Auxiliary Material for the Slides “Calculational Design of [In]Correctness Transformational Program Logics by Abstract Interpretation” at POPL 2024, London

PATRICK COUSOT

We study transformational program logics for correctness and incorrectness that we extend to explicitly handle both termination and nontermination. We show that the logics are abstract interpretations of the right image transformer for a natural relational semantics covering both finite and infinite executions. This understanding of logics as abstractions of a semantics facilitates their comparisons through their respective abstractions of the semantics (rather that the much more difficult comparison through their formal proof systems). More importantly, the formalization provides a calculational method for constructively designing the sound and complete formal proof system by abstraction of the semantics. As an example, we extend Hoare logic to cover all possible behaviors of nondeterministic programs and design a new precondition (in)correctness logic.

CCS Concepts: • Theory of computation → Logic and verification: Axiomatic semantics.

Additional Key Words and Phrases: program logic, transformer, semantics, correctness, incorrectness, termination, nontermination, abstract interpretation

ACM Reference Format:

This text contains the details of the formal development of Hoare logic, reverse Hoare logic aka incorrectness logic, and Hoare incorrectness logic.

Author’s address: Patrick Cousot, pcousot@cims.nyu.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2024 Copyright held by the owner/author(s).
ACM 2473-1421/2024/1-ART7
https://doi.org/10.1145/3632849
1 PROPERTIES OF STRONGEST POSTCONDITIONS

Lemma 1.1 (Composition). post(X \triangleright Y) = post(Y) \circ post(X).

Proof of Lem. 1.1.

\begin{align*}
post(X \triangleright Y) &= \lambda P \cdot \{\sigma'' | \exists \sigma \in P \cdot \langle \sigma, \sigma'' \rangle \in X \triangleright Y \} \\
&= \lambda P \cdot \{\sigma'' | \exists \sigma \in P \cdot \exists \sigma' \cdot \langle \sigma, \sigma' \rangle \in X \wedge \langle \sigma', \sigma'' \rangle \in Y \} \\
&= \lambda P \cdot \{\sigma'' | \exists \sigma' \cdot \sigma' \in \{\sigma' | \exists \sigma \in P \cdot \langle \sigma, \sigma' \rangle \in X \wedge \langle \sigma', \sigma'' \rangle \in Y \} \} \\
&= \lambda P \cdot \{\sigma'' | \exists \sigma' \in post(X)P \cdot \langle \sigma', \sigma'' \rangle \in Y \} \\
&= \lambda P \cdot post(Y)(post(X)P) \\
&= post(Y) \circ post(X) \quad \text{\textasciitilde (def. function composition \circ)} \quad \square
\end{align*}

Lemma 1.2 (Test). \(post[B]P = P \cap B[B]\).

Proof of Lem. 1.2.

\begin{align*}
post[B]P &= \{\sigma' | \exists \sigma \in P \cdot \langle \sigma, \sigma' \rangle \in [B] \} \\
&= \{\sigma | \sigma \in P \wedge \sigma \in B[B] \} \quad \text{\textasciitilde (def. \([B] \triangleq \{\{\sigma, \sigma \} \mid \sigma \in B[B] \}) \text{\textasciitilde}} \\
&= P \cap B[B] \quad \text{\textasciitilde (def. intersection \cup \text{\textasciitilde}} \\
&= \square
\end{align*}

Lemma 1.3 (Strongest postcondition). \(T(S) = a_G \circ \text{post}[S] = \{(P, \text{post}[S]P) \mid P \in \wp(\Sigma)\}\).

Proof of Lem. 1.3.

\begin{align*}
T(S) &= a_G \circ \text{post} \circ a_L \circ a_C([S]_L) \\
&= a_G \circ \text{post} \circ a_L([S]_L) \\
&= a_G \circ \text{post}([S]_L \cap (\Sigma \times \Sigma)) \\
&= a_G \circ \text{post}[S] \\
&= \{(P, \text{post}[S]P) \mid P \in \wp(\Sigma)\} \quad \text{\textasciitilde (def. of the angelic semantics \textasciitilde \text{\textasciitilde}} \\
&= \square
\end{align*}

Lemma 1.4 (Strongest postcondition over approximation).

