1 Introduction

Reasoning about probrams bla bla bla

2 Programming Language

2.1 Syntax

We start by defining an imperative programming language with non deterministic choice and parametric on a set *Base* of base commands, common choices for the set of base commands usually contain a command for assignments and boolean guards.

The set of valid $\mathbb{C}$ programs is defined by the following inductive definition:

Definition 1 ($\mathbb{C}$ language syntax)

$$\begin{array}{rlr}
\mathbb{C} ::= & \mathbb{0} & \textit{Always non terminating program} \\
\mid & \mathbb{1} & \textit{Identity program (skip)} \\
\mid & b & \textit{Base command} \\
\mid & C_1 \fatsemi C_2 & \textit{Program composition} \\
\mid & C_1 + C_2 & \textit{Non deterministic choice} \\
\mid & C^\star & \textit{Iteration}
\end{array}$$

Where $C, C_1, C_2 \in \mathbb{C}$ and $b \in Base$.

2.2 Semantics

Given a set $\mathbb{S}$ of states and a family of partial functions $\llbracket b \rrbracket_{base} : \mathbb{S} \hookrightarrow \mathbb{S}$ we can define inductively the semantics of a program in $\mathbb{C}$ by structural induction on the terms:

Definition 2 ($\mathbb{C}$ language semantics)

$$\begin{aligned}
\llbracket \cdot \rrbracket \ &: \ \mathcal{P}(\mathbb{S}) \to \mathcal{P}(\mathbb{S}) \\
\llbracket \mathbb{0} \rrbracket &= const\ \emptyset \\
\llbracket \mathbb{1} \rrbracket &= id \\
\llbracket b \rrbracket &= \lambda P \to \{x \mid \llbracket b \rrbracket_{base}(p) \downarrow = x \ \wedge p \ \in P\} \\
\llbracket C_1 \fatsemi C_2 \rrbracket &= \llbracket C_2 \rrbracket \circ \llbracket C_1 \rrbracket \\
\llbracket C_1 + C_2 \rrbracket &= \lambda P \to \llbracket C_1 \rrbracket P \cup \llbracket C_2 \rrbracket P \\
\llbracket C^\star \rrbracket &= \lambda P \to lfp(\lambda P' \to P \cup \llbracket C \rrbracket P')
\end{aligned}$$

Theorem 1 ($\llbracket \cdot \rrbracket$ is monotone)

$$P \subseteq Q \implies \llbracket C \rrbracket(P) \subseteq \llbracket C \rrbracket(Q)$$

This framework is general enough to cover non deterministic imperative languages, for example if we include a base command for boolean guards $e?$ where e is a boolean valued expression on $\mathbb{S}$ we can define the usual control flow statements `if` b `then` C_1 `else` C_2 as $(e? \fatsemi C_1) + (\neg e? \fatsemi C_2)$ and `while` e `do` C `done` as $(e? \fatsemi C)^\star \fatsemi \neg e?$.

3 Best Abstract Inductive semantics

We can abstract the *Best Abstract Inductive Semantics* of a program in $\mathbb{C}$ parametrically on a complete lattice A that has as elements collections of program states and a family of monotone functions $\llbracket b \rrbracket_{base} : A \to A \quad \forall\ b \in \mathit{Base}$ by structural induction on the syntax of $\mathbb{C}$:

Definition 3 (Best abstract inductive semantics)

$$\begin{aligned}
\llbracket \cdot \rrbracket^A_{bai} &: A \to A \\
\llbracket \mathbb{0} \rrbracket^A_{bai} &= \mathit{const}\ \bot_A \\
\llbracket \mathbb{1} \rrbracket^A_{bai} &= id_A \\
\llbracket b \rrbracket^A_{bai} &= \lambda P \to \llbracket b \rrbracket_{base}(P) \\
\llbracket C_1 \fatsemi C_2 \rrbracket^A_{bai} &= \llbracket C_2 \rrbracket^A_{bai} \text{o} \llbracket C_1 \rrbracket^A_{bai} \\
\llbracket C_1 + C_2 \rrbracket^A_{bai} &= \lambda P \to \llbracket C_1 \rrbracket^A_{bai}(P) \vee_A \llbracket C_2 \rrbracket^A_{bai}(P) \\
\llbracket C^\star \rrbracket^A_{bai} &= \lambda P \to lfp(\lambda P' \to P \vee_A \llbracket C \rrbracket^A_{bai} P')
\end{aligned}$$

Theorem 2 (Semantic equivalence) *If we take as the lattice $\mathcal{P}(\mathbb{S})$ and as $\llbracket b \rrbracket^A_{bai} = \lambda P \to \{x \mid \llbracket b \rrbracket(p) \downarrow = x\ \wedge\ p \in P\}$ the two semantics are identical.*

$$\llbracket C \rrbracket^{\mathcal{P}(\mathbb{S})}_{bai}(P) = \llbracket C \rrbracket(P)$$

Theorem 3 ($\llbracket \cdot \rrbracket^A_{bai}$ is monotone)

$$P \leq_A Q \implies \llbracket C \rrbracket^A_{bai}(P) \leq_A \llbracket C \rrbracket^A_{bai}(Q)$$

3.1 Obtaining BAIs from other BAIs via Galois connections

Given a Galois connection $\langle D, \leq_D \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leq_A \rangle$ if we have define a *Best Abstract Inductive Semantics* on D with the semantics of basic commands $\llbracket b \rrbracket^D_{bai}$ we can define a *Best Abstract Inductive Semantics* on A with semantics of basic commands $\llbracket b \rrbracket^A_{bai} = \alpha \circ \llbracket b \rrbracket^D_{bai} \circ \gamma$.

Theorem 4 (Soundness)

$$\alpha(\llbracket C \rrbracket^D_{bai}(P) \leq_D \llbracket C \rrbracket^A_{bai}(\alpha(P))$$

(Here soundness is intended in an abstract interpretation sense)

4 Abstract Hoare logic

In this section we will define when an Abstract Hoare triple is valid, give some inference rules for it BLAH.

