Calculational Design of [In]Correctness Transformational Program Logics by Abstract Interpretation

Patrick Cousot
Courant Institute, New York University
Objective

Method to design program transformational logics

Transformational logic = Hoare style logics \{P\} S \{Q\}
Method to design a program transformational logics

1. Define the natural relational semantics \([S]_\perp\) of the programming language (in structural fixpoint form)
Method to design a program transformational logics

1. Define the natural relational semantics $\llbracket S \rrbracket_\perp$ of the programming language (in structural fixpoint form)

2. Define the theory of the logics as an abstraction $\alpha(\{\llbracket S \rrbracket_\perp\})$ of the collecting semantics $\{\llbracket S \rrbracket_\perp\}$ (strongest (hyper) property)

Theory of a logic = the subset of all true formulas
Method to design a program transformational logics

1. Define the **natural relational semantics** $⟦S⟧⊥$ of the programming language (in structural fixpoint form)

2. Define the **theory** of the logics as an abstraction $α(⟦S⟧⊥)$ of the collecting semantics $⟦S⟧⊥$ (strongest (hyper) property)

3. Calculate the theory $α(⟦S⟧⊥)$ in structural fixpoint form by **fixpoint abstraction**

*Theory of a logic = the subset of all true formulas*
Method to design a program transformational logics

1. Define the natural relational semantics $\llbracket S \rrbracket_\bot$ of the programming language (in structural fixpoint form)

2. Define the theory of the logics as an abstraction $\alpha(\llbracket S \rrbracket_\bot)$ of the collecting semantics $\{\llbracket S \rrbracket_\bot\}$ (strongest (hyper) property)

3. Calculate the theory $\alpha(\llbracket S \rrbracket_\bot)$ in structural fixpoint form by fixpoint abstraction

4. Calculate the proof system by fixpoint induction and Aczel correspondence between fixpoints and deductive systems

Theory of a logic = the subset of all true formulas
Two simple examples*: Hoare (HL) and reverse Hoare aka incorrectness (IL) logics

* not in the paper (where the examples are more complicated).
General Idea

\( \text{HL} = \text{strongest postcondition abstraction of the collecting semantics} \)

- over approximating consequence abstraction
- over approximating fixpoint induction
- Aczel correspondence fixpoint ⇔ proof system
General Idea

**HL** = strongest postcondition abstraction of the collecting semantics
+ over approximating consequence abstraction
+ over approximating fixpoint induction
+ Aczel correspondence fixpoint $\leftrightarrow$ proof system

**IL** = strongest postcondition abstraction of the collecting semantics
+ under approximating consequence abstraction
+ under approximating fixpoint induction
+ Aczel correspondence fixpoint $\leftrightarrow$ proof system
I. Angelic relational semantics $\llbracket S \rrbracket^e$

- **Syntax**: 
  
  $S \in \mathcal{S} ::= x = A \mid \text{skip} \mid S;S \mid \text{if} \ (B) \ S \ \text{else} \ S \mid \text{while} \ (B) \ S \mid x = [a, b] \mid \text{break}$

- **States**: $\Sigma$

- **Angelic relational semantics**: $\llbracket S \rrbracket^e \in \wp(\Sigma \times \Sigma)$

* plus unbounded nondeterminism, breaks, and nontermination $\bot$ in the paper.
1. Angelic relational semantics $[[S]]$ (in deductive form)

- Notations using judgements:
  - $\sigma \vdash S \Rightarrow^e \sigma'$ for $(\sigma, \sigma') \in [[S]]^e$
  - $\sigma \vdash \text{while}(B) \ S \Rightarrow^i \sigma'$ for $\sigma$ leads to $\sigma'$ after 0 or more iterations
1. Angelic relational semantics $[[S]]$ (in deductive form)

- Notations using judgements:
  - $\sigma \vdash S \Rightarrow^e \sigma'$ for $\langle \sigma, \sigma' \rangle \in [[S]]^e$
  - $\sigma \vdash \text{while}(B) S \Rightarrow \sigma'$ for $\sigma$ leads to $\sigma'$ after 0 or more iterations

