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 choices and non deterministic iteration.

To keep the framework as general as possible the language is parametric on a set *Base* of base commands, common choices for the set of base commands usually include commands for variable assignement and boolean guards.

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

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

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

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

2.2 Semantics

Given a set $\mathbb{S}$ that represent the collection of all the possible states and family of partial functions $[\![b]\!]_{base} : \mathbb{S} \hookrightarrow \mathbb{S}$ we can define inductively the denotational semantics of $\mathbb{C}$ programs:

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

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

Clearly our definition is monotone

Theorem 1 ($[\![\cdot]\!]$ is monotone)

$$P \subseteq Q \implies [\![C]\!](P) \subseteq [\![C]\!](Q)$$

This framework is general enough to describe non deterministic imperative languages.

For example if we include a base command for boolean guards $e?$ whose semantics is to discard all the states that don't satisfy the assertion e, we can easily define the usual control flow statements:

- `if` b `then` C_1 `else` C_2 can be encoded as $(e? \fatsemi C_1) + (\neg e? \fatsemi C_2)$
- `while` e `do` C `done` can be encoded as $(e? \fatsemi C)^\star \fatsemi \neg e?$

3 Abstract Inductive Semantics

From the theory of abstract interpretation we know that we can be even more general and define the semantics of our programs on some complete lattice A, this definition is also parametric on a family of monotone functions $[\![b]\!]^A_{base} : A \to A \quad \forall\, b \in Base$ that describe the behaviour of the base commands.

Since in this context we aren't necessarily interested in approximating the denotational interpreter we will call from now on the following definition *Abstract Inductive Semantics.*

Definition 3 (Abstract inductive semantics)

$$
\begin{aligned}
[\![\cdot]\!]^A_{ais} &: A \to A \\
[\![\mathbb{1}]\!]^A_{ais} &= id_A \\
[\![b]\!]^A_{ais} &= \lambda P \to [\![b]\!]^A_{base}(P) \\
[\![C_1 \fatsemi C_2]\!]^A_{ais} &= [\![C_2]\!]^A_{ais} \circ [\![C_1]\!]^A_{ais} \\
[\![C_1 + C_2]\!]^A_{ais} &= \lambda P \to [\![C_1]\!]^A_{ais}(P) \vee_A [\![C_2]\!]^A_{ais}(P) \\
[\![C^\star]\!]^A_{ais} &= \lambda P \to lfp(\lambda P' \to P \vee_A [\![C]\!]^A_{ais} P')
\end{aligned}
$$

Clearly if the domain A is the lattice $\mathcal{P}(\mathbb{S})$ and we keep the semantics of the base commands as in Definition 2 we are giving describing the denotational semantics:

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

$$[\![C]\!]^{\mathcal{P}(\mathbb{S})}_{ais}(P) = [\![C]\!](P)$$

And the definition is still monotone with respect to the order on A:

Theorem 3 ($[\![\cdot]\!]^A_{ais}$ is monotone)

$$P \leq_A Q \implies [\![C]\!]^A_{ais}(P) \leq_A [\![C]\!]^A_{ais}(Q)$$

3.1 Galois connections

Given a Galois connection $\langle D, \leq_D \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leq_A \rangle$, if we have defined a *Abstract Inductive Semantics* on the domain D with the semantics of basic commands $[\![b]\!]^D_{ais}$, we can define an *Abstract Inductive Semantics* on the domain A with the semantics of basic commands $[\![b]\!]^A_{ais} = \alpha \circ [\![b]\!]^D_{ais} \circ \gamma$.

Theorem 4 (Soundness)

$$\alpha([\![C]\!]^D_{ais}(P)) \leq_D [\![C]\!]^A_{ais}(\alpha(P))$$

(Here, soundness is intended in an abstract interpretation sense)

All this results come from the theory of abstract interpretation.

4 Abstract Hoare logic

Definition 4 (Abstract Hoare triple) *Fixed a complete lattice A and the semantics of the base commands $[\![b]\!]^A_{base}$, 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:*

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

The definition is nonother that the definition of the standard Hoare triples ($\{P\} \; C \; \{Q\} \iff [\![C]\!](P) \subseteq Q$) but defined with respect to the *Abstract inductive semantics.*

4.1 Inference Rule

As in Hoare logic, we can provide a set of rules to derive valid triples compositionally.

Definition 5 (Abstract Hoare rules)

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

The identity command does not change the state, so if P holds before, it will hold after the execution.

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

For a basic command b, if P holds before the execution, then $[\![b]\!]^A_{base}(P)$ holds after the execution.

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

If executing C_1 from state P leads to state Q, and executing C_2 from state Q leads to state R, then executing C_1 followed by C_2 from state P leads to state R.

