Is Peter Correct or Incorrect?

Patrick Cousot

Courant Institute, New York University

Peter's Incorrectness Logic

- In POPL 2020, Peter O'Hearn introduced the nonconformist idea of an incorrectness logic

> We explore our hypothesis by defining incorrectness logic, a formalism that is similar to Hoare's logic of program correctness [Hoare 1969], except that it is oriented to proving incorrectness rather than correctness.

Peter's Incorrectness Logic

- In POPL 2020, Peter O'Hearn introduced the nonconformist idea of an incorrectness logic

> We explore our hypothesis by defining incorrectness logic, a formalism that is similar to Hoare's logic of program correctness [Hoare 1969], except that it is oriented to proving incorrectness rather than correctness.

- Is it?

Peter's Incorrectness Logic

- And he moderately enjoyed other approaches to incorrectness
- Such as ``necessary preconditions''

The concept of *necessary preconditon* [Cousot et al. 2013] is related. A necessary precondition for a program is a predicate which, whenever falsified, leads to divergence or an error, but never to successful termination.

Peter's Incorrectness Logic

- And he moderately enjoyed other approaches to incorrectness
- Such as ``necessary preconditions"

The concept of *necessary preconditon* [Cousot et al. 2013] is related. A necessary precondition for a program is a predicate which, whenever falsified, leads to divergence or an error, but never to successful termination.

- But he doesn't really like it!

... Finally, there are programs for which no non-trivial necessary pre-condition exists (e.g., `skip + error()`), but where perfectly fine presumptions exist for incorrectness logic.

Peter's Incorrectness Logic

- And he moderately enjoyed other approaches to incorrectness
- Such as ``necessary preconditions''

The concept of *necessary preconditon* [Cousot et al. 2013] is related. A necessary precondition for a program is a predicate which, whenever falsified, leads to divergence or an error, but never to successful termination.

- But he doesn't really like it!

... Finally, there are programs for which no non-trivial necessary pre-condition exists (e.g., `skip + error()`), but where perfectly fine presumptions exist for incorrectness logic.

- Should he?

Peter’s Incorrectness Logic

In summary, there is a rich variety of problems for both experimental and theoretical work to bring the foundations of reasoning about program incorrectness onto a par with the extensively developed foundations for correctness.

An *A Parte* on Singularities of Logics

Emptiness versus Universality

- Emptiness: some programs satisfy no formula of the logic
  - Ex. 1: a potentially nonterminating program satisfies no formula of the Manna-Pnueli total correctness logic
  - Ex. 2: Peter's example for "necessary preconditions"

Emptiness versus Universality

- Emptiness: some programs satisfy no formula of the logic
  - Ex. 1: a potentially nonterminating program satisfies no formula of the Manna-Pnueli total correctness logic
  - Ex. 2: Peter's example for "necessary preconditions"
- Universality: some programs satisfy all formulas of the logic
  - Ex. 1: `W = while (true) skip` satisfies all Hoare triples {P} `W` {Q}

Emptiness versus Universality

- Emptiness: some programs satisfy no formula of the logic
  - Ex. 1: a potentially nonterminating satisfies no formula of the Manna-Pnueli total correctness logic
  - Ex. 2: Peter's example for ``necessary preconditions"
- Universality: some programs satisfy all formulae of the logic
  - Ex. 1: `W` = `while (true) skip` satisfies all Hoare triples {P} `W` {Q}
- Same in logic: false is never satisfied and true is always satisfied

Foundations of Reasoning on Logics

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 $[\![S]\!]_\perp$ of the programming language (in structural fixpoint form)
2. Define the theory of the logics as an abstraction $\alpha(\{[\![S]\!]_\perp\})$ of the collecting semantics $\{[\![S]\!]_\perp\}$ (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 $[\![\mathsf{S}]\!]_\perp$ of the programming language (in structural fixpoint form)
2. Define the theory of the logics as an abstraction $\alpha(\{[\![\mathsf{S}]\!]_\perp\})$ of the collecting semantics $\{[\![\mathsf{S}]\!]_\perp\}$ (strongest (hyper) property)
3. Calculate the theory $\alpha(\{[\![\mathsf{S}]\!]_\perp\})$ 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 $[\![S]\!]_\perp$ of the programming language (in structural fixpoint form)
2. Define the theory of the logics as an abstraction $\alpha(\{[\![S]\!]_\perp\})$ of the collecting semantics $\{[\![S]\!]_\perp\}$ (strongest (hyper) property)
3. Calculate the theory $\alpha(\{[\![S]\!]_\perp\})$ 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

The Design of
Hoare Incorrectness Logic ($\overline{\text{HL}}$)

1) Relational semantics

1. Angelic relational semantics $[\![S]\!]^e$

- Syntax*:

$$\texttt{S} \in \mathbb{S} ::= \texttt{x = A} \mid \texttt{skip} \mid \texttt{S;S} \mid \texttt{if (B) S else S} \mid \texttt{while (B) S}$$

- States: Σ

- Angelic relational semantics: $[\![S]\!]^e \in \wp(\Sigma \times \Sigma)$ (e = ends)

1. Angelic relational semantics $[\![\mathtt{S}]\!]$ (in deductive form)

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

I. Angelic relational semantics $[\![S]\!]$ (in deductive form)

- Notations using judgements:
  - $\sigma \vdash \mathtt{S} \stackrel{e}{\Rightarrow} \sigma'$ for $\langle \sigma, \sigma' \rangle \in [\![\mathtt{S}]\!]^e$
  - $\sigma \vdash \mathtt{while(B)\ S} \stackrel{i}{\Rightarrow} \sigma'$ for σ leads to σ' after 0 or more iterations
- Semantics of the conditional iteration* $\mathtt{W} = \mathtt{while(B)\ S}$:

$$\text{(a)} \quad \sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma \qquad\qquad \text{(b)} \quad \frac{\mathcal{B}[\![\mathtt{B}]\!]\sigma, \quad \sigma \vdash \mathtt{S} \stackrel{e}{\Rightarrow} \sigma', \quad \sigma' \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma''}{\sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma''} \tag{2}$$

$$\text{(a)} \quad \frac{\sigma \vdash \mathtt{W} \stackrel{i}{\Rightarrow} \sigma', \quad \mathcal{B}[\![\neg\mathtt{B}]\!]\sigma'}{\sigma \vdash \mathtt{W} \stackrel{e}{\Rightarrow} \sigma'} \tag{3}$$

\

1. Angelic relational semantics $[\![\mathtt{S}]\!]$ (in fixpoint form)

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

$$F^e(X) \triangleq \mathsf{id} \cup ([\![\mathsf{B}]\!] \mathbin{;} [\![\mathsf{S}]\!]^e \mathbin{;} X), \quad X \in \wp(\Sigma \times \Sigma) \tag{49}$$

$$[\![\mathtt{while\ (B)\ S}]\!]^e \triangleq \mathsf{lfp}^{\subseteq} F^e \mathbin{;} [\![\neg\mathsf{B}]\!] \tag{51}$$

- Derived using Aczel correspondence between deductive systems and set-theoretic fixpoints (forthcoming)

II) Abstraction of the semantics to the theory

Exact abstractions

Abstraction

- Hyper properties to properties abstraction:

$$\langle \wp(\wp(\Sigma \times \Sigma)), \subseteq \rangle \xleftarrow[\alpha_C]{\gamma_C} \langle \wp(\Sigma \times \Sigma), \subseteq \rangle \qquad \alpha_C(P) \triangleq \bigcup P \qquad \gamma_C(S) \triangleq \wp(S)$$