- Semantics of the conditional iteration* $W = \text{while}(B) S$:

  \[ 
  \begin{align*}
  (a) \quad & \sigma \vdash W \overset{i}{\Rightarrow} \sigma \\
  (b) \quad & \frac{B[B] \sigma, \quad \sigma \vdash S \Rightarrow^e \sigma', \quad \sigma' \vdash W \overset{i}{\Rightarrow} \sigma''}{\sigma \vdash W \overset{i}{\Rightarrow} \sigma''} \\
  \end{align*} \]

  \[ 
  \begin{align*}
  (a) \quad & \frac{\sigma \vdash W \overset{i}{\Rightarrow} \sigma', \quad B[-B] \sigma'}{\sigma \vdash W \overset{e}{\Rightarrow} \sigma'} \\
  \end{align*} \]

* plus breaks, and co-induction for nontermination $\perp$ in the paper.
I. Angelic relational semantics $\llbracket S \rrbracket$ (in fixpoint form)

- **Semantics of the conditional iteration**\(^*\) $W = \text{while}(B) \ S$:

\[
F^e(X) \triangleq \text{id} \cup (\llbracket B \rrbracket \circ \llbracket S \rrbracket^e \circ X), \quad X \in \wp(\Sigma \times \Sigma)
\]

\[
\llbracket \text{while } (B) \ S \rrbracket^e \triangleq \text{lfp} \subseteq F^e \circ \llbracket \neg B \rrbracket
\]

- Derived using Aczel correspondence between deductive systems and set-theoretic fixpoints, see Ex. II.5.1
Aczel correspondence between deductive systems and fixpoints

- Rules: \( \frac{P}{c} \) (\( \mathcal{U} \) universe, \( P \in \mathcal{P}_{\text{fin}}(\mathcal{U}) \) premiss, \( c \in \mathcal{U} \) conclusion, \( \emptyset \) axiom)
Aczel correspondence between deductive systems and fixpoints

- Rules: \( \frac{P}{c} \) (\( U \) universe, \( P \in \wp_{\text{fin}}(U) \) premiss, \( c \in U \) conclusion, \( \emptyset \) axiom)

- Deductive system: \( R = \{ \frac{P_i}{c_i} \mid i \in \Delta \} \), \( R \in \wp(\wp_{\text{fin}}(U) \times U) \)
Aczel correspondence between deductive systems and fixpoints

- Rules: $\frac{P}{c}$ (universal, $P \in \wp_{\text{fin}}(U)$ premiss, $c \in U$ conclusion, $\emptyset$ axiom)

- Deductive system: $R = \{ \frac{P_i}{c_i} \mid i \in \Delta \}$, $R \in \wp(\wp_{\text{fin}}(U) \times U)$

- Subset of the universe $U$ defined by $R$:

  \[ F(R)X \triangleq \{ c \mid \exists \frac{P}{c} \in R . P \subseteq X \} \]

  - proof theoretic $\downarrow$

  \[ \ellfp \subseteq F(R) \]

  - model theoretic (gfp for coinduction)

  \[ \{ t_n \in U \mid \exists t_1, \ldots, t_{n-1} \in U . \forall k \in [1, n] . \exists \frac{P}{c} \in R . P \subseteq \{ t_1, \ldots, t_{k-1} \} \land t_k = c \} \]

  - consequence operator
Aczel correspondence between deductive systems and fixpoints

- Rules: $\frac{P}{c}$ (U universe, $P \in \wp_{\text{fin}}(U)$ premiss, $c \in U$ conclusion, $\emptyset$ axiom)

- Deductive system: $R = \left\{ \frac{P_i}{c_i} \mid i \in \Delta \right\}$, $R \in \wp(\wp_{\text{fin}}(U) \times U)$

- Subset of the universe $U$ defined by $R$:
  \[
  \{ t_n \in U \mid \exists t_1, \ldots, t_{n-1} \in U. \forall k \in [1, n]. \exists \frac{P}{c} \in R. P \subseteq \{ t_1, \ldots, t_{k-1} \} \land t_k = c \}
  \]
  \[
  \text{proof theoretic} \downarrow
  \]
  \[
  \text{lfp} \subseteq F(R)
  \]
  \[
  F(R)X \supseteq \left\{ c \mid \exists \frac{P}{c} \in R. P \subseteq X \right\}
  \]
  \[
  \text{model theoretic (gfp for coinduction)} \leftarrow
  \]
  \[
  \text{consequence operator} \leftarrow
  \]

- Deductive system defining $\text{lfp} \subseteq F$:
  \[
  R_F \supseteq \left\{ \frac{P}{c} \mid P \subseteq U \land c \in F(P) \right\}
  \]
2. Abstraction (much simplified)