Definition 4 (Abstract Hoare triple) *Fixed a complete lattice A and the semantics of the base commands $[\![b]\!]^A_{bai}$, an Abstract Hoare triple is valid if and only if executing the bai of a command C of some precondition captured by the element P of A is overapproximated by some element Q of A:*

$$\langle P\rangle_A\ C\ \langle Q\rangle \iff [\![C]\!]^A_{bai}(P) \leq_A Q$$

4.1 Inference rule

As in standard Hoare logic we can give a set of inference rules to derive valid triples from smaller ones:

Definition 5 (Abstract Hoare inference rules) $$\frac{}{\vdash \langle P\rangle_A\ \mathbb{0}\ \langle \bot\rangle}\ (\mathbb{0})$$

$$\frac{}{\vdash \langle P\rangle_A\ \mathbb{1}\ \langle P\rangle}\ (\mathbb{1})$$

$$\frac{}{\vdash \langle P\rangle_A\ b\ \langle [\![b]\!]_{base}(P)\rangle}\ (b)$$

$$\frac{\vdash \langle P\rangle_A\ C_1\ \langle Q\rangle \qquad \vdash \langle Q\rangle_A\ C_2\ \langle R\rangle}{\vdash \langle P\rangle_A\ C_1 \fatsemi C_2\ \langle R\rangle}\ (\fatsemi)$$

$$\frac{\vdash \langle P\rangle_A\ C_1\ \langle Q\rangle \qquad \vdash \langle P\rangle_A\ C_2\ \langle Q\rangle}{\vdash \langle P\rangle_A\ C_1 + C_2\ \langle Q\rangle}\ (+)$$

$$\frac{\vdash \langle P\rangle_A\ C\ \langle P\rangle}{\vdash \langle P\rangle_A\ C^*\ \langle P\rangle}\ (*)$$

$$\frac{P \leq P' \qquad \vdash \langle P'\rangle_A\ C\ \langle Q'\rangle \qquad Q' \leq Q}{\vdash \langle P\rangle_A\ C\ \langle Q\rangle}\ (\leq)$$

Example 4.1 (Derivation in the integer interval domain) *We will start by defining our running example the integer interval domain: The elements of the interval domain on one variable are defined by $Int = \{[x,y] \mid x \leq y \quad x,y \in \mathbb{Z} \cup \{+\infty, -\infty\}\} \cup \{\bot\}$ where the order relation is given by $\bot \leq_{Int} a\ \forall a \in Int$ and $[a,b] \leq_{Int} [c,d] \iff c \leq a \land b \leq d$ and clearly $\top = [-\infty, +\infty]$. The definition can be lifted pointwise for domains with an arbitrary number of variables.*

As basic command we will add $e?$ for boolean tests discarding all the states that don't satisfy the condition e and $x := y$ assigning the result of evaluating expression y to the variable x and fix $[\![b]\!]^A_{bai} = \alpha \circ [\![b]\!] \circ \gamma$.

Then the following is a valid derivation for program $C = (x := 1 + x := 2) \fatsemi x := x + 1$

$$\dfrac{\dfrac{\dfrac{\top \leq \top \quad \vdash \langle \top \rangle_A\ x := 1\ \langle x \in [1,1] \rangle \quad x \in [1,1] \leq x \in [1,2]}{\vdash \langle \top \rangle_A\ x := 1\ \langle x \in [1,2] \rangle} \quad \dfrac{\top \leq \top \quad \vdash \langle \top \rangle_A\ x := 2\ \langle x \in [2,2] \rangle \quad x \in [2,2] \leq x \in [1,2]}{\vdash \langle \top \rangle_A\ x := 2\ \langle x \in [1,2] \rangle}}{\vdash \langle \top \rangle_A\ x := 1 + x := 2\ \langle x \in [1,2] \rangle} \quad \vdash \langle x \in [1,2] \rangle_A\ x := x + 1\ \langle x \in [2,3] \rangle}{\vdash \langle \top \rangle_A\ C\ \langle x \in [2,3] \rangle}$$

And clearly $[\![C]\!]^A_{bai}(\top) = x \in [2,3]$.

Theorem 5 (The proofsystem is sound)

$$\vdash \langle P \rangle_A\ C\ \langle Q \rangle \implies \langle P \rangle_A\ C\ \langle Q \rangle$$

Proof 1 *By structural induction on the last rule applied in the derivation of $\vdash \langle P \rangle_A\ C\ \langle Q \rangle$:*

- $(\mathbb{0})$*: Then the last step in the derivation was:*

  $$\dfrac{}{\vdash \langle P \rangle_A\ \mathbb{0}\ \langle \bot \rangle}\ (\mathbb{0})$$

  The triple is valid since:

  $$[\![\mathbb{0}]\!]^A_{bai}(P) = \bot_A \qquad \textit{By definition of } [\![\cdot]\!]^A_{bai}$$

- $(\mathbb{1})$*: Then the last step in the derivation was:*

  $$\dfrac{}{\vdash \langle P \rangle_A\ \mathbb{1}\ \langle P \rangle}\ (\mathbb{1})$$

  The triple is valid since:

  $$[\![\mathbb{1}]\!]^A_{bai}(P) = P \qquad \textit{By definition of } [\![\cdot]\!]^A_{bai}$$

- (b)*: Then the last step in the derivation was:*

  $$\dfrac{}{\vdash \langle P \rangle_A\ b\ \langle [\![b]\!]_{base}(P) \rangle}\ (b)$$

  The triple is valid since:

  $$[\![b]\!]^A_{bai}(P) = [\![b]\!]_{base}(P) \qquad \textit{By definition of } [\![\cdot]\!]^A_{bai}$$

- $(\fatsemi)$*: Then the last step in the derivation was:*

  $$\dfrac{\vdash \langle P \rangle_A\ C_1\ \langle Q \rangle \qquad \vdash \langle Q \rangle_A\ C_2\ \langle R \rangle}{\vdash \langle P \rangle_A\ C_1 \fatsemi C_2\ \langle R \rangle}\ (\fatsemi)$$

  By inductive hypothesis: $[\![C_1]\!]^A_{bai}(P) \leq_A Q$ and $[\![C_2]\!]^A_{bai}(Q) \leq_A R$.