Abstraction

- Hyper properties to properties abstraction:

$$\langle \wp(\wp(\Sigma \times \Sigma)), \subseteq \rangle \xrightleftharpoons[\alpha_C]{\gamma_C} \langle \wp(\Sigma \times \Sigma), \subseteq \rangle \qquad \alpha_C(P) \triangleq \bigcup P \qquad \gamma_C(S) \triangleq \wp(S)$$

- Post-image isomorphism:

$$\langle \wp(\Sigma \times \Sigma), \subseteq \rangle \xrightleftharpoons[\text{post}]{\widetilde{\text{pre}}} \langle \wp(\Sigma) \to \wp(\Sigma), \subseteq \rangle \quad \text{post}(R) \triangleq \lambda P \cdot \{\sigma' \mid \exists \sigma \in P \wedge \langle \sigma, \sigma' \rangle \in R\}$$

$$\widetilde{\text{pre}}(R) \triangleq \lambda X \cdot \{\sigma \mid \forall \sigma' \in Q \,.\, \langle \sigma, \sigma' \rangle \in R\}$$

Abstraction

- Hyper properties to properties abstraction:

$$\langle \wp(\wp(\Sigma \times \Sigma)),\ \subseteq\rangle \underset{\alpha_C}{\overset{\gamma_C}{\leftrightarrows}} \langle \wp(\Sigma \times \Sigma),\ \subseteq\rangle \qquad \alpha_C(P) \triangleq \bigcup P \qquad \gamma_C(S) \triangleq \wp(S)$$

- Post-image isomorphism:

$$\langle \wp(\Sigma \times \Sigma),\ \subseteq\rangle \underset{\text{post}}{\overset{\widetilde{\text{pre}}}{\leftrightarrows}} \langle \wp(\Sigma) \rightarrow \wp(\Sigma),\ \subseteq\rangle \quad \text{post}(R) \triangleq \boldsymbol{\lambda} P \bullet \{\sigma' \mid \exists \sigma \in P \wedge \langle \sigma,\ \sigma'\rangle \in R\}$$

$$\widetilde{\text{pre}}(R) \triangleq \boldsymbol{\lambda} X \bullet \{\sigma \mid \forall \sigma' \in Q\ .\ \langle \sigma,\ \sigma'\rangle \in R\}$$

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

$$\langle \wp(\Sigma) \rightarrow \wp(\Sigma),\ =\rangle \underset{\alpha_G}{\overset{\gamma_G}{\leftrightarrows}} \langle \wp_{\text{fun}}(\wp(\Sigma) \times \wp(\Sigma)),\ =\rangle \quad f \in \wp(\Sigma) \rightarrow \wp(\Sigma)$$

$$\alpha_G(f) = \{\langle P,\ f(P)\rangle \mid P \in \wp(\Sigma)\}$$

$$\gamma_G(R) \triangleq \boldsymbol{\lambda} P \bullet (Q \textit{ such that } \langle P,\ S\rangle \in R)$$

Abstraction

- Negation abstraction:

$$X \in \wp(\mathcal{X}),\ \alpha^{\neg}(X) \triangleq \neg X \text{ (where } \neg X \triangleq \mathcal{X} \setminus X)$$

$$\langle \wp(\mathcal{X}),\ \subseteq\rangle \xleftarrow[\alpha^{\neg}]{\alpha^{\neg}} \langle \wp(\mathcal{X}),\ \supseteq\rangle \quad \text{and} \quad \langle \wp(\mathcal{X}),\ \supseteq\rangle \xleftarrow[\alpha^{\neg}]{\alpha^{\neg}} \langle \wp(\mathcal{X}),\ \subseteq\rangle$$

Consequence approximation

Approximation abstraction

- The component wise approximation:

$$\langle x, y \rangle \sqsubseteq, \preceq \langle x', y' \rangle \quad \triangleq \quad x \sqsubseteq x' \wedge y \preceq y'$$

Approximation abstraction

- The component wise approximation:

$$\langle x, y\rangle \sqsubseteq,\preceq \langle x', y'\rangle \quad \triangleq \quad x \sqsubseteq x' \wedge y \preceq y'$$

- Over-approximation:

$$\text{post}(\subseteq, \supseteq) \quad = \quad \boldsymbol{\lambda} R \bullet \{\langle P, Q\rangle \mid \exists \langle P', Q'\rangle \in R \,.\, P \subseteq P' \wedge Q' \subseteq Q\}$$

Approximation abstraction

- The component wise approximation:

$$\langle x, y\rangle \sqsubseteq,\preceq \langle x', y'\rangle \quad \triangleq \quad x \sqsubseteq x' \wedge y \preceq y'$$

- Over-approximation:

$$\text{post}(\subseteq, \supseteq) \quad = \quad \boldsymbol{\lambda} R \bullet \{\langle P, Q\rangle \mid \exists \langle P', Q'\rangle \in R \,.\, P \subseteq P' \wedge Q' \subseteq Q\}$$

- Under-approximation:

$$\text{post}(\supseteq, \subseteq) \quad = \quad \boldsymbol{\lambda} R \bullet \{\langle P, Q\rangle \mid \exists \langle P', Q'\rangle \in R \,.\, P' \subseteq P \wedge Q \subseteq Q'\}$$

Comparing logics through their theories

Comparing logics through their theories

- Strongest postcondition logic (SL): $\mathcal{T}(\mathsf{S}) \triangleq \alpha_{\mathrm{G}} \circ \mathbf{post} \circ \alpha_C(\{[\![\mathsf{S}]\!]\})$

$$= \{\langle P,\ \mathbf{post}[\![\mathsf{S}]\!]P\rangle \mid P \in \wp(\Sigma)\}$$

Comparing logics through their theories

- Strongest postcondition logic (SL): $\mathcal{T}(\mathtt{S}) \triangleq \alpha_{\mathrm{G}} \circ \mathrm{post} \circ \alpha_{C}(\{[\![\mathtt{S}]\!]\})$
  $= \{\langle P,\ \mathrm{post}[\![\mathtt{S}]\!]P\rangle \mid P \in \wp(\Sigma)\}$

- Hoare logic (HL): $\mathcal{T}_{\mathrm{HL}}(\mathtt{S}) \triangleq \mathrm{post}(\supseteq.\subseteq) \circ \mathcal{T}(\mathtt{S})$

Comparing logics through their theories

- Strongest postcondition logic (SL): $\mathcal{T}(\mathtt{S}) \triangleq \alpha_{\mathrm{G}} \circ \mathsf{post} \circ \alpha_{C}(\{[\![\mathtt{S}]\!]\})$

$$= \{\langle P,\ \mathsf{post}[\![\mathtt{S}]\!]P\rangle \mid P \in \wp(\Sigma)\}$$

- Hoare logic (HL): $\mathcal{T}_{\mathrm{HL}}(\mathtt{S}) \triangleq \mathsf{post}(\supseteq.\subseteq) \circ \mathcal{T}(\mathtt{S})$

- Incorrectness logic (IL): $\mathcal{T}_{\mathrm{IL}}(\mathtt{S}) \triangleq \mathsf{post}(\subseteq.\supseteq) \circ \mathcal{T}(\mathtt{S})$

Comparing logics through their theories

- Strongest postcondition logic (SL): $\mathcal{T}(\mathtt{S}) \triangleq \alpha_{\mathrm{G}} \circ \mathrm{post} \circ \alpha_C(\{[\![\mathtt{S}]\!]\})$
  $= \{\langle P,\ \mathrm{post}[\![\mathtt{S}]\!]P\rangle \mid P \in \wp(\Sigma)\}$