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

If executing either C_1 or C_2 from state P leads to state Q, then executing the nondeterministic choice $C_1 + C_2$ from state P also leads to state Q.

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

If executing command C from state P leads back to state P, then executing C repeatedly (zero or more times) from state P also leads back to state P.

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

If P is stronger than P' and Q' is stronger than Q, then we can derive $\langle P\rangle_A\ C\ \langle Q\rangle$ from $\langle P'\rangle_A\ C\ \langle Q'\rangle$.

All the rules follow the spirit of those in Hoare logic.
Clearly as in Hoare logic the proof system is sound:

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 $\vdash \langle P\rangle_A\ C\ \langle Q\rangle$:*

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

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

  The triple is valid since:

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

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

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

  The triple is valid since:

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

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

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

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

The triple is valid since:

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

- ($+$): *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: $[\![C_1]\!]^A_{ais}(P) \leq Q$ *and* $[\![C_2]\!]^A_{ais}(P) \leq Q$.

The triple is valid since:

$$\begin{aligned} [\![C_1 + C_2]\!]^A_{ais}(P) &= [\![C_1]\!]^A_{ais}(P) \vee [\![C_2]\!]^A_{ais}(P) && \textit{By definition of } [\![\cdot]\!]^A_{ais} \\ &\leq_A Q \vee Q \\ &= Q \end{aligned}$$

- ($\star$): *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: $[\![C]\!]^A_{ais}P \leq P$

$$[\![C^\star]\!]^A_{ais}(P) = lfp(\lambda P' \to P \vee_A [\![C]\!]^A_{ais}(P'))$$

$$\begin{aligned} (\lambda P' \to P \vee_A [\![C]\!]^A_{ais}(P'))(P) &= P \vee [\![C]\!]^A_{ais}(P) && \textit{since } [\![C]\!]^A_{ais}(P) \leq P \\ &= P \end{aligned}$$

Hence P *is a fixpoint for* $\lambda P' \to P \vee_A [\![C]\!]^A_{ais}(P')$

Thus $lfp(\lambda P' \to P \vee_A [\![C]\!]^A_{ais}(P')) \leq_A P$

- $(\leq)$: *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: $[\![C]\!]^A_{ais}(P') \leq Q'$.

$$\begin{aligned} [\![C]\!]^A_{ais}(P)[\![C]\!]^A_{ais}(P') & \qquad \textit{By monotonicity of } [\![\cdot]\!]^A_{ais} \\ \leq Q' & \qquad \textit{By inductive hypothesis} \\ \leq Q & \end{aligned}$$

And as Hoare logic is also relative complete in general.

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

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

Proof 2 *By structural induction on C:*

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

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

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

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

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

$$\frac{\begin{array}{c}\textit{(Inductive hypothesis)}\\ \vdash \langle P \rangle_A\ C_1\ \langle [\![C_1]\!]^A_{ais}(P) \rangle\end{array} \qquad \begin{array}{c}\textit{(Inductive hypothesis)}\\ \vdash \langle [\![C_1]\!]^A_{ais}(P) \rangle_A\ C_2\ \langle [\![C_2]\!]^A_{ais}([\![C_1]\!]^A_{ais}(P)) \rangle\end{array}}{\vdash \langle P \rangle_A\ C_1 \fatsemi C_2\ \langle [\![C_2]\!]^A_{ais}([\![C_1]\!]^A_{ais}(P)) \rangle}\ (\fatsemi)$$

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

$$\frac{\dfrac{P \leq P \qquad \begin{array}{c}\textit{(Inductive hypothesis)}\\ \vdash \langle P \rangle_A\ C_1\ \langle [\![C_1]\!]^A_{ais}(P) \rangle\end{array} \qquad [\![C_1]\!]^A_{ais}(P) \leq [\![C_1]\!]^A_{ais}(P) \vee [\![C_2]\!]^A_{ais}(P)}{\vdash \langle P \rangle_A\ C_1\ \langle [\![C_1]\!]^A_{ais}(P) \vee [\![C_2]\!]^A_{ais}(P) \rangle}\ (\leq) \qquad \pi_1}{\vdash \langle P \rangle_A\ C_1 + C_2\ \langle [\![C_1]\!]^A_{ais}(P) \vee [\![C_2]\!]^A_{ais}(P) \rangle}\ (+)$$

Where π_1:

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

- *$C^\star$: By definition $[\![C^\star]\!]^A_{ais}(P) = lfp(\lambda P' \to P \vee [\![C]\!]^A_{ais}(S')$ and let's call this value K, by K being a fixpoint the following fact is true $K = P \vee [\![C]\!]^A_{ais}(K)$ hence the following facts are true:*
  - *α_1: $K \ge P$*
  - *α_2: $K \ge [\![C]\!]^A_{ais}(K)$*