\(T_{HL}(S) \triangleq \text{post}(\geq \subseteq) \circ T(S) = \{\langle P, Q \rangle \mid \text{post}[S]P \subseteq Q \} = \text{post}(\geq \subseteq) \circ T(S) \)

Proof of Lem. 1.4.

\begin{align*}
\text{post}(\geq \subseteq) \circ T(S) &= \text{post}(\geq \subseteq)(T(S)) \quad \text{\textasciitilde (def. function composition \circ)} \\
&= \text{post}(\geq \subseteq)\{(P, \text{post}[S]P) \mid P \in \wp(\Sigma)\} \quad \text{\textasciitilde (Lem. 1.3)} \\
&= \{(P', Q') \mid \exists \langle P, Q \rangle \in \{(P, \text{post}[S]P) \mid P \in \wp(\Sigma)\} \cdot \langle P, Q, (P', Q') \rangle \in \geq \subseteq \} \quad \text{\textasciitilde (def. \(10\) of post \text{\textasciitilde}} \\
&= \{(P', Q') \mid \exists P \cdot \langle P, \text{post}[S]P \rangle \geq \subseteq \langle P', Q' \rangle \} \quad \text{\textasciitilde (def. \text{\textasciitilde}} \\
&= \{(P', Q') \mid \exists P \cdot P \geq \subseteq \text{post}[S]P \subseteq Q' \} \quad \text{\textasciitilde (def. \text{\textasciitilde}} \\
&= \{(P', Q') \mid \exists P \cdot P' \in P \wedge \text{post}[S]P \subseteq Q' \} \quad \text{\textasciitilde (def. \text{\textasciitilde}}
= \{ q \in Q : \exists p \in P \text{ such that } \langle q, p \rangle \in R \}
\[
= \text{post}[-B] \circ \lambda P \cdot \text{lfp}^e \tilde{F}_P^e
\]
\[\{\text{pointwise commutation Lem. 1.6 and pointwise abstraction Cor. II.2.2}\} \square
\]

**Corollary 1.8 (Conditional iteration strongest postcondition graph).** \(\mathcal{T}(w) = \{(P, \text{post}[-B](\text{lfp}^e \tilde{F}_P^e)) \mid P \in \wp(\Sigma)\}\) where \(\tilde{F}_P^e(X) \triangleq P \cup \text{post}([B] ; [S]^e)X\).

**Proof of Cor. 1.8.**

\[
\mathcal{T}(w)
\]
\[= \alpha_G \circ \text{post}([w]) \quad \{\text{Lem. 1.3}\}
\]
\[= \alpha_G \circ \text{post}[-B] \circ \lambda P \cdot \text{lfp}^e \tilde{F}_P^e \quad \{\text{Th. 1.7}\}
\]
\[= \{\langle P, \text{post}[-B](\text{lfp}^e \tilde{F}_P^e) \rangle \mid P \in \wp(\Sigma)\}\] \[\{\text{def. (7) of } \alpha_G\} \square\]
2 CALCULATIONAL DESIGN OF HOARE LOGIC HL

2.1 Calculational Design of Hoare Logic Theory

**Theorem 2.1 (Theory of Hoare Logic HL).**

\[ T_{HL}(w) \triangleq \text{post}(\preceq, \subseteq) \circ T(w) \]

\[ = \{ (P, Q) \mid \exists I . P \subseteq I \land (I \cap \mathcal{B}[\mathcal{B}], I) \in T_{HL}(S) \land (I \cap \neg \mathcal{B}[\mathcal{B}]) \subseteq Q \} \]