- Hoare logic (HL): $\mathcal{T}_{\mathrm{HL}}(\mathtt{S}) \triangleq \mathrm{post}(\supseteq.\subseteq) \circ \mathcal{T}(\mathtt{S})$

- Incorrectness logic (IL): $\mathcal{T}_{\mathrm{IL}}(\mathtt{S}) \triangleq \mathrm{post}(\subseteq.\supseteq) \circ \mathcal{T}(\mathtt{S})$

- Hoare incorrectness logic ($\overline{\mathrm{HL}}$): $\mathcal{T}_{\overline{\mathrm{HL}}}(\mathtt{S}) \triangleq \mathrm{post}(\supseteq.\subseteq) \circ \alpha^{\neg} \circ \mathcal{T}_{\mathrm{HL}}(\mathtt{S})$

Comparing logics through their theories

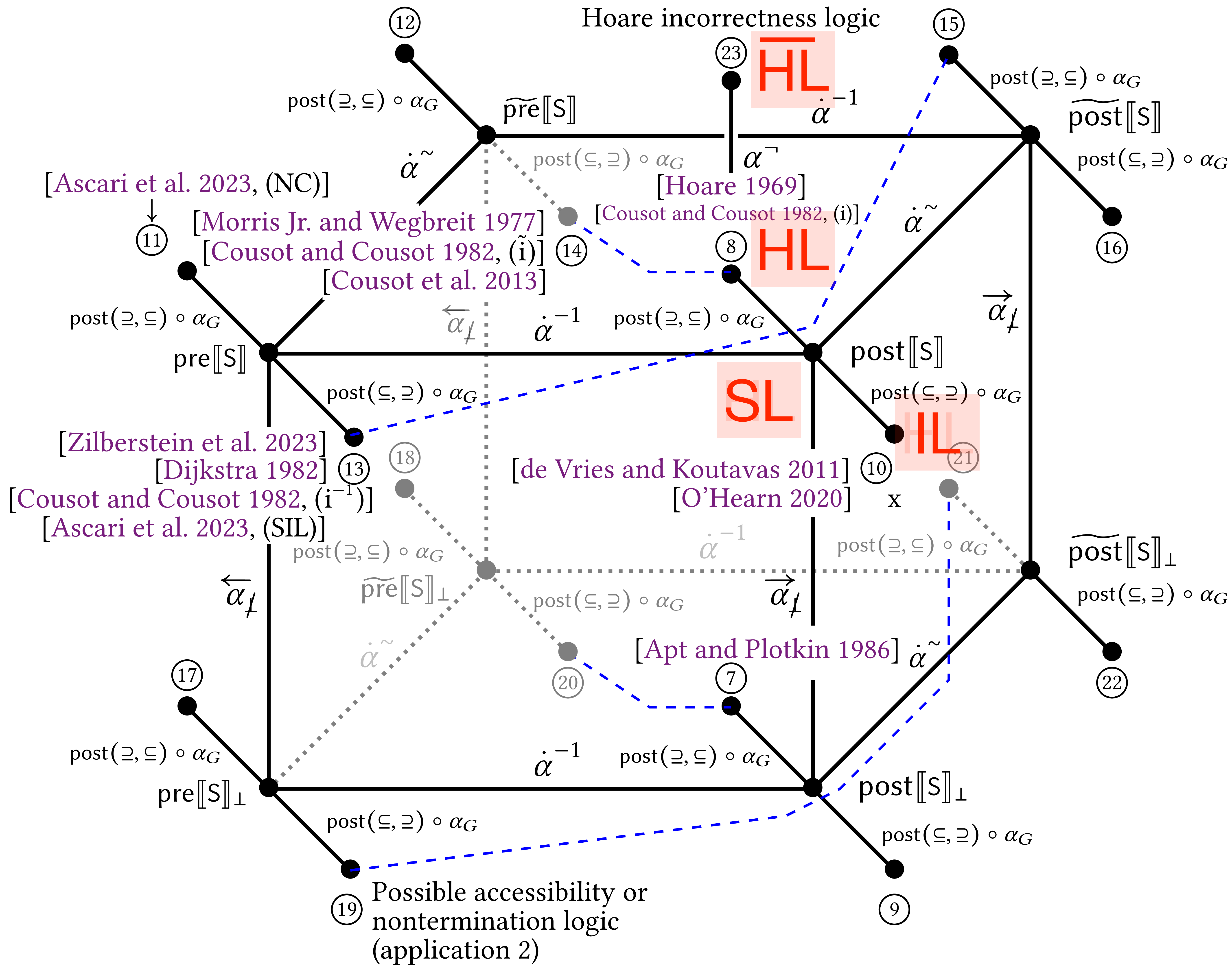

Fig. 3. Hierarchical taxonomy of transformational assertional logics

Fixpoint abstraction

2. Abstraction

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

THEOREM II.2.1 (FIXPOINT ABSTRACTION). *If* $\langle C, \sqsubseteq\rangle \xrightleftharpoons[\alpha]{\gamma} \langle A, \preceq\rangle$ *is a Galois connection between complete lattices* $\langle C, \sqsubseteq\rangle$ *and* $\langle A, \preceq\rangle$, $f \in C \xrightarrow{i} C$ *and* $\bar{f} \in A \xrightarrow{i} A$ *are increasing and commuting, that is,* $\alpha \circ f = \bar{f} \circ \alpha$, *then* $\alpha(\mathsf{lfp}^{\sqsubseteq} f) = \mathsf{lfp}^{\preceq} \bar{f}$ *(while semi-commutation* $\alpha \circ f \preceq \bar{f} \circ \alpha$ *implies* $\alpha(\mathsf{lfp}^{\sqsubseteq} f) \preceq \mathsf{lfp}^{\preceq} \bar{f}$*).*

2. Abstraction

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

THEOREM II.2.1 (FIXPOINT ABSTRACTION). *If* $\langle C, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \preceq\rangle$ *is a Galois connection between complete lattices* $\langle C, \sqsubseteq\rangle$ *and* $\langle A, \preceq\rangle$, $f \in C \xrightarrow{i} C$ *and* $\bar{f} \in A \xrightarrow{i} A$ *are increasing and commuting, that is,* $\alpha \circ f = \bar{f} \circ \alpha$, *then* $\alpha(\mathsf{lfp}^{\sqsubseteq} f) = \mathsf{lfp}^{\preceq} \bar{f}$ *(while semi-commutation* $\alpha \circ f \preceq \bar{f} \circ \alpha$ *implies* $\alpha(\mathsf{lfp}^{\sqsubseteq} f) \preceq \mathsf{lfp}^{\preceq} \bar{f}$*).*

- We get a fixpoint definition of the theory of strongest postconditions logic (SL)
- For the iteration `W = while (B) S` :

$$\mathcal{T}(\mathtt{W}) \triangleq \{\langle P, \mathsf{post}[\![\neg\mathtt{B}]\!](\mathsf{lfp}^{\subseteq} \boldsymbol{\lambda} X \cdot P \cup \mathsf{post}([\![\mathtt{B}]\!] \,\mathring{;}\, [\![\mathtt{S}]\!]^{e})X)\rangle \mid P \in \wp(\Sigma)\}$$

1 PROPERTIES OF STRONGEST POSTCONDITIONS

Lemma 1.1 (Composition). $\mathsf{post}(X \mathbin{;} Y) = \mathsf{post}(Y) \circ \mathsf{post}(X)$.

