Z} \cup \{\bot, ?\}, \leq)$ in which the elements of $\mathbb{Z}$ are incomparable to each other and $?$ is the greatest element). This gives us the following and completes the proof for this case.

$$
m[x \mapsto \bigsqcup\{A[a]a \sigma \mid \sigma \leq_M m\}] \leq_M m[x \mapsto eval(a, m)] = f_b^{CP}(m)
$$
Widening operator induced by Galois connection

Suppose we are given

- complete lattices \((L, \leq_L)\) and \((M, \leq_M)\)
- monotonic functions \(\alpha : L \to M\) and \(\gamma : M \to L\) such that
\[ L \xrightarrow{\gamma} M \xleftarrow{\alpha} L \]
is a Galois connection
- widening operator \(\nabla_M : M \times M \to M\)

Consider a monotonic function \(f : L \to L\). The ascending chain \((f^n(\bot))_n\) may not stabilize, or may do so in too many iterations.

**Goal:** approximate \(f\)

**Approaches:**

- use the function \(\alpha \circ f \circ \gamma : M \to M\) to approximate \(f\)
- use \(\nabla_L : L \times L \to L\) defined by
\[ l_1 \nabla_L l_2 = \gamma(\alpha(l_1) \nabla_M \alpha(l_2))\]
Widening operator induced by Galois connection

Suppose we are given

- complete lattices \((L, \leq_L)\) and \((M, \leq_M)\)
- monotonic functions \(\alpha : L \rightarrow M\) and \(\gamma : M \rightarrow L\) such that \(\xymatrix{L \ar@<-1ex>[r]_{\alpha} & M \ar@<1ex>[l]^{\gamma}}\) is a Galois connection
- widening operator \(\nabla_M : M \times M \rightarrow M\)

Consider a monotonic function \(f : L \rightarrow L\). The ascending chain \((f^n(\bot))_n\) may not stabilize, or may do so in too many iterations.

If \(\nabla_L : L \times L \rightarrow L\) defined by \(l_1 \nabla_L l_2 = \gamma(\alpha(l_1) \nabla_M \alpha(l_2))\) is a widening operator, we can approximate \(\text{lfp}(f)\) while doing the computation over \(L\), without reducing the precision of all operations.
Widening operator induced by Galois connection

Lemma Suppose that we are given

- complete lattices \((L, \leq_L)\) and \((M, \leq_M)\)
- monotonic functions \(\alpha : L \rightarrow M\) and \(\gamma : M \rightarrow L\) such that \(L \xleftarrow{\gamma} M \xrightarrow{\alpha} M\) is a Galois connection
- widening operator \(\nabla_M : M \times M \rightarrow M\)

The function \(\nabla_L : L \times L \rightarrow L\) is defined by the formula

\[
l_1 \nabla_L l_2 = \gamma(\alpha(l_1) \nabla_M \alpha(l_2)).
\]

Then it holds that

- If \((M, \leq_M)\) satisfies the Ascending Chain Condition, then \(\nabla_L\) is a widening operator.
- If \(L \xleftarrow{\gamma} M\) is a Galois insertion, then \(\nabla_L\) is a widening operator.
Proof of the second statement

We have to show that:

(1) For every $l_1, l_2 \in L$ it holds that $l_1 \sqcup l_2 \leq_L l_1 \nabla_L l_2$.

(2) For every ascending sequence $l_0 \leq_L l_1 \leq_L l_2 \leq_L \ldots$ the sequence $l_0^{\nabla_L} \leq_L l_1^{\nabla_L} \leq_L l_2^{\nabla_L} \leq_L \ldots$, where $l_0^{\nabla_L} = l_0$ and $l_{i+1}^{\nabla_L} = l_i^{\nabla_L} \nabla_L l_{i+1}$ for all $i$, eventually stabilizes.