- The composition of these abstractions is

- This is an oversimplification of Fig. 1 of the paper, forgetting about nontermination including total correctness and relational predicates.
2. Abstraction (much simplified)

• Hyper properties to properties abstraction:

\[
\langle \varphi(\varphi(\Sigma \times \Sigma)), \subseteq \rangle \xleftarrow{\gamma_C} \langle \varphi(\Sigma \times \Sigma), \subseteq \rangle \\
\alpha_C(P) = \bigcup P \\
\gamma_C(S) = \varphi(S)
\]
2. Abstraction (much simplified)

- Hyper properties to properties abstraction:

\[
\langle \varnothing (\varnothing (\Sigma \times \Sigma)), \subseteq \rangle \xrightarrow{\gamma_C \alpha_C} \langle \varnothing (\Sigma \times \Sigma), \subseteq \rangle \quad \alpha_C(P) \doteq \bigcup P \quad \gamma_C(S) \doteq \varnothing (S)
\]

- Post-image isomorphism:

\[
\langle \varnothing (\Sigma \times \Sigma), \subseteq \rangle \xleftarrow{\text{pre}} \langle \varnothing (\Sigma) \rightarrow \varnothing (\Sigma), \subseteq \rangle \quad \text{post}(R) \doteq \lambda P \cdot \{ \sigma' \mid \exists \sigma \in P \land (\sigma, \sigma') \in R \}
\]

\[
\text{pre}(R) \doteq \lambda X \cdot \{ \sigma \mid \forall \sigma' \in Q \cdot (\sigma, \sigma') \in R \}
\]
2. Abstraction (much simplified)

- Hyper properties to properties abstraction:
  \[ \langle \mathcal{P}(\Sigma \times \Sigma) \rangle \subseteq \xhookrightarrow{\gamma_C/\alpha_C} \langle \mathcal{P}(\Sigma \times \Sigma) \rangle, \subseteq \]

\[ \alpha_C(P) \doteq \bigcup P \]

\[ \gamma_C(S) \doteq \mathcal{P}(S) \]

- Post-image isomorphism:
  \[ \langle \mathcal{P}(\Sigma \times \Sigma) \rangle \subseteq \xhookrightarrow{\text{post}} \langle \mathcal{P}(\Sigma) \rangle \rightarrow \langle \mathcal{P}(\Sigma) \rangle, \subseteq \]

\[ \text{post}(R) \doteq \lambda P \cdot \{ \sigma' \mid \exists \sigma \in P \wedge \langle \sigma, \sigma' \rangle \in R \} \]

\[ \text{pre}(R) \doteq \lambda X \cdot \{ \sigma \mid \forall \sigma' \in Q. \langle \sigma, \sigma' \rangle \in R \} \]

- Graph isomorphism (a function is isomorphic to its graph, which is a function relation):.../

\[ \langle \mathcal{P}(\Sigma) \rightarrow \mathcal{P}(\Sigma), = \rangle \xhookrightarrow{\gamma_G/\alpha_G} \langle \mathcal{P}_{\text{fun}}(\mathcal{P}(\Sigma \times \mathcal{P}(\Sigma))), = \rangle \]

\[ f \in \mathcal{P}(\Sigma) \rightarrow \mathcal{P}(\Sigma) \]

\[ \alpha_G(f) = \{ (P, f(P)) \mid P \in \mathcal{P}(\Sigma) \} \]

\[ \gamma_G(R) \doteq \lambda P \cdot (Q \text{ such that } \langle P, S \rangle \in R) \]
2. Abstraction (much simplified)

- **Strongest postcondition logic theory** (common to HL and IL with no consequence rule):

\[
\mathcal{T}(S) \triangleq \alpha_G \circ \text{post} \circ \alpha_C(\{\llbracket S \rrbracket\})
\]

\[
= \{ \langle P, \text{post}[S]P \rangle \mid P \in \wp(\Sigma) \}
\]
2. Abstraction (much simplified)

• Strongest postcondition logic theory (common to HL and IL with no consequence rule):

\[
\mathcal{T}(S) \triangleq \alpha_G \circ \text{post} \circ \alpha_C(\{\llbracket S \rrbracket\}) = \{\langle P, \text{post}\llbracket S \rrbracket P \rangle \mid P \in \wp(\Sigma)\}
\]

• Notation: \( \{P\} S \{Q\} \triangleq \langle P, Q \rangle \in \mathcal{T}(S) \)