Proof of Lem. 1.1.

$$\begin{aligned}
&\mathsf{post}(X \mathbin{;} Y)\\
=\ &\lambda P \bullet \{\sigma'' \mid \exists \sigma \in P \,.\, \langle \sigma,\sigma''\rangle \in X \mathbin{;} Y\} && \wr\text{def. post}\wr\\
=\ &\lambda P \bullet \{\sigma'' \mid \exists \sigma \in P \,.\, \exists \sigma' \,.\, \langle \sigma,\sigma'\rangle \in X \wedge \langle \sigma',\sigma''\rangle \in Y\} && \wr\text{def. } ;\wr\\
=\ &\lambda P \bullet \{\sigma'' \mid \exists \sigma' \,.\, \sigma' \in \{\sigma' \mid \exists \sigma \in P \,.\, \langle \sigma,\sigma'\rangle \in X\} \wedge \langle \sigma',\sigma''\rangle \in Y\} && \wr\text{def. } \exists \text{ and } \in\wr\\
=\ &\lambda P \bullet \{\sigma'' \mid \exists \sigma' \in \mathsf{post}(X)P \,.\, \langle \sigma',\sigma''\rangle \in Y\} && \wr\text{def. post}\wr\\
=\ &\lambda P \bullet \mathsf{post}(Y)(\mathsf{post}(X)P) && \wr\text{def. post}\wr\\
=\ &\mathsf{post}(Y) \circ \mathsf{post}(X) && \wr\text{def. function composition } \circ\wr \quad \square
\end{aligned}$$

Lemma 1.2 (test). $\mathsf{post}\llbracket \mathtt{B}\rrbracket P = P \cap \mathcal{B}\llbracket \mathtt{B}\rrbracket$.

Proof of Lem. 1.2.

$$\begin{aligned}
&\mathsf{post}\llbracket \mathtt{B}\rrbracket P\\
=\ &\{\sigma' \mid \exists \sigma \in P \,.\, \langle \sigma,\sigma'\rangle \in \llbracket \mathtt{B}\rrbracket\} && \wr\text{def. post}\wr\\
=\ &\{\sigma \mid \sigma \in P \wedge \sigma \in \mathcal{B}\llbracket \mathtt{B}\rrbracket\} && \wr\text{def. } \llbracket \mathtt{B}\rrbracket \triangleq \{\langle \sigma,\sigma\rangle \mid \sigma \in \mathcal{B}\llbracket \mathtt{B}\rrbracket\}\wr\\
=\ &P \cap \mathcal{B}\llbracket \mathtt{B}\rrbracket && \wr\text{def. intersection } \cup\wr \quad \square
\end{aligned}$$

Lemma 1.3 (Strongest postcondition). $\mathcal{T}(\mathtt{S}) = \alpha_G \circ \mathsf{post}\llbracket \mathtt{S}\rrbracket = \{\langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle \mid P \in \wp(\Sigma)\}$.

Proof of Lem. 1.3.

$$\begin{aligned}
&\mathcal{T}(\mathtt{S})\\
=\ &\alpha_G \circ \mathsf{post} \circ \alpha_{\not\perp} \circ \alpha_C(\{\llbracket \mathtt{S}\rrbracket_\perp\}) && \wr\text{def. } \mathcal{T}\wr\\
=\ &\alpha_G \circ \mathsf{post} \circ \alpha_{\not\perp}(\llbracket \mathtt{S}\rrbracket_\perp) && \wr\text{def. } \alpha_C\wr\\
=\ &\alpha_G \circ \mathsf{post}(\llbracket \mathtt{S}\rrbracket_\perp \cap (\Sigma \times \Sigma)) && \wr\text{def. } \alpha_{\not\perp}\wr\\
=\ &\alpha_G \circ \mathsf{post}\llbracket \mathtt{S}\rrbracket && \wr\text{def. (1) of the angelic semantics } \llbracket \mathtt{S}\rrbracket\wr\\
=\ &\{\langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle \mid P \in \wp(\Sigma)\} && \wr\text{def. } \alpha_G\wr \quad \square
\end{aligned}$$

Lemma 1.4 (Strongest postcondition over approximation).

$$\mathcal{T}_{\mathrm{HL}}(\mathtt{S}) \quad\triangleq\quad \mathsf{post}(\supseteq,\subseteq) \circ \mathcal{T}(\mathtt{S}) \quad=\quad \{\langle P,\ Q\rangle \mid \mathsf{post}\llbracket \mathtt{S}\rrbracket P \subseteq Q\} \quad=\quad \mathsf{post}(=,\subseteq) \circ \mathcal{T}(\mathtt{S})$$

Proof of Lem. 1.4.

$$\begin{aligned}
&\mathsf{post}(\supseteq,\subseteq) \circ \mathcal{T}(\mathtt{S})\\
=\ &\mathsf{post}(\supseteq,\subseteq)(\mathcal{T}(\mathtt{S})) && \wr\text{def. function composition } \circ\wr\\
=\ &\mathsf{post}(\supseteq,\subseteq)(\{\langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle \mid P \in \wp(\Sigma)\}) && \wr\text{Lem. 1.3}\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists \langle P,\ Q\rangle \in \{\langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle \mid P \in \wp(\Sigma)\} \,.\, \langle\langle P,\ Q\rangle,\ \langle P',\ Q'\rangle\rangle \in \supseteq,\subseteq\} && \wr\text{def. (10) of post}\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists P \,.\, \langle\langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle,\ \langle P',\ Q'\rangle\rangle \in \supseteq,\subseteq\} && \wr\text{def. } \in\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists P \,.\, \langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle \supseteq,\subseteq \langle P',\ Q'\rangle\} && \wr\text{def. } \in\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists P \,.\, P \supseteq P' \wedge \mathsf{post}\llbracket \mathtt{S}\rrbracket P \subseteq Q'\} && \wr\text{def. } \supseteq,\subseteq\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists P \,.\, P' \subseteq P \wedge \mathsf{post}\llbracket \mathtt{S}\rrbracket P \subseteq Q'\} && \wr\text{def. } \supseteq\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \mathsf{post}\llbracket \mathtt{S}\rrbracket P' \subseteq Q'\}
\end{aligned}$$

$\wr(\subseteq)$ by Galois connection (12), post is increasing so that $P' \subseteq P \wedge \mathsf{post}\llbracket \mathtt{S}\rrbracket P \subseteq Q'$ implies $\mathsf{post}\llbracket \mathtt{S}\rrbracket P' \subseteq \mathsf{post}\llbracket \mathtt{S}\rrbracket P \wedge \mathsf{post}\llbracket \mathtt{S}\rrbracket P \subseteq Q'$ hence $\mathsf{post}\llbracket \mathtt{S}\rrbracket P' \subseteq Q'$ by transitivity;
$(\supseteq)$ take $P = P'\wr$

$$\begin{aligned}
=\ &\{\langle P',\ Q'\rangle \mid \exists P \,.\, P' = P \wedge \mathsf{post}\llbracket \mathtt{S}\rrbracket P \subseteq Q'\} && \wr\text{def. } =\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists P \,.\, \langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle =,\subseteq \langle P',\ Q'\rangle\} && \wr\text{def. } =,\subseteq\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists P \,.\, \langle\langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle,\ \langle P',\ Q'\rangle\rangle \in =,\subseteq\} && \wr\text{def. } \in\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists \langle P,\ Q\rangle \in \{\langle P,\ \mathsf{post}\llbracket \mathtt{S}\rrbracket P\rangle \mid P \in \wp(\Sigma)\} \,.\, \langle\langle P,\ Q\rangle,\ \langle P',\ Q'\rangle\rangle \in =,\subseteq\} && \wr\text{def. } \in\wr\\
=\ &\{\langle P',\ Q'\rangle \mid \exists \langle P,\ Q\rangle \in \mathcal{T}(\mathtt{S}) \,.\, \langle\langle P,\ Q\rangle,\ \langle P',\ Q'\rangle\rangle \in =,\subseteq\} && \wr\text{Lem. 1.3}\wr\\
=\ &\mathsf{post}(=,\subseteq)(\mathcal{T}(\mathtt{S})) && \wr\text{def. (10) of post}\wr\\
=\ &\mathsf{post}(=,\subseteq) \circ \mathcal{T}(\mathtt{S}) && \wr\text{def. function composition } \circ\wr \quad \square
\end{aligned}$$

For simplicity, we consider conditional iteration `W = while (B) S` with no `break`.