  The triple is valid since:

  $$\begin{aligned} [\![C_1 \fatsemi C_2]\!]^A_{bai}(P) &= [\![C_2]\!]^A_{bai}([\![C_1]\!]^A_{bai}(P)) && \textit{By definition of } [\![\cdot]\!]^A_{bai} \\ &\leq_A [\![C_2]\!]^A_{bai}(Q) && \textit{By monotonicity of } [\![\cdot]\!]^A_{bai} \\ &\leq_A R \end{aligned}$$

- $(+)$*: Then the last step in the derivation was:*

$$\frac{\vdash \langle P\rangle_A\ C_1\ \langle Q\rangle \qquad \vdash \langle P\rangle_A\ C_2\ \langle Q\rangle}{\vdash \langle P\rangle_A\ C_1 + C_2\ \langle Q\rangle}\ (+)$$

By inductive hypothesis: $\llbracket C_1\rrbracket^A_{bai}(P) \leq Q$ *and* $\llbracket C_2\rrbracket^A_{bai}(P) \leq Q$*.*

The triple is valid since:

$$\begin{aligned} \llbracket C_1 + C_2\rrbracket^A_{bai}(P) &= \llbracket C_1\rrbracket^A_{bai}(P) \vee \llbracket C_2\rrbracket^A_{bai}(P) \quad && \textit{By definition of } \llbracket\cdot\rrbracket^A_{bai} \\ &\leq_A Q \vee Q \\ &= Q \end{aligned}$$

- $(\star)$*: Then the last step in the derivation was:*

$$\frac{\vdash \langle P\rangle_A\ C\ \langle P\rangle}{\vdash \langle P\rangle_A\ C^\star\ \langle P\rangle}\ (*)$$

By inductive hypothesis: $\llbracket C\rrbracket^A_{bai}P \leq P$

$$\llbracket C^\star\rrbracket^A_{bai}(P) = lfp(\lambda P' \to P \vee_A \llbracket C\rrbracket^A_{bai}(P'))$$

$$\begin{aligned} (\lambda P' \to P \vee_A \llbracket C\rrbracket^A_{bai}(P'))(P) &= P \vee \llbracket C\rrbracket^A_{bai}(P) \quad && \textit{since } \llbracket C\rrbracket^A_{bai}(P) \leq P \\ &= P \end{aligned}$$

Hence P *is a fixpoint for* $\lambda P' \to P \vee_A \llbracket C\rrbracket^A_{bai}(P')$

Thus $lfp(\lambda P' \to P \vee_A \llbracket C\rrbracket^A_{bai}(P')) \leq_A P$

- $(\leq)$*: Then the last step in the derivation was:*

$$\frac{P \leq P' \qquad \vdash \langle P'\rangle_A\ C\ \langle Q'\rangle \qquad Q' \leq Q}{\vdash \langle P\rangle_A\ C\ \langle Q\rangle}\ (\leq)$$

By inductive hypothesis: $\llbracket C\rrbracket^A_{bai}(P') \leq Q'$*.*

$$\begin{aligned} \llbracket C\rrbracket^A_{bai}(P)\llbracket C\rrbracket^A_{bai}(P') \quad && \textit{By monotonicity of } \llbracket\cdot\rrbracket^A_{bai} \\ \leq Q' \quad && \textit{By inductive hypothesis} \\ \leq Q \end{aligned}$$

Theorem 6 (Relative $[\![\cdot]\!]^A_{bai}$-completeness)

$$\vdash \langle P\rangle_A\ C\ \langle [\![C]\!]^A_{bai}(P)\rangle$$

Proof 2 *By structural induction on C:*

- $\mathbb{0}$*: By definition* $[\![\mathbb{0}]\!]^A_{bai}(P) = \bot$

$$\dfrac{}{\vdash \langle P\rangle_A\ \mathbb{0}\ \langle \bot\rangle}\,(\mathbb{0})$$

- $\mathbb{1}$*: By definition* $[\![\mathbb{1}]\!]^A_{bai}(P) = P$

$$\dfrac{}{\vdash \langle P\rangle_A\ \mathbb{1}\ \langle P\rangle}\,(\mathbb{1})$$

- b*: By definition* $[\![b]\!]^A_{bai}(P) = [\![b]\!]_{base}(P)$

$$\dfrac{}{\vdash \langle P\rangle_A\ b\ \langle [\![b]\!]_{base}(P)\rangle}\,(b)$$

- $C_1 \fatsemi C_2$*: By definition* $[\![C_1 \fatsemi C_2]\!]^A_{bai}(P) = [\![C_2]\!]^A_{bai}([\![C_1]\!]^A_{bai}(P))$

$$\dfrac{\overset{\textit{(Inductive hypothesis)}}{\vdash \langle P\rangle_A\ C_1\ \langle [\![C_1]\!]^A_{bai}(P)\rangle} \qquad \overset{\textit{(Inductive hypothesis)}}{\vdash \langle [\![C_1]\!]^A_{bai}(P)\rangle_A\ C_2\ \langle [\![C_2]\!]^A_{bai}([\![C_1]\!]^A_{bai}(P))\rangle}}{\vdash \langle P\rangle_A\ C_1 \fatsemi C_2\ \langle [\![C_2]\!]^A_{bai}([\![C_1]\!]^A_{bai}(P))\rangle}\,(\fatsemi)$$