**Proof of Th. 2.1.**

\[ T_{HL}(w) \]

\[ = \text{post}(\preceq, \subseteq) \circ T(w) \]

\[ = \{ (P', Q') \mid (P, Q) \in T(w) . (P, Q) = \subseteq (P', Q') \} \]

\[ = \{ (P', Q') \mid (P, Q) \in T(w) . P = P' \land Q \subseteq Q' \} \]

\[ = \{ (P, Q') \mid \exists Q . (P, Q) \in T(w) . Q \subseteq Q' \} \]

\[ = \{ (P, Q') \mid \exists Q . \text{post}[-\mathcal{B}](\text{lfp} \subseteq \bar{F}_{P}) \subseteq Q \land Q \subseteq Q' \} \]

\[ \{ (\subseteq) \exists Q . \text{post}[-\mathcal{B}](\text{lfp} \subseteq \bar{F}_{P}) \subseteq Q \land Q \subseteq Q' \text{ and transitivity; } \}

\[ (\subseteq) \text{ take } Q = Q' \}

\[ = \{ (P, Q') \mid \exists Q . \text{lfp} \subseteq \bar{F}_{P} \subseteq Q \land \text{post}[-\mathcal{B}](Q) \subseteq Q' \}

\[ \{ (\subseteq) \text{ take } Q = \text{lfp} \subseteq \bar{F}_{P} ; \ (\subseteq) \text{ post}[-\mathcal{B} \text{ is increasing by (12)} \}

\[ = \{ (P, Q') \mid \exists I . \bar{F}_{P}(I) \subseteq I \land \text{post}[-\mathcal{B}](I) \subseteq Q' \}

\[ \{ (\subseteq) \text{ I } Q \text{ implies post}[-\mathcal{B}](I) \subseteq \text{post}[-\mathcal{B}](Q) \text{ since post}[-\mathcal{B} \text{ is increasing by (12) hence post}[-\mathcal{B}](I) \subseteq Q' \text{ by transitivity; } \}

\[ (\subseteq) \text{ take } Q = I' \}

\[ \{ (P, Q) \mid \exists I . P \cup \text{post}([\mathcal{B}] ; [S]^\ast)(I) \subseteq I \land \text{post}[-\mathcal{B}](I) \subseteq Q \}

\[ \{ \text{ renaming, def. } \bar{F}_{P} \}

\[ = \{ (P, Q) \mid \exists I . P \cup \text{post}([\mathcal{B}] ; [S]) (I) \subseteq I \land \text{post}[-\mathcal{B}](I) \subseteq Q \}

\[ \{ [S] = [S] \text{ in absence of breaks} \}

\[ = \{ (P, Q) \mid \exists I . P \subseteq I \land \text{post}([\mathcal{B}] ; [S]) (I) \subseteq I \land \text{post}[-\mathcal{B}](I) \subseteq Q \}

\[ \{ \text{ def. } \subseteq \text{ and } \cup \}

\[ = \{ (P, Q) \mid \exists I . P \subseteq I \land \text{post}([\mathcal{B}] ; [S]) (I) \subseteq I \land \text{post}[-\mathcal{B}](I) \subseteq Q \}

\[ \{ \text{ composition Lem. 1.1} \}

\[ = \{ (P, Q) \mid \exists I . P \subseteq I \land \text{post}([\mathcal{B}] ; [S]) (I \cap \mathcal{B}[\mathcal{B}], I) \subseteq I \land (I \cap \mathcal{B}[\mathcal{B}]) \subseteq Q \}

\[ \{ \text{ test Lem. 1.2} \}

\[ = \{ (P, Q) \mid \exists I . P \subseteq I \land (I \cap \mathcal{B}[\mathcal{B}], I) \subseteq 

\[ \{ \text{ def. } \subseteq \}

\[ = \{ (P, Q) \mid \exists I . P \subseteq I \land (I \cap \mathcal{B}[\mathcal{B}], I) \subseteq (P, Q) \mid \text{post}(\preceq, \subseteq) \circ T(S) \land (I \cap \mathcal{B}[\mathcal{B}]) \subseteq Q \}