Lemma 1.5 (Commutation). $\mathsf{post} \circ F'^e = \bar{F}^e \circ \mathsf{post}$ *where* $\bar{F}^e(X) \triangleq \mathrm{id} \mathbin{\dot\cup} (\mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e) \circ X)$ *and* $F'^e \triangleq \lambda X \bullet \mathrm{id} \cup (X \mathbin{;} \llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e)$, $X \in \wp(\Sigma \times \Sigma)$ *by (70).*

Proof of Lem. 1.5.

$$\begin{aligned}
&\mathsf{post}(F'^e(X)) && \wr\text{where } X \in \wp(\Sigma)\wr\\
=\ &\mathsf{post}(\mathrm{id} \cup (X \mathbin{;} \llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e)) && \wr\text{def. } F^e\wr\\
=\ &\mathsf{post}(\mathrm{id}) \mathbin{\dot\cup} \mathsf{post}(X \mathbin{;} \llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e) && \wr\text{join preservation in Galois connection (12)}\wr\\
=\ &\mathrm{id} \mathbin{\dot\cup} (\mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e) \circ \mathsf{post}(X)) && \wr\text{def. post and composition Lem. 1.1}\wr\\
=\ &\bar{F}^e(\mathsf{post}(X)) && \wr\text{def. } \bar{F}^e\wr \quad \square
\end{aligned}$$

Lemma 1.6 (Pointwise commutation). $\forall X \in \wp(\Sigma) \to \wp(\Sigma) \,.\, \forall P \in \wp(\Sigma) \,.\, \bar{F}^e(X)P \triangleq \bar{\bar{F}}{}^e_P(X(P))$ *where* $\bar{\bar{F}}{}^e_P(X) \triangleq P \cup \mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e)X$.

Proof of Lem. 1.6.

$$\begin{aligned}
&\bar{F}^e(X)P\\
=\ &(\mathrm{id} \mathbin{\dot\cup} (\mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e) \circ X))P && \wr\text{def. } \bar{F}^e\wr\\
=\ &\mathrm{id}(P) \cup (\mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e) \circ X)(P) && \wr\text{pointwise def. } \dot\cup \text{ and function composition } \circ\wr\\
=\ &P \cup \mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e)(X(P)) && \wr\text{def. identity id and function application}\wr\\
=\ &\bar{\bar{F}}{}^e_P(X(P)) && \wr\text{def. } \bar{\bar{F}}{}^e_P(X) \triangleq P \cup \mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e)X\wr \quad \square
\end{aligned}$$

Theorem 1.7 (Iteration strongest postcondition). $\mathsf{post}\llbracket \mathtt{W}\rrbracket P = \mathsf{post}\llbracket \neg\mathtt{B}\rrbracket(\mathsf{lfp}^{\subseteq} \bar{\bar{F}}{}^e_P)$ *where* $\bar{\bar{F}}{}^e_P(X) \triangleq P \cup \mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e)X$.

Proof of Th. 1.7.

$$\begin{aligned}
&\mathsf{post}\llbracket \mathtt{W}\rrbracket\\
=\ &\mathsf{post}(\mathsf{lfp}^{\subseteq} F^e \mathbin{;} \llbracket \neg\mathtt{B}\rrbracket) && \wr\text{def. (49) of } \llbracket \mathtt{W}\rrbracket \text{ in absence of break}\wr\\
=\ &\mathsf{post}\llbracket \neg\mathtt{B}\rrbracket \circ \mathsf{post}(\mathsf{lfp}^{\subseteq} F^e) && \wr\text{composition Lem. 1.1}\wr\\
=\ &\mathsf{post}\llbracket \neg\mathtt{B}\rrbracket \circ \mathsf{post}(\mathsf{lfp}^{\subseteq} F'^e) && \wr\text{since } \mathsf{lfp}^{\subseteq} F^e = \mathsf{lfp}^{\subseteq} F'^e \text{ in (70)}\wr\\
=\ &\mathsf{post}\llbracket \neg\mathtt{B}\rrbracket(\mathsf{lfp}^{\subseteq} \bar{F}^e) && \wr\text{commutation Lem. 1.5 and fixpoint abstraction Th. II.2.2}\wr\\
=\ &\mathsf{post}\llbracket \neg\mathtt{B}\rrbracket \circ \lambda P \bullet \mathsf{lfp}^{\subseteq} \bar{\bar{F}}{}^e_P && \wr\text{pointwise commutation Lem. 1.6 and pointwise abstraction Cor. II.2.2}\wr \quad \square
\end{aligned}$$

Corollary 1.8 (Conditional iteration strongest postcondition graph). $\mathcal{T}(\mathtt{W}) = \{\langle P,\ \mathsf{post}\llbracket \neg\mathtt{B}\rrbracket(\mathsf{lfp}^{\subseteq} \bar{\bar{F}}{}^e_P)\rangle \mid P \in \wp(\Sigma)\}$ *where* $\bar{\bar{F}}{}^e_P(X) \triangleq P \cup \mathsf{post}(\llbracket \mathtt{B}\rrbracket \mathbin{;} \llbracket \mathtt{S}\rrbracket^e)X$.

Proof of Cor. 1.8.

$$\begin{aligned}
&\mathcal{T}(\mathtt{W})\\
=\ &\alpha_G \circ \mathsf{post}(\llbracket \mathtt{W}\rrbracket) && \wr\text{Lem. 1.3}\wr\\
=\ &\alpha_G \circ \mathsf{post}\llbracket \neg\mathtt{B}\rrbracket \circ \lambda P \bullet \mathsf{lfp}^{\subseteq} \bar{\bar{F}}{}^e_P && \wr\text{Th. 1.7}\wr\\
=\ &\{\langle P,\ \mathsf{post}\llbracket \neg\mathtt{B}\rrbracket(\mathsf{lfp}^{\subseteq} \bar{\bar{F}}{}^e_P)\rangle \mid P \in \wp(\Sigma)\} && \wr\text{def. (7) of } \alpha_G\wr \quad \square
\end{aligned}$$

IV) Design of the proof system

Aczel correspondence

Aczel correspondence between deductive systems and fixpoints

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

Aczel correspondence between deductive systems and fixpoints

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

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

Aczel correspondence between deductive systems and fixpoints

- Rules: $\frac{P}{c}$ ($\mathcal{U}$ universe, $P \in \wp_{\mathrm{fin}}(\mathcal{U})$ premiss, $c \in \mathcal{U}$ conclusion, $\frac{\varnothing}{c}$ axiom)
- Deductive system : $R = \left\{ \frac{P_i}{c_i} \;\middle|\; i \in \Delta \right\}$, $\quad R \in \wp(\wp_{\mathrm{fin}}(\mathcal{U}) \times \mathcal{U})$
- Subset of the universe $\mathcal{U}$ defined by R:

proof theoretic ↓

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

$$= \mathrm{lfp}^{\subseteq} F(R)$$

← model theoretic (gfp for coinduction)

$$F(R)X \triangleq \left\{ c \;\middle|\; \exists \frac{P}{c} \in R \,.\, P \subseteq X \right\}$$