- $C_1 + C_2$*: By definition* $[\![C_1 + C_2]\!]^A_{bai}(P) = [\![C_1]\!]^A_{bai}(P) \vee [\![C_2]\!]^A_{bai}(P)$

$$\dfrac{\dfrac{P \le P \qquad \overset{\textit{(Inductive hypothesis)}}{\vdash \langle P\rangle_A\ C_1\ \langle [\![C_1]\!]^A_{bai}(P)\rangle} \qquad [\![C_1]\!]^A_{bai}(P) \le [\![C_1]\!]^A_{bai}(P) \vee [\![C_2]\!]^A_{bai}(P)}{\vdash \langle P\rangle_A\ C_1\ \langle [\![C_1]\!]^A_{bai}(P) \vee [\![C_2]\!]^A_{bai}(P)\rangle}\,(\le) \qquad \pi_1}{\vdash \langle P\rangle_A\ C_1 + C_2\ \langle [\![C_1]\!]^A_{bai}(P) \vee [\![C_2]\!]^A_{bai}(P)\rangle}\,(+)$$

Where π_1*:*

$$\dfrac{P \le P \qquad \overset{\textit{(Inductive hypothesis)}}{\vdash \langle P\rangle_A\ C_2\ \langle [\![C_2]\!]^A_{bai}(P)\rangle} \qquad [\![C_2]\!]^A_{bai}(P) \le [\![C_1]\!]^A_{bai}(P) \vee [\![C_2]\!]^A_{bai}(P)}{\vdash \langle P\rangle_A\ C_2\ \langle [\![C_1]\!]^A_{bai}(P) \vee [\![C_2]\!]^A_{bai}(P)\rangle}\,(\le)$$

- $C^\star$*: By definition* $[\![C^\star]\!]^A_{bai}(P) = lfp(\lambda p' \to p \vee [\![c]\!]^A_{bai}(s'))$ *and let's call this value* K*, since* K *is a fixpoint of* $\lambda P' \to P \vee [\![C]\!]^A_{bai}(S')$*:*

$$K = P \vee [\![C]\!]^A_{bai}(K)$$

Hence:

- *α_1: $K \geq P$*
- *α_2: $K \geq \llbracket C \rrbracket^A_{bai}(K)$*

$$\dfrac{\alpha_1 \qquad \dfrac{\dfrac{K \leq K \qquad \overset{\textit{(Inductive hypothesis)}}{\vdash \langle K \rangle_A \; C \; \langle \llbracket C \rrbracket^A_{bai}(K) \rangle} \qquad \alpha_2}{\vdash \langle K \rangle_A \; C \; \langle K \rangle}\,(\leq)}{\vdash \langle K \rangle_A \; C^\star \; \langle K \rangle}\,(\star) \qquad K \leq K}{\vdash \langle P \rangle_A \; C^\star \; \langle K \rangle}\,(\leq)$$

Theorem 7 (Relative completeness)

$$\langle P \rangle_A \; C \; \langle Q \rangle \implies \vdash \langle P \rangle_A \; C \; \langle Q \rangle$$

Proof 3 *By definition of $\langle P \rangle_A \; C \; \langle Q \rangle \iff Q \geq \llbracket C \rrbracket^A_{bai}(P)$*

$$\dfrac{P \leq P \qquad \overset{\textit{By Theorem 6}}{\vdash \langle P \rangle_A \; C \; \langle \llbracket C \rrbracket^A_{bai}(P) \rangle} \qquad Q \geq \llbracket C \rrbracket^A_{bai}(P)}{\vdash \langle P \rangle_A \; C \; \langle Q \rangle}\,(\leq)$$

4.2 Merge rules

In standard Hoare loigc the following rule is valid:

$$\dfrac{\vdash \{P_1\} \; C \; \{Q\} \qquad \vdash \{P_2\} \; C \; \{Q\}}{\vdash \{P_1 \vee P_2\} \; C \; \{Q\}}\,(merge)$$

And can be used to easily merge the results obtained by different analysis, but in general the same rule in the context of abstract Hoare logic:

$$\dfrac{\vdash \langle P_1 \rangle_A \; C \; \langle Q \rangle \qquad \vdash \langle P_2 \rangle_A \; C \; \langle Q \rangle}{\vdash \langle P_1 \vee P_2 \rangle_A \; C \; \langle Q \rangle}\,(merge)$$

is unsound.

Example 4.2 (Unsoundness of the rule $(merge)$**)** *Let A and $\llbracket b \rrbracket^A_{bai}$ be the same as in 4.1 and $C = (x = 3? \fatsemi x := 400) + (x \neq 3? \fatsemi x := x + 1)$*

Then we can derive the following triples:

- $\vdash \langle x \in [1, 2] \rangle_A \; C \; \langle x \in [2, 3] \rangle$
- $\vdash \langle x \in [4, 5] \rangle_A \; C \; \langle x \in [5, 6] \rangle$

But applying the rule $(merge)$ would allow to derive the following: $\vdash \langle x \in [1, 5] \rangle_A \; C \; \langle x \in [2, 6] \rangle$, that is unsound since $\llbracket C \rrbracket^A_{bai}(x \in [1, 5]) = x \in [2, 400]$.

The cause of the issue is caused by the non additivity of the base commands, in fact:

Theorem 8 (Additivity of the semantics on attitive base commands) *if* $\forall P_1, P_2 \in A$ *and* $b \in Base$ $\llbracket B \rrbracket^A_{bai}(P_1 \vee P_2) = \llbracket B \rrbracket^A_{bai}(P_1) \vee \llbracket P_2 \rrbracket^A_{bai}$ *then:* $\forall P_1, P_2 \in A \quad C \in \mathbb{C}$

$$\llbracket C \rrbracket^A_{bai}(P_1) \vee \llbracket C \rrbracket^A_{bai}(P_2) = \llbracket C \rrbracket^A_{bai}(P_1 \vee P_2)$$