\[ \{ \text{ Lem. 1.4} \}

\[ = \{ (P, Q) \mid \exists I . P \subseteq I \land (I \cap \mathcal{B}[\mathcal{B}], I) \subseteq T_{HL}(S) \land (I \cap \mathcal{B}[\mathcal{B}]) \subseteq Q \}

\[ \{ \text{ Lem. 1.4} \} \]
2.2 Hoare logic rules

**Theorem 2.2** (Hoare rules for conditional iteration).

\[
P \subseteq I, \{I \cap B[B]\} S \{I\}, (I \cap \neg B[B]) \subseteq Q
\]

\[
\{P\} \text{while (B) } S \{Q\}
\]

\[(1)\]

**Proof of Th. 2.2.** We write \(\{P\} S \{Q\} \triangleq (P, Q) \in T_{HL}(S)\); By structural induction (\(S\) being a strict component of while (B) S), the rule for \(\{P\} S \{Q\}\) have already been defined;

By Aczel method, the (constant) fixpoint \(\text{lfp} \subseteq \lambda X \cdot S\) is defined by \(\{c \mid c \in S\}\);

So for while (B) S we have an axiom \(\{P\} \text{while (B) } S \{Q\}\) with side condition \(P \subseteq I, \{I \cap B[B]\} S \{I\}, (I \cap \neg B[B]) \subseteq Q\);

Traditionally, the side condition is written as a premiss, to get (1).
3 CALCULATIONAL DESIGN OF REVERSE HOARE AKA INCORRECTNESS LOGIC (IL)

3.1 Calculational Design of Reverse Hoare aka Incorrectness Logic Theory

Theorem 3.1 (Theory of IL)

\[ \mathcal{T}_{IL}(w) \triangleq \text{post}(\subseteq_\omega) \circ \mathcal{T}(w) \]

= \{ (P, Q) \mid Q \subseteq \text{post}[w]P \}

\[ \triangleq \text{order dual of Lem. 1.4} \]

= \{ (P, Q) \mid Q \subseteq \text{post}[-B]([\text{lf}p e \tilde{F}_P]) \}

\[ \text{Th. 1.7 where } \tilde{F}_P(X) \triangleq P \cup \text{post}([B] ; [S] e)X \]

= \{ (P, Q) \mid \exists I . Q \subseteq \text{post}([-B]) (I) \land I \subseteq \text{lf}p e \tilde{F}_P \}

\[ \text{def. } \tilde{F}_P \]

\[ \text{fixpoint underapproximation Th. II.3.6} \]

= \{ (P, Q) \mid \exists J^n . n < \omega \} . J^0 = \emptyset \land J^{n+1} \subseteq \tilde{F}_P(J^n) \land Q \subseteq \text{post}([-B]) (\bigcup_{n<\omega} J^n) \}

\[ \text{getting rid of } J^0 = \emptyset \]

= \{ (P, Q) \mid \exists J^n . n \in \mathbb{N} \} . J^0 = P \land J^{n+1} \subseteq \text{post}([B] ; [S] e)(J^n) \land Q \subseteq \text{post}([-B]) (\bigcup_{n\in\mathbb{N}} J^n) \}

\[ \text{changing } n \text{ to } n^+ \]

= \{ (P, Q) \mid \exists J^n . n \in \mathbb{N} \} . J^0 = P \land J^{n+1} \subseteq \text{post}([S] e)(J^n \land B[B]) \land Q \subseteq (\bigcup_{n} J^n) \land B[-B] \}

\[ \text{Lem. 1.2} \]

= \{ (P, Q) \mid \exists J^n . n \in \mathbb{N} \} . J^0 = P \land (J^n \land B[B] , J^{n+1}) \in \{ (P', Q') \mid Q' \subseteq \text{post}([S] e)P \} \land Q \subseteq (\bigcup_{n} J^n) \land B[-B] \}

\[ \text{def. } \epsilon \]