← consequence operator

Aczel correspondence between deductive systems and fixpoints

- Rules: $\dfrac{P}{c}$ ($\mathcal{U}$ universe, $P \in \wp_{\text{fin}}(\mathcal{U})$ premiss, $c \in \mathcal{U}$ conclusion, $\dfrac{\varnothing}{c}$ axiom)
- Deductive system : $R = \left\{\dfrac{P_i}{c_i} \;\middle|\; i \in \Delta\right\}, \quad R \in \wp(\wp_{\text{fin}}(\mathcal{U}) \times \mathcal{U})$
- Subset of the universe $\mathcal{U}$ defined by R:

proof theoretic ↓

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

$$= \quad \mathsf{lfp}^{\subseteq} F(R)$$

← model theoretic (gfp for coinduction)

$$F(R)X \triangleq \left\{c \;\middle|\; \exists \frac{P}{c} \in R \,.\, P \subseteq X\right\}$$

← consequence operator

- Deductive system defining $\mathsf{lfp}^{\subseteq} F$: $\quad R_F \triangleq \left\{\dfrac{P}{c} \;\middle|\; P \subseteq \mathcal{U} \wedge c \in F(P)\right\}$

Why not using Aczel method to get the proof system at this point?

- We get a sound and complete proof system

Why not using Aczel method to get the proof system at this point?

- We get a sound and complete proof system
- BUT impractical:
  - you first prove the strongest postcondition, and then
  - use the consequence rule to approximate!

Fixpoint induction

Fixpoint induction

THEOREM H.3 (NON EMPTY INTERSECTION WITH ABSTRACTION OF LEAST FIXPOINT). *Assume that (1)* $\langle L, \sqsubseteq, \bot, \top, \sqcap, \sqcup\rangle$ *is an atomic complete lattice; (2)* $f \in L \to L$ *preserves nonempty joins* $\sqcup$*; (3)* $\langle L, \sqsubseteq\rangle \xrightleftharpoons[\alpha]{\gamma} \langle \bar{L}, \preceq, \curlywedge\rangle$*; (4)* $\bar{Q} \in \bar{L} \setminus \{0\}$ *where* $0 \triangleq \alpha(\bot)$*; (5) There exists an inductive invariant* $I \in L$ *of* f *(i.e.* $f(I) \sqsubseteq I$*); (6)* $\langle W, \preccurlyeq\rangle$ *is a well-founded set and* $v \in \mathsf{atoms}(I) \to W$ *is a (variant) function; (7) There exists a sequence* $\langle a_i \in \mathsf{atoms}(I), i \in [1, \infty]\rangle$ *that (7.a)* $a_1 \in f(\bot)$*, (7.b)* $\forall i \in [1, \infty] \,.\, a_{i+1} \in \mathsf{atoms}(f(a_i))$*, (7.c)* $\forall i \in [1, \infty] \,.\, (a_i \neq a_{i+1}) \Rightarrow (v(a_i) \succ v(a_{i+1})$*, (7.d)* $\forall i \in [1, \infty] \,.\, (v(a_i) \not\succ v(a_{i+1}) \Rightarrow \alpha(a_i) \curlywedge \bar{Q} \neq 0$*; Then, hypotheses (1) to (7) imply* $\alpha(\mathsf{lfp}^{\sqsubseteq} f) \curlywedge \bar{Q} \neq 0$*. Conversely (1) to (4) and* $\mathsf{lfp}^{\sqsubseteq} f \sqcap \gamma(\bar{Q}) \neq \bot$ *imply (5) to (7).*

Calculational design of the proof system

$\overline{\text{HL}}$ does not need a consequence rule

THEOREM 4.1 (EQUIVALENT DEFINITIONS OF $\overline{\text{HL}}$ THEORIES).

$$\mathcal{T}_{\overline{HL}}(\mathtt{S}) \triangleq \text{post}(\subseteq,\supseteq) \circ \alpha^{\neg} \circ \mathcal{T}_{HL}(\mathtt{S}) = \alpha^{\neg} \circ \mathcal{T}_{HL}(\mathtt{S})$$

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

PROOF OF TH. 4.1.

$$
\begin{array}{ll}
\mathcal{T}_{\overline{\text{HL}}}(\mathtt{S}) = \text{post}(\subseteq,\supseteq) \circ \alpha^{\neg} \circ \mathcal{T}_{\text{HL}}(\mathtt{S}) & \text{(def. } \mathcal{T}_{\overline{\text{HL}}}\text{)}\\
= \text{post}((\subseteq,\supseteq)(\neg\{\langle P,\ Q\rangle \mid \text{post}[\![\mathtt{S}]\!]P \subseteq Q\}) & \text{(Lem. 1.4 and def. (30) of } \alpha^{\neg}\text{)}\\
= \text{post}(\subseteq,\supseteq)(\{\langle P,\ Q\rangle \mid \neg(\text{post}[\![\mathtt{S}]\!]P \subseteq Q)\}) & \text{(def. } \neg\text{)}\\
= \text{post}(\subseteq,\supseteq)(\{\langle P,\ Q\rangle \mid \text{post}[\![\mathtt{S}]\!]P \cap \neg Q \neq \varnothing\}) & \text{(def. } \subseteq \text{ and } \neg\text{)}\\
= \{\langle P',\ Q'\rangle \mid \exists\langle P,\ Q\rangle \in \{\langle P,\ Q\rangle \mid \text{post}[\![\mathtt{S}]\!]P \cap \neg Q \neq \varnothing\}\ .\ \langle P,\ Q\rangle \subseteq,\supseteq \langle P',\ Q'\rangle\} & \text{(def. post)}\\
= \{\langle P',\ Q'\rangle \mid \exists\langle P,\ Q\rangle\ .\ \text{post}[\![\mathtt{S}]\!]P \cap \neg Q \neq \varnothing \wedge \langle P,\ Q\rangle \subseteq,\supseteq \langle P',\ Q'\rangle\} & \text{(def. } \in\text{)}\\
= \{\langle P',\ Q'\rangle \mid \exists\langle P,\ Q\rangle\ .\ \text{post}[\![\mathtt{S}]\!]P \cap \neg Q \neq \varnothing \wedge P \subseteq P' \wedge Q \supseteq Q'\} & \text{(component wise def. of } \subseteq,\supseteq\text{)}\\
= \{\langle P',\ Q'\rangle \mid \exists Q\ .\ \text{post}[\![\mathtt{S}]\!]P' \cap \neg Q \neq \varnothing \wedge Q \supseteq Q'\} &
\end{array}
$$

($\subseteq$) if $P \subseteq P'$ then $\text{post}[\![\mathtt{S}]\!]P \subseteq \text{post}[\![\mathtt{S}]\!]P'$ by (12) so that $\text{post}[\![\mathtt{S}]\!]P \cap \neg Q \neq \varnothing$ implies $\text{post}[\![\mathtt{S}]\!]P' \cap \neg Q \neq \varnothing$;
($\supseteq$) conversely, if $\exists Q\ .\ \text{post}[\![\mathtt{S}]\!]P'$, then $\exists P\ .\ \text{post}[\![\mathtt{S}]\!]P \cap \neg Q \neq \varnothing \wedge P \subseteq P'$ by choosing $P = P'$.