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

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 $\langle P \rangle_A\ C\ \langle Q \rangle \iff Q \ge [\![C]\!]^A_{ais}(P)$*

$$\dfrac{P \le P \qquad \overset{\textit{By Theorem 6}}{\vdash \langle P \rangle_A\ C\ \langle [\![C]\!]^A_{ais}(P) \rangle} \qquad Q \ge [\![C]\!]^A_{ais}(P)}{\vdash \langle P \rangle_A\ C\ \langle Q \rangle}\ (\le)$$

4.2 Instantiations of Abstract Hoare logic

In this chapter we will show that Abstract Hoare Logic is general enough that given a suitable domain is able to obtain the same judgements as other Hoare-Like logics

4.2.1 Hoare logic

As already shown in Theorem 2 when the domain is chosen to be $\mathcal{P}(\mathbb{S})$ the semantics corresponds to the denotational semantics of $\mathbb{C}$ hence Abstract Hoare logic instantiated to the $\mathcal{P}(\mathbb{S})$ domain gives us a sound and (relative) complete logic equivalent to standard Hoare logic.

4.2.2 Hyper Hoare logic

Hyper properties are used to express behaviour of some program with respect to all the possible executions, these property cannot be represented by some element of $\mathcal{P}(\mathbb{S})$ but they can be by elements of $\mathcal{P}(\mathcal{P}(\mathbb{S}))$.

Example 4.1 (Determinism) *To prove that a program C is deterministic (up to termination) using standard Hoare logic would require us to prove an infinite number of triples: $\forall P \in \mathcal{P}(\mathbb{S})$ such that $|P| = 1$ $\{P\}$ C $\{Q\}$ where $|Q| = 1$.*

Meaning that from the singleton collection of states that satisfy P executing C would reach another singleton collection of states Q.

This property could be easily be proved by a single triple $\{\{P \mid |P| = 1\}\}$ C $\{\{Q \mid |Q| = 1\}\}$ if we could pick as pre and post condition elements of $\mathcal{P}(\mathcal{P}(\mathbb{S}))$.

We will pick as the domain the following lattice:

Definition 6 (Hyper domain) *Fixed a domain B and some set K his hyper domain $H(B)_K$ is*

$$H(B)_K = K \to (B + undef)$$

The lattice on $B+undef$ is defined as the one on B but with $\emptyset > undef$, and the one on $K \to (B + undef)$ is the pointwise lift of the one on $(B + undef)$.

The semantics of the base commands instead is simply the pointwise lift of the semantics of base commands for $\mathcal{P}(B)$:

$$[\![b]\!]^{H(B)_K}_{base}(\chi) = \lambda r \to [\![b]\!]^{B}_{base}(\chi(r))$$

The abstract semantics defined by $H(B)_K$ is the pointwise lift of the one defined by B:

Theorem 8 (Hyper semantics is the pointwise lift of the base semantics)

$$[\![C]\!]^{H(B)_K}_{ais}(\chi) = \lambda r \to [\![C]\!]^{B}_{ais}(\chi(r))$$

Proof 4 *By structural induction on C:*

- $\mathbb{1}$*:*

$$\begin{aligned}[\![\mathbb{1}]\!]^{H(B)_K}_{ais}(\chi) &= \chi \\ &= \lambda r \to \chi(r) \\ &= \lambda r \to [\![\mathbb{1}]\!]^{B}_{ais}(\chi(r))\end{aligned}$$

- b*:*

$$[\![b]\!]^{H(B)_K}_{ais}(\chi) = \lambda r \to [\![b]\!]^{B}_{ais}(\chi(r))$$

- $C_1 \fatsemi C_2$:

$$\begin{aligned}
[\![C_1 \fatsemi C_2]\!]_{ais}^{H(B)_K}(\chi) &= [\![C_2]\!]_{ais}^{H(B)_K}([\![C_1]\!]_{ais}^{H(B)_K}(\chi)) \\
&= [\![C_2]\!]_{ais}^{H(B)_K}(\lambda r_1 \to [\![C_1]\!]_{ais}^{B}(\chi(r_1)) && \textit{By inductive hypothesis} \\
&= \lambda r_2 \to [\![C_2]\!]_{ais}^{B}(\lambda r_1 \to [\![C_1]\!]_{ais}^{B}(\chi(r_1))(r_2)) && \textit{By inductive hypothesis} \\
&= \lambda r_2 \to [\![C_2]\!]_{ais}^{B}([\![C_1]\!]_{ais}^{B}(\chi(r_2))) \\
&= \lambda r_2 \to [\![C_1 \fatsemi C_2]\!]_{ais}^{B}(\chi(r_2))
\end{aligned}$$