= \{ (P, Q) \mid \exists J^n . n \in \mathbb{N} \} . J^0 = P \land (J^n \land B[B] , J^{n+1}) \in \mathcal{T}_{IL}(S) \land Q \subseteq (\bigcup_{n} J^n) \land B[-B] \}

\[ \text{def. } \mathcal{T}_{IL} \]

\[ \square \]
3.2 Calculational design of IL rules

\[ J^0 = P, [J^n \cap B[B]] S [J^{n+1}], Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap B[-B] \]

\[ \quad \frac{}{[P] \text{while } (B) S [Q]} \] (2)

**Proof.** We write \([P] S [Q] \triangleq \{P, Q\} \in T_{IL}(S);

By structural induction (\(S\) being a strict component of \(\text{while } (B) S\)), the rule for \([P] S [Q]\) have already been defined;

By Aczel method, the (constant) fixpoint \(\text{lfp} \subseteq \lambda X::S\) is defined by \(\{\emptyset \mid c \in S\} ;

So for while \((B) S\) we have an axiom \(\{P\} \text{while } (B) S \{Q\}\) with side condition \(J^0 = P, [J^n \cap B[B]] S [J^{n+1}], Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap B[-B] ;

Traditionally, the side condition is written as a premiss, to get (2).
4 CALCULATIONAL DESIGN OF HOARE INCORRECTNESS LOGIC

4.1 Calculational Design of Hoare Incorrectness Logic Theory

Theorem 4.1 (Equivalent definitions of \( \mathcal{HL} \) theories).

\[ \mathcal{T}_{\mathcal{HL}}(w) \triangleq \text{post}(\leq, \geq) \circ \alpha^{-} \circ \mathcal{T}_{\mathcal{HL}}(w) \quad \text{W = while (B) S} \]

Observe that Th. 4.1 shows that \( \text{post}(\leq, \geq) \) can be dispensed with. This implies that the consequence rule is useless for Hoare incorrectness logic.

Proof of Th. 4.1.

\[ \mathcal{T}_{\mathcal{HL}}(w) = \text{post}(\leq, \geq) \circ \alpha^{-} \circ \mathcal{T}_{\mathcal{HL}}(w) \quad \text{[def. } \mathcal{T}_{\mathcal{HL}}]\]

\[ = \text{post}(\leq, \geq)(\{P, Q\} \mid \text{post}[w]P \subseteq Q) \quad \text{[Lem. 1.4 and def. (30) of } \alpha^{-}] \]

\[ = \text{post}(\leq, \geq)(\{P, Q\} \mid \neg(\text{post}[w]P \subseteq Q)) \quad \text{[def. } \neg \text{]} \]

\[ = \{P', Q'\} \mid \exists \{P, Q\} \in \{P, Q\} \mid \text{post}[w]P \cap \neg Q \cap \emptyset \}

\[ = \{P', Q'\} \mid \exists \{P, Q\} . \text{post}[w]P \cap \neg Q \cap \emptyset \cap \{P', Q'\} \quad \text{[def. } \emptyset \text{]} \]

\[ = \{P', Q'\} \mid \exists \{P, Q\} . \text{post}[w]P \cap \neg Q \cap \emptyset \cap P \subseteq P' \cap Q \supseteq Q' \}

\[ \text{[component wise def. of } \leq, \geq] \]

\[ = \{P', Q'\} \mid \exists \{P, Q\} . \text{post}[w]P \cap \neg Q \cap \emptyset \cap Q \supseteq Q' \]

\[ \text{[def. } \leq \text{ and } \neg \text{]} \]

Theorem 4.2 (Theory of \( \mathcal{HL} \)).

\[ \mathcal{T}_{\mathcal{HL}}(w) = \{\{P, Q\} \mid \forall n \geq 1 . \exists \{\sigma_i \in I, i \in [1, n]\} . \sigma_i \in P \land

\[ \forall i \in [1, n] . (B[B] \cap \{\sigma_i\} \cap \neg \{\sigma_{i+1}\}) \in \mathcal{T}_{\mathcal{HL}}(S) \land \sigma_n \notin B[B] \land S \notin Q} \]