• The next step is to express this theory in fixpoint form
2. Abstraction (much simplified)

- The abstraction of a fixpoint is a fixpoint (POPL 79)

**Theorem II.2.1 (Fixpoint abstraction).** If \( \langle C, \sqsubseteq \rangle \xleftarrow{\mathit{i}} \langle A, \leq \rangle \) is a Galois connection between complete lattices \( \langle C, \sqsubseteq \rangle \) and \( \langle A, \leq \rangle \), \( f \in C \xrightarrow{i} C \) and \( \bar{f} \in A \xrightarrow{i} A \) are increasing and commuting, that is, \( \alpha \circ f = \bar{f} \circ \alpha \), then \( \alpha(\text{lfp}^\sqsubseteq f) = \text{lfp}^{\leq} \bar{f} \) (while semi-commutation \( \alpha \circ f \leq \bar{f} \circ \alpha \) implies \( \alpha(\text{lfp}^\sqsubseteq f) \leq \text{lfp}^{\leq} \bar{f} \)).
2. Abstraction (much simplified)

• The abstraction of a fixpoint is a fixpoint (POPL 79)

  Theorem II.2.1 (Fixpoint abstraction). If $\langle C, \sqsubseteq \rangle \overset{\text{fixpoint}}{\xrightarrow{\alpha}} \langle A, \preceq \rangle$ is a Galois connection between complete lattices $\langle C, \sqsubseteq \rangle$ and $\langle A, \preceq \rangle$, $f \in C \overset{i}{\to} C$ and $\tilde{f} \in A \overset{i}{\to} A$ are increasing and commuting, that is, $\alpha \circ f = \tilde{f} \circ \alpha$, then $\alpha(\lfp^\sqsubseteq f) = \lfp^\preceq \tilde{f}$ (while semi-commutation $\alpha \circ f \preceq \tilde{f} \circ \alpha$ implies $\alpha(\lfp^\sqsubseteq f) \preceq \lfp^\preceq \tilde{f}$).

• We get a fixpoint definition of the theory of strongest postconditions logics (common to HL and IL with no consequences at all)

• For the iteration $W = \text{while } (B) S :$

  $$\mathcal{T}(W) \triangleq \{ \langle P, \post\neg B (\lfp^\sqsubseteq \lambda X \cdot P \cup \post(\llbracket B \rrbracket ; \llbracket S \rrbracket^e X)) \rangle \mid P \in \wp(\Sigma) \}$$
Lemma 1.2 (test).

\[
\begin{align*}
\text{def. post} & \quad \{ \text{def. } post \} \\
\text{def. } post & \quad \{ \text{def. } post \} \\
\text{def. } post & \quad \{ \text{def. } post \} \\
\text{def. } post & \quad \{ \text{def. } post \} \\
\end{align*}
\]

For simplicity, we consider conditional iteration \( \text{while } (B) \) with no break.

**Lemma 1.5 (Construction)**

**Proof of Lem. 1.5.**

**Lemma 1.6 (Pointwise Computation).**

**Proof of Lem. 1.6.**

**Theorem 1.7 (Iteration Strongest Postcondition).**

**Proof of Th. 1.7.**
3. Approximation

- The component wise approximation:

\[ \langle x, y \rangle \sqsubseteq, \sqsubseteq \langle x', y' \rangle \iff x \sqsubseteq x' \land y \leq y' \]
3. Approximation

• The component wise approximation:

\[ \langle x, y \rangle \subseteq, \leq \langle x', y' \rangle \triangleq x \subseteq x' \land y \leq y' \]

• The over approximation abstraction for HL:

\[
\text{post}(\subseteq, \supseteq) = \lambda R \cdot \{ \langle P, Q \rangle \mid \exists \langle P', Q' \rangle \in R . \ P \subseteq P' \land Q' \subseteq Q \}
\]

\[
\mathcal{T}_{\text{HL}}(S) \triangleq \text{post}(\supseteq \subseteq) \circ \mathcal{T}(S)
\]
3. Approximation

- The component wise approximation:
  \[ \langle x, y \rangle \subseteq, \leq \langle x', y' \rangle \triangleq x \subseteq x' \land y \leq y' \]

- The **over** approximation abstraction for HL:
  \[ \text{post}(\subseteq, \supseteq) = \lambda R \cdot \{ \langle P, Q \rangle \mid \exists \langle P', Q' \rangle \in R . P \subseteq P' \land Q' \subseteq Q \} \]
  \[ \mathcal{T}_{\text{HL}}(S) \triangleq \text{post}(\supseteq, \subseteq) \circ \mathcal{T}(S) \]