Proof 4 *By structural induction on* C*:*

- $\mathbb{0}$*: By definition* $\llbracket \mathbb{0} \rrbracket^A_{bai}(P_1 \vee P_2) = \bot$ *and* $\llbracket \mathbb{0} \rrbracket^A_{bai}(P_i) = \bot$
- $\mathbb{1}$*: By definition* $i \in \{1,2\}$ $\llbracket \mathbb{1} \rrbracket^A_{bai}(P_i) = P_i$ *and* $\llbracket \mathbb{1} \rrbracket^A_{bai}(P_1 \vee P_2) = P_1 \vee P_2$
- b*:*

$$\llbracket b \rrbracket^A_{bai}(P_1 \vee P_2) = \llbracket b \rrbracket^A_{bai}(P_1) \vee \llbracket b \rrbracket^A_{bai}(P_2) \qquad \textit{By additivity of } \llbracket b \rrbracket^A_{bai}$$

- $C_1 \fatsemi C_2$*:*

$$\begin{aligned}
\llbracket C_1 \fatsemi C_2 \rrbracket^A_{bai}(P_1 \vee P_2) &= \llbracket C_2 \rrbracket^A_{bai}(\llbracket C_1 \rrbracket^A_{bai}(P_1 \vee P_2)) \\
&= \llbracket C_2 \rrbracket^A_{bai}(\llbracket C_1 \rrbracket^A_{bai}(P_1) \vee \llbracket C_1 \rrbracket^A_{bai}(P_2)) \quad \textit{By inductive hypothesis} \\
&= \llbracket C_1 \fatsemi C_2 \rrbracket^A_{bai}(P_1) \vee \llbracket C_2 \fatsemi C_2 \rrbracket^A_{bai}(P_2) \quad \textit{By inductive hypothesis}
\end{aligned}$$

- $C_1 + C_2$*:*

$$\begin{aligned}
\llbracket C_1 + C_2 \rrbracket^A_{bai}(P_1 \vee P_2) &= \llbracket C_1 \rrbracket^A_{bai}(P_1 \vee P_2) \vee \llbracket C_2 \rrbracket^A_{bai}(P_1 \vee P_2) \\
&= \llbracket C_1 \rrbracket^A_{bai}(P_1) \vee \llbracket C_2 \rrbracket^A_{bai}(P_2) \\
&\quad \vee \llbracket C_2 \rrbracket^A_{bai}(P_1) \vee \llbracket C_2 \rrbracket^A_{bai}(P_2) \qquad \textit{By inductive hypothesis} \\
&= \llbracket C_1 + C_2 \rrbracket^A_{bai}(P_1) \vee \llbracket C_1 + C_2 \rrbracket^A_{bai}(P_2) \qquad \textit{Rearranging } \vee
\end{aligned}$$

- $C^\star$*:*

$$\llbracket C^\star \rrbracket^A_{bai}(P_1 \vee P_2) = lfp(\lambda P' \to P_1 \vee P_2 \vee \llbracket C \rrbracket^A_{bai}(P'))$$

Let $F_i = lfp(\lambda P' \to P_i \vee \llbracket C \rrbracket^A_{bai}(P'))$

$$\begin{aligned}
(\lambda P' \to P_1 \vee P_2 \vee_A \llbracket C \rrbracket^A_{bai}(P'))(F_1 \vee F_2) &= P_1 \vee P_2 \vee \llbracket C \rrbracket^A_{bai}(F_1 \vee F_2) \\
&\textit{By inductive hypothesis} \\
&= P_1 \vee P_2 \vee \llbracket C \rrbracket^A_{bai}(F_1) \vee \llbracket C \rrbracket^A_{bai}(F_2) \\
&= P_1 \vee \llbracket C \rrbracket^A_{bai}(F_1) \vee P_2 \vee \llbracket C \rrbracket^A_{bai}(F_2) \\
&= F_1 \vee F_2
\end{aligned}$$

Thus $F_1 \vee F_2$ is a fixpoint for $\lambda P' \to P_i \vee [\![C]\!]^A_{bai}(P')$.

Following the same reasoning is also the least one by $F_1 \vee F_2$ beeing the smallest $K \geq F_1$ and $K \geq F_2$

Theorem 9 (Soundness of the rule *merge* on attitive base commands) *if $\forall P_1, P_2 \in A$ and $b \in Base$ $[\![B]\!]^A_{bai}(P_1 \vee P_2) = [\![B]\!]^A_{bai}(P_1) \vee [\![P_2]\!]^A_{bai}$ then: $\forall P_2, P_2, Q \in A \quad C \in \mathbb{C}$*

$$[\![C]\!]^A_{bai}(P_1) \leq Q \text{ and } [\![C]\!]^A_{bai}(P_2) \leq Q \implies [\![C]\!]^A_{bai}(P_1 \vee P_2) \leq Q$$

Proof 5 *Let $S_i = [\![C]\!]^A_{bai}(P_i)$ by Theorem 8 $[\![C]\!]^A_{bai}(P_1 \vee P_2) = [\![C]\!]^A_{bai}(P_1) \vee [\![C]\!]^A_{bai}(P_2) \leq Q \vee Q = Q$*

But requiring all the base commands to be additive is a strong requirement with $\mathcal{P}(\mathbb{S})$ the requirement is equivalent as requiring γ especially when the base commands are defined trough a Galois connection to be additive, a requiring satisfied by a very small portion of abstract domains utilized in practice.

Theorem 10 (Base commands additivity) *Given a Galois connection $\langle D, \leq_D \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leq_A \rangle$ and $[\![b]\!]^D_{bai}$ additive.*

$$\alpha \circ [\![b]\!]^D_{bai} \circ \gamma \text{ additive} \iff \gamma \text{ additive}$$

Proof 6

$$\begin{aligned}
\alpha([\![b]\!]^A_{bai}(\gamma(P_1 \vee P_2))) &= \alpha([\![b]\!]^A_{bai}(\gamma(P_1) \vee \gamma(P_2))) \\
&= \alpha([\![b]\!]^A_{bai}(\gamma(P_1)) \vee [\![b]\!]^A_{bai}(\gamma(P_2))) \\
&= \alpha([\![b]\!]^A_{bai}(\gamma(P_1))) \vee \alpha([\![b]\!]^A_{bai}(\gamma(P_2)))
\end{aligned}$$

4.2.1 Local condition

Given that additivity conditions on the domain A are too restrictive we will look at local conditions.