- $C_1 + C_2$:

$$\begin{aligned}
[\![C_1 + C_2]\!]_{ais}^{H(B)_K}(\chi) &= [\![C_1]\!]_{ais}^{H(B)_K}(\chi) \vee [\![C_2]\!]_{ais}^{H(B)_K}(\chi) \\
&= (\lambda r_1 \to [\![C_1]\!]_{ais}^{B}(\chi(r_1))) \vee (\lambda r_2 \to [\![C_1]\!]_{ais}^{B}(\chi(r_2))) && \textit{By inductive hypothesis} \\
&= \lambda r \to [\![C_1]\!]_{ais}^{B}(\chi(r)) \vee [\![C_2]\!]_{ais}^{B}(\chi(r)) \\
&= \lambda r \to [\![C_1 + C_2]\!]_{ais}^{B}(\chi(r))
\end{aligned}$$

- $C^\star$:

$$\begin{aligned}
[\![C^\star]\!]_{ais}^{H(B)_K}(\chi) &= lfp(\lambda\psi \to \chi \vee [\![C]\!]_{ais}^{H(B)_K}(\psi)) \\
&= lfp(\lambda\psi \to \chi \vee \lambda r \to [\![C]\!]_{ais}^{B}(\psi(r))) && \textit{By inductive hypothesis} \\
&= lfp(\lambda\psi \to \lambda r \to \chi(r) \vee [\![C]\!]_{ais}^{B}(\psi(r))) \\
&\textit{By } H(B)_K \textit{ being the pointwise lift of } B + undef \\
&\textit{his least fixpoint is the fixpoint of his components} \\
&= \lambda r \to lfp(\lambda P \to \chi(r) \vee [\![C]\!]_{ais}^{B} P) \\
&= \lambda r \to [\![C^\star]\!]_{ais}^{B}(\chi(r))
\end{aligned}$$

Definition 7 (Hyper instantiation) *Given a complete lattice B and a set K and his denotation on $\mathcal{P}(B)$ the instantiation of the hyper domain $H(\mathcal{P}(B))_K$ is an injective function $idx : \mathcal{P}(B) \to K$*

Given any hyper instantiation we can define:

$$\alpha(\chi) = \{\chi(k)\downarrow \mid k \in K\}$$

and:

$$\gamma(\mathcal{X}) = \lambda r \to \begin{cases} P & \exists P \in \mathcal{X} \ s.t.\ idx(P) = r \\ undef & otherwise \end{cases}$$

Theorem 9 (Idk)

$$\alpha([\![C]\!]_{ais}^{H(\mathcal{P}(B))_K}(\gamma(\mathcal{X}))) = \{[\![C]\!]_{ais}^{\mathcal{P}(B)}(P) \mid P \in \mathcal{X}\}$$

Proof 5

$$
\begin{aligned}
\alpha(\llbracket C \rrbracket_{ais}^{H(\mathcal{P}(B))_K}(\gamma(\mathcal{X}))) &= \alpha(\lambda r \to \llbracket C \rrbracket_{ais}^{\mathcal{P}(B))}(\gamma(\mathcal{X})(r))) && \textit{By theorem 8} \\
&= \{\llbracket C \rrbracket_{ais}^{\mathcal{P}(B))}(\gamma(\mathcal{X})(r)) \downarrow \mid k \in K\} && \textit{By the definition of } \alpha \\
&= \{\llbracket C \rrbracket_{ais}^{\mathcal{P}(B))}(P) \mid P \in \mathcal{X}\} && \textit{By the definition of } \gamma \textit{ and injectivity}
\end{aligned}
$$

Let $B = \mathbb{S}$ since $|\mathbb{S}| = |\mathbb{N}|$ there is a bijection $n : \mathbb{S} \to \mathbb{N}$ and since $|\mathcal{P}(\mathbb{N})| = |\mathbb{R}|$ there is another bijection $m : \mathcal{P}(\mathbb{N}) \to \mathbb{R}$, let the hyper instatiation for $H(\mathcal{P}(\mathbb{S}))_K$ be $\lambda r \to m(n(r))$.

Hence from theorem 9 $\llbracket C \rrbracket_{ais}^{H(\mathcal{P}(\mathbb{S}))_{\mathbb{R}}}(P)$ computer the strongest hyper post condition of program C from the hyper precondition P.

It follows that the abstract Hoare logic on the domain $H(\mathcal{P}(\mathbb{S}))_{\mathbb{R}}$ is a sound and complete proofsystem for deriving hyperproperties.

Example 4.2 (Determinism in abstract Hoare logic) *...*