- The (order dual) **under** approximation abstraction for IL:
  \[ \text{post}(\supseteq, \subseteq) = \lambda R \cdot \{ \langle P, Q \rangle \mid \exists \langle P', Q' \rangle \in R . P' \subseteq P \land Q \subseteq Q' \} \]
  \[ \mathcal{T}_{\text{IL}}(S) \triangleq \text{post}(\subseteq, \supseteq) \circ \mathcal{T}(S) \]

- Shows what it shared by HL and IL: all but the consequence rule (?)
4. Fixpoint induction

- Deriving the proof system at this stage by Aczel correspondence would be great!
- A common part and different consequence rules for HL and IL
4. Fixpoint induction

• Deriving the proof system at this stage by Aczel correspondence would be great!

• A common part and different consequence rules for HL and IL

• But then the HL proof system for iteration would be
  1. Prove strongest postconditions (⇒⇒⇒⇒⇒⇒ total correctness)
  2. Approximate with a consequence rule to get partial correctness

• This is sound and complete
4. Fixpoint induction

- Deriving the proof system at this stage by Aczel correspondence would be great!

- A common part and different consequence rules for HL and IL

- But then the HL proof system for iteration would be
  1. Prove strongest postconditions (total correctness)
  2. Approximate with a consequence rule to get partial correctness

- This is sound and complete

- But too demanding $\Rightarrow$ not so great!

- What we miss is fixpoint induction
4. Fixpoint induction

**Theorem II.3.1 (Park Fixpoint over Approximation)**

Let \( \langle L, \subseteq, \bot, \top, \cup, \cap \rangle \) be a complete lattice, \( f \in L \xrightarrow{i} L \) be increasing, and \( p \in L \). Then \( \text{lfp}^E f \subseteq p \) if and only if \( \exists i \in L. f(i) \subseteq i \land i \subseteq p \).
4. Fixpoint induction

**Theorem II.3.6 (Fixpoint Under Approximation by Transfinite Iterates)**

Let \( f \in L \xrightarrow{i} L \) be an increasing function on a cpo \( \langle L, \sqsubseteq, \bot, \sqcup \rangle \). \( P \sqsubseteq \text{lfp}^L f \), if and only if there exists an increasing transfinite sequence \( \langle X^\delta, \delta \in \mathbb{O} \rangle \) such that

1. \( X^0 = \bot \),
2. \( X^{\delta+1} \subseteq f(X^\delta) \) for successor ordinals,
3. \( \bigsqcup_{\delta < \lambda} X^\delta \) exists for limit ordinals \( \lambda \) such that \( X^\lambda \sqsubseteq \bigsqcup_{\delta < \lambda} X^\delta \), and
4. \( \exists \delta \in \mathbb{O} . \ P \sqsubseteq X^\delta \).

\( \delta \) bounded by \( \omega \) for continuous \( f \).
5. Calculational design of HL

- **Theory of HL** (for iteration):

  \[ T_{HL}(w) \triangleq \text{post}(\supseteq \subseteq) \circ T(w) \]
  
  \[ = \{ \langle P, Q \rangle \mid \exists I . P \subseteq I \land \langle I \cap B[B], I \rangle \in T_{HL}(S) \land (I \cap \neg B[B]) \subseteq Q \} \]
5. Calculational design of HL

- **Theory of HL** (for iteration):

\[ T_{HL}(W) \triangleq \text{post}(\supseteq \subseteq) \circ T(W) \]

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

- **HL proof system**:

**Theorem 3 (Hoare rules for conditional iteration).**

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

\[ \{ P \} \text{ while } (B) \text{ S } \{ Q \} \]
2 CALCULATIONAL DESIGN OF HOARE LOGIC HL

2.1 Calculational Design of Hoare Logic Theory

Theorem 2.1 (Theory of Hoare Logic HL).