One could think that requiring γ to be additive on $P_1 \vee P_2$ is enough giving a rule like this:

$$\frac{\vdash \langle P_1 \rangle_A \; C \; \langle Q \rangle \qquad \vdash \langle P_2 \rangle_A \; C \; \langle Q \rangle \qquad \gamma(P_1 \vee P_2) = \gamma(P_1) \vee \gamma(P_2)}{\vdash \langle P_1 \vee P_2 \rangle_A \; C \; \langle Q \rangle} \; (merge)$$

Example 4.3 ((*merge*) rule is unsound) *Let A and $[\![b]\!]^A_{bai}$ be the same as in 4.1.*

By picking $P_1 = x \in [0,1]$ and $P_2 = x \in [2,2]$ we can obtain the following judgments:

- $\langle P_1 \rangle_A \; C \; \langle \bot \rangle$

- $\langle P_2\rangle_A\ C\ \langle \bot\rangle$

And by applying the $(merge)$ *rule we can obtain:* $\langle P_1 \vee P_2\rangle_A\ C\ \langle Q\rangle$ *but by running command* C *on* $P_1 \vee P_2$ *we obtain* $x \in [1,1]$.

- α_1*:* $K \geq P$
- α_2*:* $K \geq [\![C]\!]^A_{bai}(K)$

$$\dfrac{\alpha_1 \qquad \dfrac{\dfrac{K \leq K \qquad \overset{\textit{(Inductive hypothesis)}}{\vdash \langle K\rangle_A\ C\ \langle [\![C]\!]^A_{bai}(K)\rangle} \qquad \alpha_2}{\vdash \langle K\rangle_A\ C\ \langle K\rangle}\,(\leq)}{\vdash \langle K\rangle_A\ C^\star\ \langle K\rangle}\,(\star) \qquad K \leq K}{\vdash \langle P\rangle_A\ C^\star\ \langle K\rangle}\,(\leq)$$

Theorem 11 (Relative completeness)

$$\langle P\rangle_A\ C\ \langle Q\rangle \implies \vdash \langle P\rangle_A\ C\ \langle Q\rangle$$

Proof 7 *By definition of* $\langle P\rangle_A\ C\ \langle Q\rangle \iff Q \geq [\![C]\!]^A_{bai}(P)$

$$\dfrac{P \leq P \qquad \overset{\textit{By Theorem 6}}{\vdash \langle P\rangle_A\ C\ \langle [\![C]\!]^A_{bai}(P)\rangle} \qquad Q \geq [\![C]\!]^A_{bai}(P)}{\vdash \langle P\rangle_A\ C\ \langle Q\rangle}\,(\leq)$$

4.3 Merge rules

In standard Hoare loigc the following rule is valid:

$$\dfrac{\vdash \{P_1\}\ C\ \{Q\} \qquad \vdash \{P_2\}\ C\ \{Q\}}{\vdash \{P_1 \vee P_2\}\ C\ \{Q\}}\,(merge)$$

And can be used to easily merge the results obtained by different analysis, but in general the same rule in the context of abstract Hoare logic:

$$\dfrac{\vdash \langle P_1\rangle_A\ C\ \langle Q\rangle \qquad \vdash \langle P_2\rangle_A\ C\ \langle Q\rangle}{\vdash \langle P_1 \vee P_2\rangle_A\ C\ \langle Q\rangle}\,(merge)$$

is unsound.

Example 4.4 (Unsoundness of the rule $(merge)$**)** *Let* A *and* $[\![b]\!]^A_{bai}$ *be the same as in 4.1 and* $C = (x = 3? \fatsemi x := 400) + (x \neq 3? \fatsemi x := x + 1)$

Then we can derive the following triples:

- $\vdash \langle x \in [1,2]\rangle_A\ C\ \langle x \in [2,3]\rangle$
- $\vdash \langle x \in [4,5]\rangle_A\ C\ \langle x \in [5,6]\rangle$

But applying the rule (merge) would allow to derive the following: $\vdash \langle x \in [1,5]\rangle_A\ C\ \langle x \in [2,6]\rangle$*, that is unsound since* $[\![C]\!]^A_{bai}(x \in [1,5]) = x \in [2,400]$.

The cause of the issue is caused by the non additivity of the base commands, in fact:

Theorem 12 (Additivity of the semantics on attitive base commands) *if* $\forall P_1, P_2 \in A$ *and* $b \in Base$ $[\![B]\!]^A_{bai}(P_1 \vee P_2) = [\![B]\!]^A_{bai}(P_1) \vee [\![P_2]\!]^A_{bai}$ *then:* $\forall P_1, P_2 \in A \quad C \in \mathbb{C}$

$$[\![C]\!]^A_{bai}(P_1) \vee [\![C]\!]^A_{bai}(P_2) = [\![C]\!]^A_{bai}(P_1 \vee P_2)$$

Proof 8 *By structural induction on* C*:*

- $\mathbb{0}$*: By definition* $[\![\mathbb{0}]\!]^A_{bai}(P_1 \vee P_2) = \perp$ *and* $[\![\mathbb{0}]\!]^A_{bai}(P_i) = \perp$
- $\mathbb{1}$*: By definition* $i \in \{1,2\}$ $[\![\mathbb{1}]\!]^A_{bai}(P_i) = P_i$ *and* $[\![\mathbb{1}]\!]^A_{bai}(P_1 \vee P_2) = P_1 \vee P_2$
- b*:*

$$[\![b]\!]^A_{bai}(P_1 \vee P_2) = [\![b]\!]^A_{bai}(P_1) \vee [\![b]\!]^A_{bai}(P_2) \qquad \text{By additivity of } [\![b]\!]^A_{bai}$$