Proof of Th. 4.2.

\[ \mathcal{T}_{\mathcal{HL}}(w) \]

\[ = \{\{P, Q\} \mid \text{post}[\neg B](\text{fpl} \cap \neg Q \cap \emptyset) \}

\[ \quad [\text{def. } \alpha^{-} \text{ and } \mathcal{T}_{\mathcal{HL}} \text{ for Hoare logic}] \]

\[ = \{\{P, Q\} \mid \text{fpl} \cap \neg B \cap \neg Q \cap \emptyset \}

\[ \quad [\text{induction principle Th. H.3}] \]

\[ = \{\{P, Q\} \mid \exists I \in \varphi(S) . \text{fpl}(I) \subseteq I \land \exists (W, \leq) \in \mathcal{WF} . \exists v \in I \rightarrow W . \exists \sigma_i \in I, i \in [1, \infty] . \sigma_i \in F_p(\sigma) \land \forall i \in [1, \infty] . \sigma_{i+1} \in F_p(\sigma_i) \land \forall i \in [1, \infty] . (\sigma_i \neq \sigma_{i+1}) \Rightarrow (v(\sigma_i) > v(\sigma_{i+1}) \land \forall i \in [1, \infty] . (v(\sigma_i) \neq v(\sigma_{i+1}) \Rightarrow \{\sigma_i \cap \text{pre}[\neg B](\neg Q) \neq 0} \]

\[ \quad \text{[def. } F_p(X) \triangleq P \cup \text{post}(\emptyset \mid \emptyset)X \subseteq, \text{and post, which is } \emptyset \text{-strict}] \]

Proc. ACM Program. Lang., Vol. 8, No. POPL, Article 7. Publication date: January 2024.
So for

TheoRem 4.3 ( Aczel method

\[\exists \sigma_i \in I, i \in \{1, n\} \land \sigma_i \in P \land \forall i \in \{1, n\} \not\neg (\{\langle B \rangle \land \{\langle \sigma_i \rangle \}) \mathcal{S} (\neg \{\sigma_i \}) \land \sigma_n \not\sigma B \land \sigma_n \not\sigma Q\]

\[\exists \{P, Q\} \mid (P) \mathcal{S} (Q) \not\exists \{P, Q\} \in \mathcal{H}(S);\]

By structural induction (S being a strict component of while (B) S), the rule for (P) S (Q) have already been defined;

By Aczel method, the (constant) fixpoint lfp ∈ \(\lambda X \cdot S\) is defined by \(\{c \mid c \in S\};\)

So for while (B) S we have an axiom \(\exists \{P, Q\} \mid (P) \mathcal{S} (Q) \not\exists \{P, Q\} \in \mathcal{H}(S);\)

\[\exists \{P, Q\} \mid (P) \mathcal{S} (Q) \not\exists \{P, Q\} \in \mathcal{H}(S);\]

Traditionally, the side condition is written as a premiss, to get (3).
This is nothing but debugging formalized as a logic since $\langle \sigma_i \in I, i \in [1,n] \rangle$ is a finite iteration in the loop starting with $P$ true and finishing with $Q$ false, which is obviously a counter example to Hoare triple $\{P\}$ while (B) $S\{Q\}$. Notice that recursively $\langle B[B] \cap \{\sigma_i\} \rangle S\langle \{\sigma_{i+1}\} \rangle$ enforces the execution of the loop body $S$ to start in state $\sigma_i$ and terminate in state $\sigma_{i+1}$. 