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

\[ \{ (P, Q) | \exists I. P \subseteq I \land (I \cap B[\bar{b}]), I \in T_{HL}(s) \land (I \cap \sim B[\bar{b}]) \subseteq Q \} \]

Proof of Th. 2.1.

\[ T_{HL}(w) \]

= \text{post}(\preceq) \circ T(w) \quad \{ \text{def. } T_{HL} \} \]

= \text{post}(\preceq) \circ T(w) \quad \{ \text{Lem. 1.4} \} \]

= \{ (P', Q') | (P, Q) \in T(w) \} \quad \{ \text{def. post} \} \]

= \{ (P', Q') | (P, Q) \in T(w) \land (P' \cap Q' \subseteq Q') \} \quad \{ \text{component wise def. } \preceq \} \]

= \{ (P', Q') \mid (P, Q) \in T(w) \land Q \subseteq Q' \} \quad \{ \text{Lem. 1.4} \} \]

= \{ (P', Q') \mid (P, Q) \in T(w) \land Q \subseteq Q' \} \quad \{ \text{Th. 1.7} \} \]

2.2 Hoare logic rules

Theorem 2.2 (Hoare rules for conditional iteration).

\[ P \subseteq I, \{ I \cap B[\bar{b}] \} \subseteq \{ I \} \land \{ I \cap \sim B[\bar{b}] \} \subseteq Q \]

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

Proof of Th. 2.2. We write \{ P \} S \{ Q \} = \{ P, Q \} \subseteq 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 \textbf{Aczel method}, the (constant) fixpoint \text{lfp} \lambda X . S is defined by \{ \frac{\varnothing}{\varnothing} | e \in S \};

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

Traditionally, the side condition is written as a premiss, to get (1).

Sound and complete by construction

Machine checkable, if not machine checked!
Surprised to find a variant of HL proof system

We also have (post is increasing):

\[ \mathcal{T}_{\text{HL}}(S) = \text{post}(=, \subseteq) \circ \mathcal{T}(S) \]

yields the sound and complete proof system:

**\( \subseteq \) comes from** Th. II.3.1

\[ P \subseteq I, \quad \{I \cap B[B]\} \subseteq \{I\} \]

\[ \{P\} \text{ while } (B) \subseteq \{I \cap \neg B[B]\} \]

\[ \{P\} \subseteq \{Q\}, \quad Q \subseteq Q' \]

\[ \{P\} \subseteq \{Q'\} \]
Surprised to find a variant of HL proof system

We also have (post is increasing):

\[ \mathcal{T}_{\text{HL}}(s) = \text{post}(=, \subseteq) \circ \mathcal{T}(s) \]

yields the sound and complete proof system:

- \( \subseteq \) comes from Th. II.3.1
- \( P \subseteq I, \quad \{I \cap B[B]\} \subseteq \{I\} \)
- \( \{P\} \text{ while (B) } S \{I \cap \neg B[B]\} \)
- \( \{P\} \text{ s } \{Q\}, \quad Q \subseteq Q' \)
- \( \{P\} \text{ s } \{Q'\} \)

no need for Hoare left consequence rule (but for iteration):

If \(-P\{Q\}R \text{ and } -S \supseteq P \text{ then } -S\{Q\}R\)
5. Calculational design of IL

• Theory of IL (for iteration):

$$T_{IL}(W) \triangleq \text{post}(\subseteq \supseteq) \circ T(W)$$

$$= \{ \langle P, Q \rangle | \exists (J^n, n \in \mathbb{N}) . J^0 = P \wedge J^n \cap B[B], J^{n+1} \in T_{IL}(S) \wedge Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap B[B] \}$$
5. Calculational design of IL

- **Theory of IL (for iteration):**

  \[ T_{\text{IL}}(w) \triangleq \text{post}(\subseteq \triangleright) \circ T(w) \]
  \[ = \{ (P, Q) \mid \exists (J^n, n \in \mathbb{N}) \cdot J^0 = P \land (J^n \cap B[B], J^{n+1}) \in T_{\text{IL}}(S) \land Q \subseteq (\bigcup_{n \in \mathbb{N}} J^n) \cap B[-B] \} \]

- **IL proof system:**

  **Theorem 5 (IL rules for conditional iteration).**

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

  \[
  \hline
  \]

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

(similar to O’Hearn backward variant since the consequence rule can also be separated)
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).**

Let \( \mathcal{T}_L(w) \) be the set of all \( w \)-terms.