- $C_1 \fatsemi C_2$*:*

$$\begin{aligned} [\![C_1 \fatsemi C_2]\!]^A_{bai}(P_1 \vee P_2) &= [\![C_2]\!]^A_{bai}([\![C_1]\!]^A_{bai}(P_1 \vee P_2)) \\ &= [\![C_2]\!]^A_{bai}([\![C_1]\!]^A_{bai}(P_1) \vee [\![C_1]\!]^A_{bai}(P_2)) && \text{By inductive hypothesis} \\ &= [\![C_1 \fatsemi C_2]\!]^A_{bai}(P_1) \vee [\![C_2 \fatsemi C_2]\!]^A_{bai}(P_2) && \text{By inductive hypothesis} \end{aligned}$$

- $C_1 + C_2$*:*

$$\begin{aligned} [\![C_1 + C_2]\!]^A_{bai}(P_1 \vee P_2) &= [\![C_1]\!]^A_{bai}(P_1 \vee P_2) \vee [\![C_2]\!]^A_{bai}(P_1 \vee P_2) \\ &= [\![C_1]\!]^A_{bai}(P_1) \vee [\![C_2]\!]^A_{bai}(P_2) \\ &\ \vee [\![C_2]\!]^A_{bai}(P_1) \vee [\![C_2]\!]^A_{bai}(P_2) && \text{By inductive hypothesis} \\ &= [\![C_1 + C_2]\!]^A_{bai}(P_1) \vee [\![C_1 + C_2]\!]^A_{bai}(P_2) && \text{Rearranging } \vee \end{aligned}$$

- $C^\star$*:*

$$[\![C^\star]\!]^A_{bai}(P_1 \vee P_2) = lfp(\lambda P' \to P_1 \vee P_2 \vee [\![C]\!]^A_{bai}(P'))$$

Let $F_i = lfp(\lambda P' \to P_i \vee [\![C]\!]^A_{bai}(P'))$

$$\begin{aligned}(\lambda P' \to P_1 \vee P_2 \vee_A [\![C]\!]^A_{bai}(P'))(F_1 \vee F_2) &= P_1 \vee P_2 \vee [\![C]\!]^A_{bai}(F_1 \vee F_2)\\ &\textit{By inductive hypothesis}\\ &= P_1 \vee P_2 \vee [\![C]\!]^A_{bai}(F_1) \vee [\![C]\!]^A_{bai}(F_2)\\ &= P_1 \vee [\![C]\!]^A_{bai}(F_1) \vee P_2 \vee [\![C]\!]^A_{bai}(F_2)\\ &= F_1 \vee F_2\end{aligned}$$

Thus $F_1 \vee F_2$ is a fixpoint for $\lambda P' \to P_i \vee [\![C]\!]^A_{bai}(P')$.
Following the same reasoning is also the least one by $F_1 \vee F_2$ beeing the smallest $K \geq F_1$ and $K \geq F_2$

Theorem 13 (Soundness of the rule *merge* on attitive base commands) *if $\forall P_1, P_2 \in A$ and $b \in Base$ $[\![B]\!]^A_{bai}(P_1 \vee P_2) = [\![B]\!]^A_{bai}(P_1) \vee [\![P_2]\!]^A_{bai}$ then: $\forall P_2, P_2, Q \in A \quad C \in \mathbb{C}$*

$$[\![C]\!]^A_{bai}(P_1) \leq Q \textit{ and } [\![C]\!]^A_{bai}(P_2) \leq Q \implies [\![C]\!]^A_{bai}(P_1 \vee P_2) \leq Q$$

Proof 9 *Let $S_i = [\![C]\!]^A_{bai}(P_i)$ by Theorem 8 $[\![C]\!]^A_{bai}(P_1 \vee P_2) = [\![C]\!]^A_{bai}(P_1) \vee [\![C]\!]^A_{bai}(P_2) \leq Q \vee Q = Q$*

But requiring all the base commands to be additive is a strong requirement with $\mathcal{P}(\mathbb{S})$ the requirement is equivalent as requiring γ especially when the base commands are defined trough a Galois connection to be additive, a requiring satisfied by a very small portion of abstract domains utilized in practice.

Theorem 14 (Base commands additivity) *Given a Galois connection $\langle D, \leq_D \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leq_A \rangle$ and $[\![b]\!]^D_{bai}$ additive.*

$$\alpha \circ [\![b]\!]^D_{bai} \circ \gamma \textit{ additive} \impliedby \gamma \textit{ additive}$$

Proof 10

$$\begin{aligned}\alpha([\![b]\!]^A_{bai}(\gamma(P_1 \vee P_2))) &= \alpha([\![b]\!]^A_{bai}(\gamma(P_1) \vee \gamma(P_2)))\\ &= \alpha([\![b]\!]^A_{bai}(\gamma(P_1)) \vee [\![b]\!]^A_{bai}(\gamma(P_2)))\\ &= \alpha([\![b]\!]^A_{bai}(\gamma(P_1))) \vee \alpha([\![b]\!]^A_{bai}(\gamma(P_2)))\end{aligned}$$

4.3.1 Local condition

Given that additivity conditions on the domain A are too restrictive we will look at local conditions.