$$= \{\langle P',\ Q'\rangle \mid \text{post}[\![\mathtt{S}]\!]P' \cap \neg Q' \neq \varnothing\}$$

($\subseteq$) if $Q \supseteq Q'$ then $\neg Q' \supseteq \neg Q$ so $\text{post}[\![\mathtt{S}]\!]P' \cap \neg Q \neq \varnothing$ implies $\text{post}[\![\mathtt{S}]\!]P' \cap \neg Q' \neq \varnothing$;
($\supseteq$) conversely $\text{post}[\![\mathtt{S}]\!]P' \cap \neg Q' \neq \varnothing$ implies $\exists Q\ .\ \text{post}[\![\mathtt{S}]\!]P' \cap \neg Q \neq \varnothing \wedge Q \supseteq Q'$ by choosing $Q = Q'$.

$$
\begin{array}{ll}
= \{\langle P,\ Q\rangle \mid \neg(\text{post}[\![\mathtt{S}]\!]P \subseteq Q)\} & \text{(def. } \subseteq \text{ and } \neg\text{)}\\
= \alpha^{\neg} \circ \mathcal{T}_{\text{HL}}(\mathtt{S}) & \text{(def. } \alpha^{\neg} \text{ and } \mathcal{T}_{\text{HL}} \text{ for Hoare logic)} \quad \square
\end{array}
$$

Theory of $\overline{\text{HL}}$

Theorem 4.2 (Theory of $\overline{\text{HL}}$).

$$\mathcal{T}_{\overline{HL}}(\texttt{W}) = \{\langle P,\ Q\rangle \mid \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \langle \mathcal{B}[\![\texttt{B}]\!] \cap \{\sigma_i\},\ \neg\{\sigma_{i+1}\}\rangle \in \mathcal{T}_{\overline{HL}}(\texttt{S}) \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \notin Q\}$$

Proof of Th. 4.2. `W = while (B) S`

$\mathcal{T}_{\overline{\text{HL}}}(\texttt{W})$

$= \{\langle P,\ Q\rangle \mid \text{post}[\![\neg\texttt{B}]\!](\text{lfp}^{\subseteq} \bar{\bar{F}}^e_P) \cap \neg Q \neq \varnothing\}$ $\quad$ ⟅Lem. 1.3, where $\bar{\bar{F}}^e_P(X) \triangleq P \cup \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)X$⟆

$= \{\langle P,\ Q\rangle \mid \text{lfp}^{\subseteq} \bar{\bar{F}}^e_P \cap \text{pre}[\![\neg\texttt{B}]\!](\neg Q) \neq \varnothing\}$ $\quad$ ⟅(39.d)⟆

$= \{\langle P,\ Q\rangle \mid \exists I \in \wp(\Sigma) \,.\, \bar{\bar{F}}^e_P(I) \subseteq I \wedge \exists \langle W,\ \preccurlyeq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \exists \langle \sigma_i \in I,\ i \in [1, \infty]\rangle \,.\, \sigma_1 \in \bar{\bar{F}}^e_P(\varnothing) \wedge \forall i \in [1, \infty] \,.\, \sigma_{i+1} \in \bar{\bar{F}}^e_P(\{\sigma_i\}) \wedge \forall i \in [1, \infty] \,.\, (\sigma_i \neq \sigma_{i+1}) \Rightarrow (\nu(\sigma_i) > \nu(\sigma_{i+1}) \wedge \forall i \in [1, \infty] \,.\, (\nu(\sigma_i) \not> \nu(\sigma_{i+1}) \Rightarrow \{\sigma_i\} \cap \text{pre}[\![\neg\texttt{B}]\!](\neg Q) \neq 0\}$ $\quad$ ⟅induction principle Th. H.3⟆

$= \{\langle P,\ Q\rangle \mid \exists I \in \wp(\Sigma) \,.\, P \subseteq I \wedge \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)I \subseteq I \wedge \exists \langle W,\ \preccurlyeq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \exists \langle \sigma_i \in I,\ i \in [1, \infty]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, \infty] \,.\, (\sigma_{i+1} \in P \vee \{\sigma_{i+1}\} \subseteq \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)\{\sigma_i\}) \wedge \forall i \in [1, \infty] \,.\, (\sigma_i \neq \sigma_{i+1}) \Rightarrow (\nu(\sigma_i) > \nu(\sigma_{i+1}) \wedge \forall i \in [1, \infty] \,.\, (\nu(\sigma_i) \not> \nu(\sigma_{i+1}) \Rightarrow \sigma_i \in \text{pre}[\![\neg\texttt{B}]\!](\neg Q)\}$
⟅def. $\bar{\bar{F}}^e_P(X) \triangleq P \cup \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)X$, $\subseteq$, and post, which is $\varnothing$-strict⟆

$= \{\langle P,\ Q\rangle \mid \exists I \in \wp(\Sigma) \,.\, P \subseteq I \wedge \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)I \subseteq I \wedge \exists \langle W,\ \preccurlyeq\rangle \in \mathfrak{Wf} \,.\, \exists \nu \in I \to W \,.\, \exists \langle \sigma_i \in I,\ i \in [1, \infty]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, \infty] \,.\, \{\sigma_{i+1}\} \subseteq \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)\{\sigma_i\} \wedge \forall i \in [1, \infty] \,.\, (\sigma_i \neq \sigma_{i+1}) \Rightarrow (\nu(\sigma_i) > \nu(\sigma_{i+1}) \wedge \forall i \in [1, \infty] \,.\, (\nu(\sigma_i) \not> \nu(\sigma_{i+1}) \Rightarrow \sigma_i \in \text{pre}[\![\neg\texttt{B}]\!](\neg Q)\}$
⟅since if $\sigma_{i+1} \in P$, we can equivalently consider the sequence $\langle \sigma_j \in I,\ j \in [i+1, \infty]\rangle$⟆

$= \{\langle P,\ Q\rangle \mid \exists I \in \wp(\Sigma) \,.\, P \subseteq I \wedge \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)I \subseteq I \wedge \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \{\sigma_{i+1}\} \subseteq \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)\{\sigma_i\} \wedge \sigma_n \in \text{pre}[\![\neg\texttt{B}]\!](\neg Q)\}$

⟅($\subseteq$) By $\langle W,\ \preccurlyeq\rangle \in \mathfrak{Wf}$, $\nu \in I \to W$, $\forall i \in [1, \infty] \,.\, (\sigma_i \neq \sigma_{i+1}) \Rightarrow (\nu(\sigma_i) > \nu(\sigma_{i+1})$, the sequence is ultimately stationary at some rank n. For then on, $\sigma_{i+1} = \sigma_i$, $i \geqslant n$ and so $\nu(\sigma_i) = \nu(\sigma_{i+1})$. Therefore $\forall i \in [1, \infty] \,.\, (\nu(\sigma_i) \not> \nu(\sigma_{i+1}) \Rightarrow \sigma_i \notin Q$ implies that $\sigma_n \in \text{pre}[\![\neg\texttt{B}]\!](\neg Q)$;
($\supseteq$) Conversely, from $\langle \sigma_i \in I,\ i \in [1, n]\rangle$ we can define $W = \{\sigma_i \mid i \in [1, n]\} \cup \{-\infty\}$ with $-\infty < \sigma_i < \sigma_{i+1}$ and $\nu(x) = (\!| x \in \{\sigma_i \mid i \in [1, n]\ ?\ x : -\infty |\!)$ and the sequence $\langle \sigma_j \in I,\ j \in [1, \infty]\rangle$ repeats σ_n ad infimum for $j \geqslant n$.⟆