**Proof of Th. 3.1.**

\[ \mathcal{T}_L(w) = \{ (P, Q) \mid \exists \text{n} \in \mathbb{N} . \quad J^n = P \land J^n \cap B[8] \mid \mathcal{Q} \subseteq (\bigcup_{n} J^n) \cap B[-8] \} \]

**3.2  Calculational design of IL rules**

\[
J^n = P, \quad [J^n \cap B[8]] \mathcal{S} [J^{n+1}] \quad \mathcal{Q} \subseteq (\bigcup_{n} J^n) \cap B[-8]
\]

**Proof.** We write \([P] \mathcal{S} [Q] = (P, Q) \in \mathcal{T}_L(s)\).

By structural induction \( \mathcal{S} \) being a strict component of \( \mathcal{S} \) and \( B[8] \) and \( B[-8] \), the rule for \([P] \mathcal{S} [Q] \) have already been defined.

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

So for while \( B \), we have an axiom \( \{ P \} \mathcal{W} (B) \mathcal{S} [Q] \) with side condition \( J^n = P, [J^n \cap B[8]] \mathcal{S} [J^{n+1}] \quad \mathcal{Q} \subseteq (\bigcup_{n} J^n) \cap B[-8] \).

Traditionally, the side condition is written as a premiss, to get (2).
Much more in the paper
Much more in the paper

- Bi-inductive relational semantics with break and non termination (⊥), for termination and nontermination proofs
Much more in the paper

- Bi-inductive relational semantics with break and non-termination (⊥), for termination and non-termination proofs
- Many more abstractions and combinations → hundreds of transformational logics theories (including property negations, proofs by contradictions, backward logics, etc.)

Fig. 3. Taxonomy of assertional logics
Much more in the paper

- Bi-inductive relational semantics with `break` and non termination (⊥), for termination and nontermination proofs
- Many more abstractions and combinations → hundreds of transformational logics theories (including property negations, proofs by contradictions, backward logics, etc.)
- Taxonomies based on theory abstractions (not proof systems)

Fig. 3. Taxonomy of assertional logics
Much more in the paper

• Many more fixpoint induction principles (including $P \subseteq \text{lfp} \subseteq F$, $\text{lfp} \subseteq F \subseteq P$, $P \subseteq \text{gfp} \subseteq F$, $\text{gfp} \subseteq F \subseteq P$, $\text{lfp} \subseteq F \cap P \neq \emptyset$, $\text{gfp} \subseteq F \cap P \neq \emptyset$, etc)
Much more in the paper

• Example I: calculational design of a logic for partial correctness + total correctness + non termination

\{ n = n \land f = 1 \}

while (n!=0) { f = f \ast n; n = n - 1;}

\{ (n \geq 0 \land f =!\underline{n}) \lor (n < 0 \land n = f = \bot) \}
Much more in the paper

- Example II: calculational design of an incorrectness logic including non-termination
• Example II: calculational design of an incorrectness logic including non-termination

• A specification for factorial:

\[
\{ n \in [-\infty, \infty] \land f \in [1,1] \}
\]

while \( (n!=0) \) \{ \( f = f \times n; \ n = n - 1; \) \}
\[
\{ f \in [1,\infty] \}
\]

• False alarm \( f \in [-\infty,0] \) with a (totally imprecise) interval analysis
Much more in the paper

• Example II: calculational design of an incorrectness logic including non
  termination

• A specification for factorial:

\[
\begin{align*}
\{ & n \in [-\infty, \infty] \land f \in [1, 1] \\
\text{while } (n!=0) \{ & f = f \times n; n = n - 1; \\
\{ & f \in [1, \infty] \}
\end{align*}
\]

• False alarm \( f \in [-\infty, 0] \) with a (totally imprecise) interval analysis

• The alarm is false by nontermination, not provable with IL
About incorrectness

- IL is not Hoare incorrectness logic (sufficient, not necessary)

\[-(\{P\} S\{Q\}) \not\implies [P] S [\neg Q]\]

\[\iff \exists R \in \varnothing(\Sigma). [P] S [R] \land R \cap \neg Q \neq \emptyset\]

\[\iff \exists \sigma \in \Sigma. [P] S [\{\sigma\}] \land \sigma \notin Q\]

- The logic \(\mathcal{T}_{HL}(w) \triangleq \text{post}(\subseteq, \supseteq) \circ \alpha^{-}\circ \mathcal{T}_{HL}(w)\) can be calculated by the design method (and does not need a consequence rule)
4 CALCULATIONAL DESIGN OF HOARE INCORRECTNESS LOGIC

4.1 Calculational Design of Hoare Incorrectness Logic Theory

Theorem 4.1 (Equivalence definitions of \( \text{HIL} \) theories).

\[
\text{THIL}(w) = \text{post}(\langle P, Q \rangle \wedge \text{post} [w] P \subseteq Q) = \langle a \wedge S \rangle \text{THIL}(w) \quad \text{w: while Bl (S)}
\]

Observe that Th. 4.1 shows that \( \text{post}(\langle a \wedge S \rangle \) can be dispensed with. This implies that the consequence rule is useless for Hoare incorrectness logic.