One could think that requiring γ to be additive on $P_1 \vee P_2$ is enough giving a rule like this:

$$\frac{\vdash \langle P_1 \rangle_A \; C \; \langle Q \rangle \qquad \vdash \langle P_2 \rangle_A \; C \; \langle Q \rangle \qquad \gamma(P_1 \vee P_2) = \gamma(P_1) \vee \gamma(P_2)}{\vdash \langle P_1 \vee P_2 \rangle_A \; C \; \langle Q \rangle} \; (merge)$$

Example 4.5 (($merge$) rule is unsound) *Let A and $[\![b]\!]^A_{bai}$ be the same as in 4.1.*

By picking $P_1 = x \in [0,1]$ and $P_2 = x \in [2,2]$ we can obtain the following judgments:

- $\langle P_1 \rangle_A \; C \; \langle \bot \rangle$
- $\langle P_2 \rangle_A \; C \; \langle \bot \rangle$

And by applying the ($merge$) rule we can obtain: $\langle P_1 \vee P_2 \rangle_A \; C \; \langle Q \rangle$ but by running command C on $P_1 \vee P_2$ we obtain $x \in [1,1]$.

$$\begin{aligned}
[\![C]\!]^A_{bai}(P_1 \vee P_2) &= [\![C]\!]^A_{bai}(x \in [0,2]) \\
&= [\![x = 1?]\!]^A_{bai}([\![x = 0? + x = 2?]\!]^A_{bai}(x \in [0,2])) \\
&= [\![x = 1?]\!]^A_{bai}([\![x = 0?]\!]^A_{bai}(x \in [0,2]) \vee [\![x = 2?]\!]^A_{bai}(x \in [0,2])) \\
&= [\![x = 1?]\!]^A_{bai}(x \in [0,0]) \vee (x \in [2,2])) \\
&= [\![x = 1?]\!]^A_{bai}(x \in [0,2]) \\
&= x \in [1,1]
\end{aligned}$$

The issue is caused by the imprecision added by the $\vee$ introduced by the non deterministic choice.

In fact the following equivalence between programs that is true on the concrete domain $(C_1 + C_2) \fatsemi C_3 \equiv (C_1 \fatsemi C_3) + (C_2 \fatsemi C_3)$ it's not true in the abstract, the program $C' = (x = 0? \fatsemi x = 1?) + (x = 2? \fatsemi x = 1?)$ that should be equivalent to C it's not: $[\![C']\!]^A_{bai}(P_1 \vee P_2) = \bot$.

The fact that programs C and C' are different when interpreted by $[\![\cdot]\!]^A_{bai}$ means that we can't look at them as a KAT, in fact this would violate the axiom $(p+q)r = pr+qr$. It should also be noted that the dual axiom $r(p+q) = rp+rq$ instead holds.

Theorem 15 $\forall C_1 C_2 C_3$

$$[\![C_1 \fatsemi (C_2 + C_3)]\!]^A_{bai} = [\![(C_1 \fatsemi C_2) + (C_1 \fatsemi C_3)]\!]^A_{bai}$$

Proof 11

$$\begin{aligned}
[\![C_1 \fatsemi (C_2 + C_3)]\!]^A_{bai}(P) &= [\![C_2]\!]^A_{bai}([\![C_1]\!]^A_{bai}(P)) \vee [\![C_3]\!]^A_{bai}([\![C_1]\!]^A_{bai}(P)) \\
&= [\![C_1 \fatsemi C_2]\!]^A_{bai}(P) \vee [\![C_1 \fatsemi C_3]\!]^A_{bai}(P) \\
&= [\![C_1 \fatsemi C_2 + C_1 \fatsemi C_3]\!]^A_{bai}(P)
\end{aligned}$$

Requiring $(C_1 + C_2) \fatsemi C_3 \equiv (C_1 \fatsemi C_3) + (C_2 \fatsemi C_3)$ is equivalent to requiring the abstract semantics of every program to be additive:

Theorem 16

$$\forall C_1 C_2 C_3 \quad \llbracket (C_1 + C_2) \fatsemi C_3 \rrbracket^A_{bai} = \llbracket (C_1 \fatsemi C_3) + (C_2 \fatsemi C_3) \rrbracket^A_{bai}$$

$$\Longleftrightarrow$$

$$\forall P_1 P_2 (decidable) C' \quad \llbracket C' \rrbracket^A_{bai}(P_1 \vee P_2) = \llbracket C_1 \rrbracket^A_{bai}(P_1) \vee \llbracket C_2 \rrbracket^A_{bai}(P_2)$$

Proof 12 $(\Longleftarrow)$*:*

$$\begin{aligned}
\llbracket (C_1 + C_2) \fatsemi C_3 \rrbracket^A_{bai}(P) &= \llbracket C_3 \rrbracket^A_{bai}(\llbracket C_1 \rrbracket^A_{bai}(P) \vee \llbracket C_2 \rrbracket^A_{bai}(P)) \\
&= \llbracket C_3 \rrbracket^A_{bai}(\llbracket C_1 \rrbracket^A_{bai}(P)) \vee \llbracket C_3 \rrbracket^A_{bai}(\llbracket C_2 \rrbracket^A_{bai}(P)) \\
&= \llbracket (C_1 \fatsemi C_3) + (C_2 \fatsemi C_3) \rrbracket^A_{bai}(P)
\end{aligned}$$

- $(\Longrightarrow)$*: For every proposition P we can construct a program $c(P)$ such that $\llbracket c(P) \rrbracket^A_{bai}(Q) = P$ (we just need to provide a program for every join-irriducible element in A)*

$$\begin{aligned}
\llbracket C \rrbracket^A_{bai}(P_1 \vee P_2) &= \llbracket C \rrbracket^A_{bai}(\llbracket c(P_1) \rrbracket^A_{bai}(Q) \vee \llbracket c(P_2) \rrbracket^A_{bai}(Q)) \\
&= \llbracket C \rrbracket^A_{bai}(\llbracket c(P_1) + c(P_2) \rrbracket^A_{bai}(Q)) \\
&= \llbracket (c(P_1) + c(P_2)) \fatsemi C \rrbracket^A_{bai}(Q)) \\
&= \llbracket (c(P_1) \fatsemi C) + (c(P_2) \fatsemi C) \rrbracket^A_{bai}(Q)) \\
&= \llbracket C \rrbracket^A_{bai}(\llbracket c(P_1) \rrbracket^A_{bai}(Q)) \vee \llbracket C \rrbracket^A_{bai}(\llbracket c(P_2) \rrbracket^A_{bai}(Q)) \\
&= \llbracket C \rrbracket^A_{bai}(P_1) \vee \llbracket C \rrbracket^A_{bai}(P_2)
\end{aligned}$$