$= \{\langle P,\ Q\rangle \mid \exists I \in \wp(\Sigma) \,.\, P \subseteq I \wedge \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)I \subseteq I \wedge \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \{\sigma_{i+1}\} \subseteq \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)\{\sigma_i\} \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \notin Q\}$ $\quad$ ⟅def. pre⟆

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \{\sigma_{i+1}\} \subseteq \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)\{\sigma_i\} \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \notin Q\}$ $\quad$ ⟅I is not used and can always be chosen to be Σ⟆

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)\{\sigma_i\} \cap \{\sigma_{i+1}\} \neq \varnothing \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \notin Q\}$ $\quad$ ⟅since $x \in X \Leftrightarrow X \cap \{x\} \neq \varnothing$⟆

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)\{\sigma_i\} \cap \neg(\neg\{\sigma_{i+1}\}) \neq \varnothing \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \notin Q\}$ $\quad$ ⟅def. $\neg X = \Sigma \setminus X$⟆

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \neg(\text{post}([\![\texttt{B}]\!] \,⨾\, [\![\texttt{S}]\!]^e)\{\sigma_i\} \subseteq (\neg\{\sigma_{i+1}\})) \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \notin Q\}$ $\quad$ ⟅$\neg(X \subseteq Y) \Leftrightarrow (X \cap \neg Y \neq \varnothing$⟆

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \neg(\text{post}([\![\texttt{S}]\!]^e)(\mathcal{B}[\![\texttt{B}]\!] \cap \{\sigma_i\}) \subseteq (\neg\{\sigma_{i+1}\})) \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \notin Q\}$ $\quad$ ⟅def. post, $[\![\texttt{B}]\!]$, and ⨾⟆

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \langle \mathcal{B}[\![\texttt{B}]\!] \cap \{\sigma_i\},\ \neg\{\sigma_{i+1}\}\rangle \in \{\langle P,\ Q\rangle \mid \neg(\text{post}([\![\texttt{S}]\!]^e)P \subseteq Q)\} \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \notin Q\}$ $\quad$ ⟅def. $\in$⟆

$= \{\langle P,\ Q\rangle \mid \exists n \geqslant 1 \,.\, \exists \langle \sigma_i \in I,\ i \in [1, n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1, n[\,.\, \langle \mathcal{B}[\![\texttt{B}]\!] \cap \{\sigma_i\},\ \neg\{\sigma_{i+1}\}\rangle \in \mathcal{T}_{\overline{\text{HL}}}(\texttt{S}) \wedge \sigma_n \notin \mathcal{B}[\![\texttt{B}]\!] \wedge \sigma_n \in Q\}$ $\quad$ ⟅def. $\mathcal{T}_{\overline{\text{HL}}}(\texttt{S})$⟆ $\square$

Proof system of $\overline{\text{HL}}$

THEOREM 4.3 ($\overline{\text{HL}}$ RULES FOR CONDITIONAL ITERATION).

$$\frac{\exists \langle \sigma_i \in I,\ i \in [1,n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1,n[\,.\, (\!|\, \mathcal{B}[\![\mathtt{B}]\!] \cap \{\sigma_i\} \,|\!)\ \mathtt{S}\ (\!|\, \neg\{\sigma_{i+1}\} \,|\!) \wedge \sigma_n \notin \mathcal{B}[\![\mathtt{B}]\!] \wedge \sigma_n \notin Q}{(\!|\, P \,|\!)\ \mathtt{while\ (B)\ S}\ (\!|\, Q \,|\!)} \quad (3)$$

PROOF OF (3). We write $(\!| P |\!)\ \mathtt{S}\ (\!| Q |\!) \triangleq \langle P,\ Q\rangle \in \overline{\text{HL}}(\mathtt{S})$;

By structural induction ($\mathtt{S}$ being a strict component of $\mathtt{while\ (B)\ S}$), the rule for $(\!| P |\!)\ \mathtt{S}\ (\!| Q |\!)$ have already been defined;

By Aczel method, the (constant) fixpoint $\mathsf{lfp}^{\subseteq}\, \boldsymbol{\lambda} X \bullet S$ is defined by $\{\frac{\emptyset}{c} \mid c \in S\}$;

So for $\mathtt{while\ (B)\ S}$ we have an axiom $\dfrac{\emptyset}{(\!| P |\!)\ \mathtt{while\ (B)\ S}\ (\!| Q |\!)}$ with side condition $\exists \langle \sigma_i \in I,\ i \in [1,n]\rangle \,.\, \sigma_1 \in P \wedge \forall i \in [1,n[\,.\, (\!|\, \mathcal{B}[\![\mathtt{B}]\!] \cap \{\sigma_i\} \,|\!)\ \mathtt{S}\ (\!|\, \neg\{\sigma_{i+1}\} \,|\!) \wedge \sigma_n \notin \mathcal{B}[\![\mathtt{B}]\!] \wedge \sigma_n \notin Q$ where $(\!|\, \mathcal{B}[\![\mathtt{B}]\!] \cap \{\sigma_i\} \,|\!)\ \mathtt{S}\ (\!|\, \neg\{\sigma_{i+1}\} \,|\!)$ is well-defined by structural induction;

Traditionally, the side condition is written as a premiss, to get (3). □

About incorrectness

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

$$
\begin{array}{rcl}
\neg(\{P\}\,\mathsf{S}\{Q\}) & \begin{array}{c}\not\Rightarrow\\ \Leftarrow\end{array} & [P]\mathsf{S}[\neg Q] \\
 & \Leftrightarrow & \exists R \in \wp(\Sigma)\,.\,[P]\,\mathsf{S}\,[R] \wedge R \cap \neg Q \neq \emptyset \\
 & \Leftrightarrow & \exists \sigma \in \Sigma\,.\,[P]\,\mathsf{S}\,[\{\sigma\}] \wedge \sigma \notin Q
\end{array}
$$

Conclusion

- Was Peter correct or incorrect?

Conclusion

- Was Peter correct or incorrect?
- In a certain sense, he was correct
- BUT he took the hardest path

Conclusion

- Was Peter correct or incorrect?
- In a certain sense, he was correct
- BUT he took the hardest path
- Hoare incorrectness logic is the easiest and most popular way

Conclusion

- Was Peter correct or incorrect?
- In a certain sense, he was correct
- BUT he took the hardest path
- Hoare incorrectness logic is the easiest and most popular way
  - It has proof verifiers and theorem provers

Conclusion

- Was Peter correct or incorrect?
- In a certain sense, he was correct
- BUT he took the hardest path
- Hoare incorrectness logic is the easiest and most popular way
  - It has proof verifiers and theorem provers
  - They are called debuggers

Conclusion

- Was Peter correct or incorrect?
- In a certain sense, he was correct
- BUT he took the hardest path
- Hoare incorrectness logic is the easiest and most popular way
  - It has proof verifiers and theorem provers
  - They are called debuggers
  - Which are therefore formal tools based on a formal logic!

Conclusion

- Was Peter correct or incorrect?
- In a certain sense, he was correct
- BUT he took the hardest path
- Hoare incorrectness logic is the easiest and most popular way
  - It has proof verifiers and theorem provers
  - They are called debuggers
  - Which are therefore formal tools based on a formal logic! 😅

The End, Thank You

The End, Thank You

Happy Sixties to Peter