(1) Since $\nabla_M$ is a widening operator, we have

$$\alpha(l_i) \leq_M \alpha(l_1) \nabla_M \alpha(l_2) \text{ for } i \in \{1, 2\}$$

Since we have a Galois connection, this implies

$$l_i \leq_L \gamma(\alpha(l_1) \nabla_M \alpha(l_2)) \text{ for } i \in \{1, 2\}$$

By the definition of $\nabla_L$ we have

$$l_i \leq_L l_1 \nabla_L l_2 \text{ for } i \in \{1, 2\}$$

Since $l_1 \sqcup l_2$ is the least upper bound, $l_1 \sqcup l_2 \leq_L l_1 \nabla_L l_2$. 
Proof of the second statement (continued)

(2) Take an ascending sequence $l_0 \leq_L l_1 \leq_L l_2 \leq_L \ldots$ and consider the sequence $l_0^\nabla_L \leq_L l_1^\nabla_L \leq_L l_2^\nabla_L \leq_L \ldots$, where $l_0^\nabla_L = l_0$ and $l_{i+1}^\nabla_L = l_i^\nabla_L \nabla_L l_{i+1}$ for all $i$.

Since $\alpha$ is monotonic, we have $\alpha(l_0) \leq_M \alpha(l_1) \leq_M \ldots$.

Since $\nabla_M$ is a widening operator, there exists $n_0$ such that $\alpha(l_n)^\nabla_M = \alpha(l_{n_0})^\nabla_M$ for all $n \geq n_0$. We can show that for all $n \geq 0$ it holds that $\alpha(l_n)^\nabla_M = \alpha(l_n^\nabla_L)$. Then, these two facts give us

$$\alpha(l_n^\nabla_L) = \alpha(l_{n_0}^\nabla_L) \text{ for all } n \geq n_0.$$ 

Thus, $\gamma(\alpha(l_n^\nabla_L)) = \gamma(\alpha(l_{n_0}^\nabla_L))$ for all $n \geq n_0$. Since every $l_n^\nabla_L$ for $n > 0$ is of the form $\gamma(m_n)$ for some $m_n \in M$, and since $\gamma \circ \alpha \circ \gamma = \gamma$, it follows that

$$l_n^\nabla_L = l_{n_0}^\nabla_L \text{ for all } n \geq n_0.$$ 

To complete the proof it remains to show that for all $n \geq 0$,

$$\alpha(l_n)^\nabla_M = \alpha(l_n^\nabla_L).$$
Proof of the second statement (continued)

We prove by induction on \( n \) that \( \alpha(l_n)^\nabla^M = \alpha(l_n^\nabla^L) \).

For \( n = 0 \), \( \alpha(l_0)^\nabla^M = \alpha(l_0) = \alpha(l_0^\nabla^L) \).

Induction hypothesis: \( \alpha(l_n)^\nabla^M = \alpha(l_n^\nabla^L) \).
For \( n + 1 \) we get:

\[
\alpha(l_{n+1})^\nabla^M = \alpha(l_n)^\nabla^M \nabla^M \alpha(l_{n+1}) \\
= \alpha(l_n^\nabla^L) \nabla^M \alpha(l_{n+1}) \quad \text{by I.H.}
\]

\[
\alpha(l_{n+1}^\nabla^L) = \alpha(l_n^\nabla^L \nabla^L l_{n+1}) \\
= \alpha(\gamma(\alpha(l_n^\nabla^L) \nabla^M \alpha(l_{n+1}))) \quad \text{by def. } \nabla^L \\
= \alpha(l_n^\nabla^L) \nabla^M \alpha(l_{n+1}) \quad \text{by def. Galois insertion}
\]

Thus, \( \alpha(l_{n+1})^\nabla^M = \alpha(l_{n+1}^\nabla^L) \).
Lemma Let

- \((L, \leq_L)\) and \((M, \leq_M)\) be complete lattices,
- \(\alpha : L \to M\) and \(\gamma : M \to L\) be monotonic functions such that \(L \xrightarrow{\alpha} M\) is a Galois insertion, such that \(\gamma(\perp_M) = \perp_L\)
- \(\nabla_M : M \times M \to M\) 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 for all monotonic \(f : L \to L\)

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

Remark If \((M, \leq_M)\) has finite height, we can take \(\nabla_M\) to be \(\sqcup_M\), and then we will have \(\text{lfp}_{\nabla_L}(f) = \gamma(\text{lfp}(\alpha \circ f \circ \gamma))\).