Proof of Th. 4.1.

\[
\begin{align*}
\text{THIL}(w) &= \text{post}(\langle P, Q \rangle \wedge \text{post} [w] P \subseteq Q) & \text{(def. \( \text{THIL} \))} \\
&= \text{post}(\langle P, Q \rangle \wedge \text{post} [w] P \subseteq Q) \quad \text{(Lemma 1.4 and def. (10) of \( a \wedge S \))} \\
&= \text{post}(\langle P, Q \rangle \wedge \text{post} [w] P \subseteq Q) \quad \text{(def. \( \text{post} \))} \\
&= \{ \langle P, Q \rangle \wedge \text{post} [w] P \subseteq Q \} \quad \text{component wise def. of \( \subseteq \)} \\
&= \{ \langle P, Q \rangle \wedge \text{post} [w] P \subseteq Q \} \quad \text{(because \( P \subseteq Q \) then \( \text{post} [w] P \subseteq \text{post} [w] Q \))} \\
&= \text{post}(\langle P, Q \rangle \wedge \text{post} [w] P \subseteq Q) \quad \text{(def. \( a \wedge S \) for \( \text{HIL} \) logic).}
\end{align*}
\]

Theorem 4.2 (Theory of \( \text{HIL} \)).

\[
\text{THIL}(w) = \{ \langle P, Q \rangle \mid \exists \{ a \wedge S \} \subseteq \text{HIL}(w) \wedge \forall i \in \{ a \wedge S \} \}
\]

Proof of Th. 4.2.

\[
\begin{align*}
\text{THIL}(w) &= \{ \langle P, Q \rangle \mid \exists \{ a \wedge S \} \subseteq \text{HIL}(w) \wedge \forall i \in \{ a \wedge S \} \} & \text{Lemma 1.3, where \( \text{HIL}(w) \subseteq \text{HIL}(w) \subseteq \text{HIL}(w) \subseteq X \subseteq Y \subseteq Z \subseteq Y \subseteq X \)} \\
&= \{ \langle P, Q \rangle \mid \exists \{ a \wedge S \} \subseteq \text{HIL}(w) \wedge \forall i \in \{ a \wedge S \} \} & \text{(19) Bl)} \\
&= \{ \langle P, Q \rangle \mid \exists \{ a \wedge S \} \subseteq \text{HIL}(w) \wedge \forall i \in \{ a \wedge S \} \} & \text{(Def. Bl (S))} \\
&= \text{THIL}(w) \quad \text{def. of \( \{ a \wedge S \} \subseteq \text{HIL}(w) \wedge \forall i \in \{ a \wedge S \} \)}
\end{align*}
\]

4.2 Calculational Design of Hoare Proof Rules

Theorem 4.3 (\( \text{HIL} \) rules for conditional iteration).

\[
\begin{align*}
\text{HIL}(w) = \{ \langle P, Q \rangle \mid \exists \{ a \wedge S \} \subseteq \text{HIL}(w) \wedge \forall i \in \{ a \wedge S \} \} & \text{with condition (3)}
\end{align*}
\]

Proof of (3). We write \( \langle P \rangle \subseteq \text{HIL}(w) \subseteq \text{HIL}(w) \subseteq (3). \)

By structural induction (5) being a strict component of \( \text{HIL}(w) \), the rule for \( \langle P \rangle \subseteq \text{HIL}(w) \subseteq (3). \) have already been defined.

By Aczel method, the (constant) fixpoint \( \text{while Bl (S)} \) is defined by \( (\mathcal{E} \in S) \).

For while Bl (S) we have an axiom \( \langle \text{while Bl (S)} \rangle \subseteq (3) \), with side condition \( \langle a \wedge S \rangle \subseteq \text{HIL}(w) \wedge \forall i \in \{ a \wedge S \} \).
Conclusion

A transformational logic is an abstract interpretation of a natural relational semantics
The End, Thank You

- slides + calculational design + recording are online on my web page (https://cs.nyu.edu/~pcousot/)
- paper + appendix = 1 clickable file on Zenodo https://zenodo.org/records/10439109