Proc. ACM Program. Lang., Vol. 8, No. POPL, Article 7. Publication date: January 2024.
5 COMPARISON OF INCORRECTNESS LOGIC AND HOARE INCORRECTNESS LOGIC

Lemma 5.1 (IL is sufficient but not necessary for incorrectness). Assuming \( Q \neq \Sigma \).
\[ \neg (\{P\} S\{Q\}) \iff \text{post}(R) P \land \neg Q \neq \emptyset \]  
\[ \iff \exists \sigma \in P . \exists \sigma' \notin Q . (\sigma, \sigma') \in [S] \]  
\[ \iff P \cap \text{pre}[S] \neg Q \neq \emptyset \]  
\[ \iff \forall \sigma' \notin Q . \exists \sigma \in P . (\sigma, \sigma') \in [S] \]  
\[ \iff [P] S[\neg Q] \]  

Proof of Lem. 5.1.
\[ \neg (\{P\} S\{Q\}) \iff \neg (\text{post}[S] P \subseteq Q) \]  
\[ \iff \text{post}[S] P \subseteq \neg Q \neq \emptyset \]  
\[ \iff \exists \sigma \in P . \exists \sigma' \notin Q . (\sigma, \sigma') \in [S] \]  
\[ \iff P \cap \text{pre}[S] \neg Q \neq \emptyset \]  
\[ \iff \forall \sigma' \notin Q . \exists \sigma \in P . (\sigma, \sigma') \in [S] \]  
\[ \iff [P] S[\neg Q] \]  
\[ \iff \text{reverse Hoare aka incorrectness logic} \]  
\[ \iff \text{def. triple} \]  
\[ \iff \text{def. post} \]  
\[ \iff \text{def. and \& \emptyset} \]  
\[ \iff \text{def. pre} \]  

(\( \iff \)) Assume \( \neg Q \neq \emptyset \) so pick \( \sigma_0 \in \neg Q \). Then, by hypothesis, \( \exists \sigma_1 \in P . (\sigma_0, \sigma_1) \in [S] \) proving \( \exists \sigma \in P . \exists \sigma' . (\sigma, \sigma') \in [S] \land \sigma' \notin Q \) with \( \sigma = \sigma_0 \) and \( \sigma' = \sigma_1 \);

(\( \not\iff \)) If \( \neg Q = \emptyset \) i.e. \( Q = \Sigma \) then \( \forall \sigma' \notin Q \). \( \exists \sigma \in P . (\sigma, \sigma') \in [S] \) is vacuously true while \( \exists \sigma' . \sigma' \notin Q \) hence \( \exists \sigma \in P . \exists \sigma' . (\sigma, \sigma') \in [S] \land \sigma' \notin Q \) is false. \( \square \)

Lemma 5.2 (Proving Hoare incorrectness with IL).
\[ \neg (\{P\} S\{Q\}) \iff \exists R \in \varphi(\Sigma) . [P] S[R] \land R \cap \neg Q \neq \emptyset \]  

Proof of Lem. 5.2.
\[ \neg (\{P\} S\{Q\}) \]  
\[ \iff \exists \sigma \in \Sigma . [P] S[\{\sigma\}] \land \sigma \notin Q \]  
\[ \iff \exists R \in \varphi(\Sigma) . [P] S[R] \land R \cap \neg Q \neq \emptyset \]  
\[ \iff \exists \sigma \in \Sigma . [P] S[\{\sigma\}] \land \sigma \notin Q \]  
\[ \iff \text{def. incorrect Hoare triple} \]  
\[ \iff \text{lem. 5.1} \]  
\[ \iff \text{commutativity and renaming} \]  
\[ \iff \text{def. } \exists \]  
\[ \iff \text{def. } \exists \]  
\[ \iff \text{def. IL} \]  
\[ \iff \text{def. IL} \]  
\[ \iff \text{def. IL} \]  
\[ \iff \text{def. IL} \]  

\((\in)\) take \( R = \{\sigma\} \);  
\((\equiv)\) since \( R \cap \neg Q \neq \emptyset \), we have \( \exists \sigma \in R . \sigma \notin Q \) and \( [P] S[\{\sigma\}] \) since otherwise we would have \( \neg (\forall \sigma'' \in \{\sigma\} . \exists \sigma' \in P . (\sigma', \sigma'') \in [S] ) \iff \forall \sigma' \in P . (\sigma, \sigma') \notin [S] ) \), in contradiction with \( [P] S[R] \) and \( \sigma \in R \). \( \square \)